Frequency Filtering

CSC 767

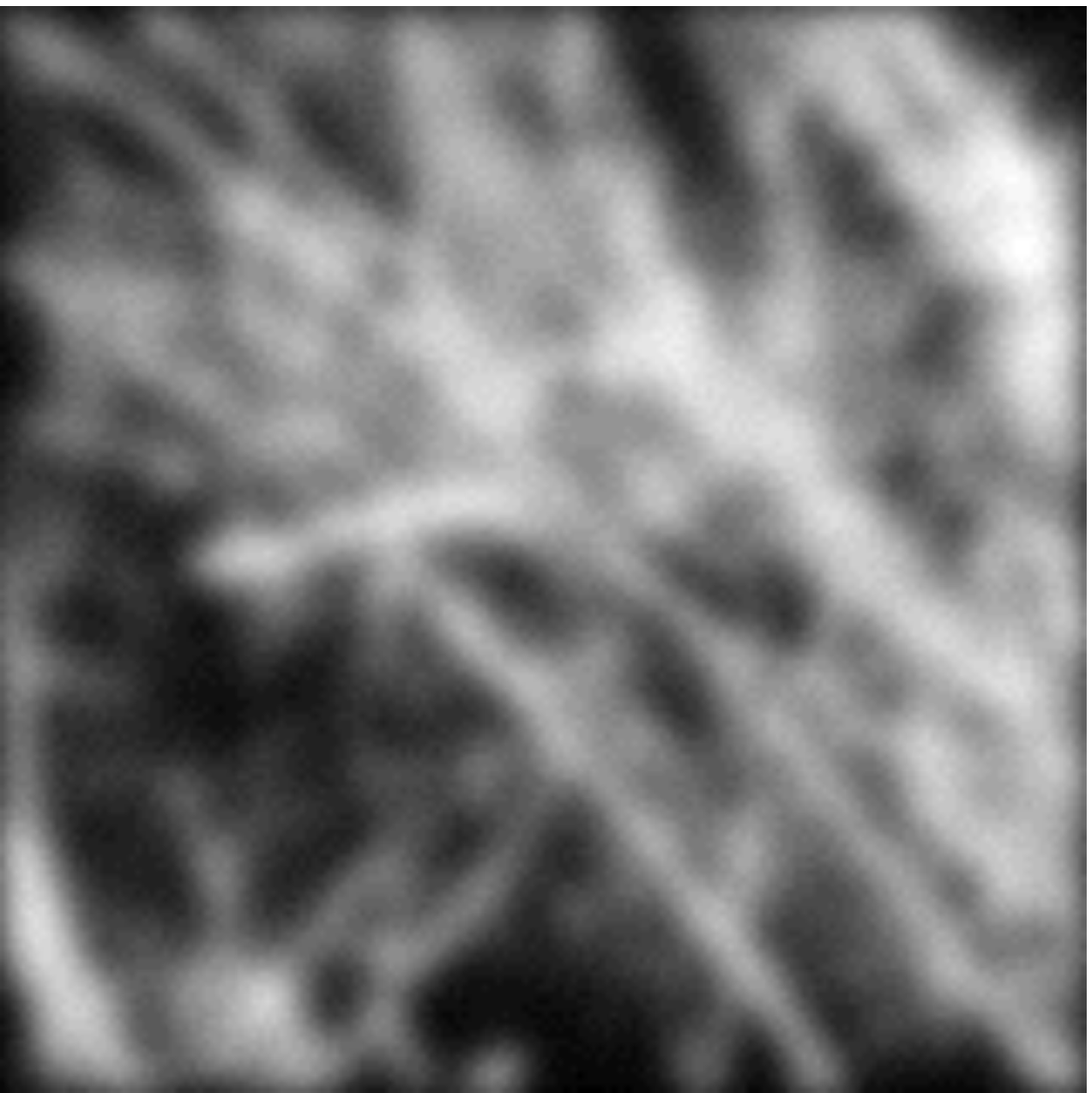
Outline

- Fourier transform and frequency domain
  - Frequency view of filtering
  - Hybrid images
  - Sampling

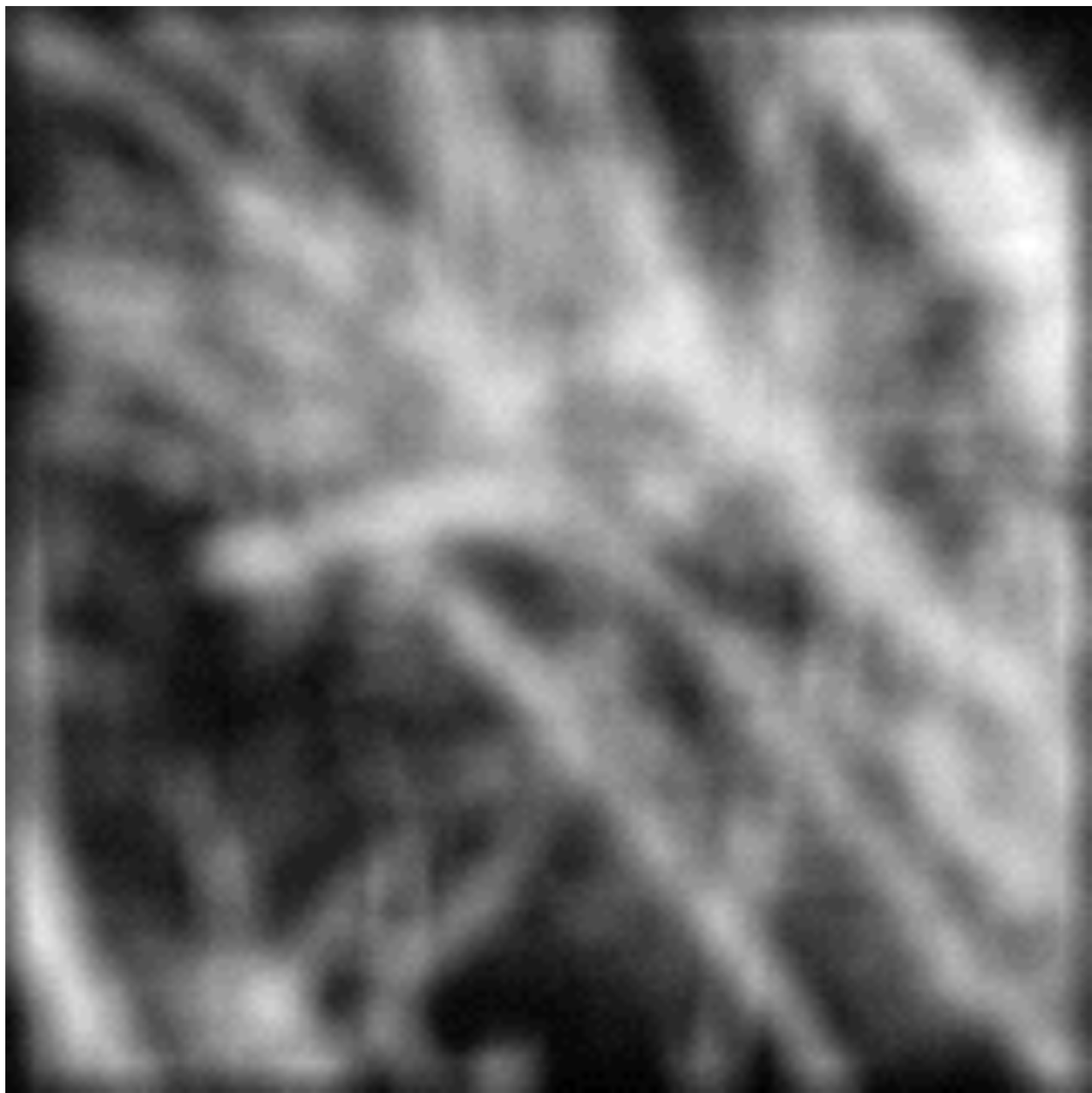
Slide: Hoiem

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

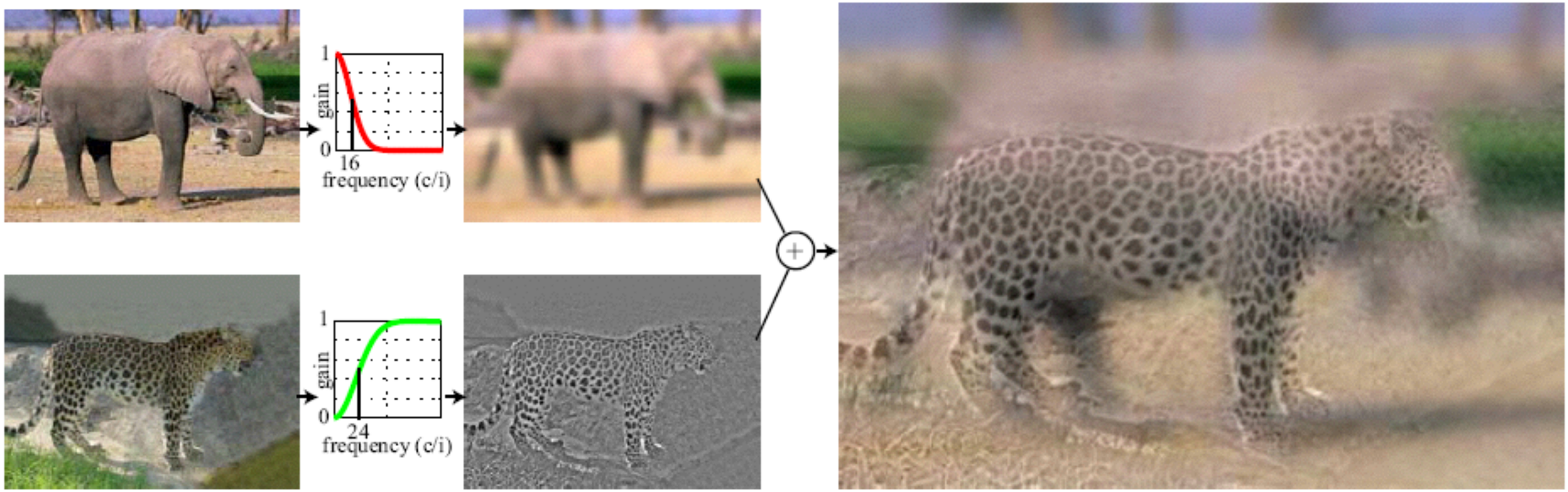
Gaussian 



Box filter 



Hybrid Images



- A. Oliva, A. Torralba, P.G. Schyns, "Hybrid Images," SIGGRAPH 2006

Fourier Transform

- 1-D Fourier transform pair
  - Forward transform $\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$
  - Inverse transform $f(x) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{2\pi i x \omega} d\omega$

- $f(x)$ is the original signal
- $\hat{f}(\omega)$ is representation of f in Fourier space

Fourier Transform

- Useful identity $e^{\imath\phi} = \cos(\phi) + \imath \sin(\phi)$

- Hence

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{-2\pi \imath x \omega} d\omega$$

$$= \int_{-\infty}^{\infty} \hat{f}(\omega) \left(\cos(2\pi x\omega) + \imath \sin(2\pi x\omega)\right) d\omega$$

- f(x) is a “sum” of sine and cosine functions
- The sine and cos functions have increasing frequencies
- They are weighted by $\hat{f}(\omega)$

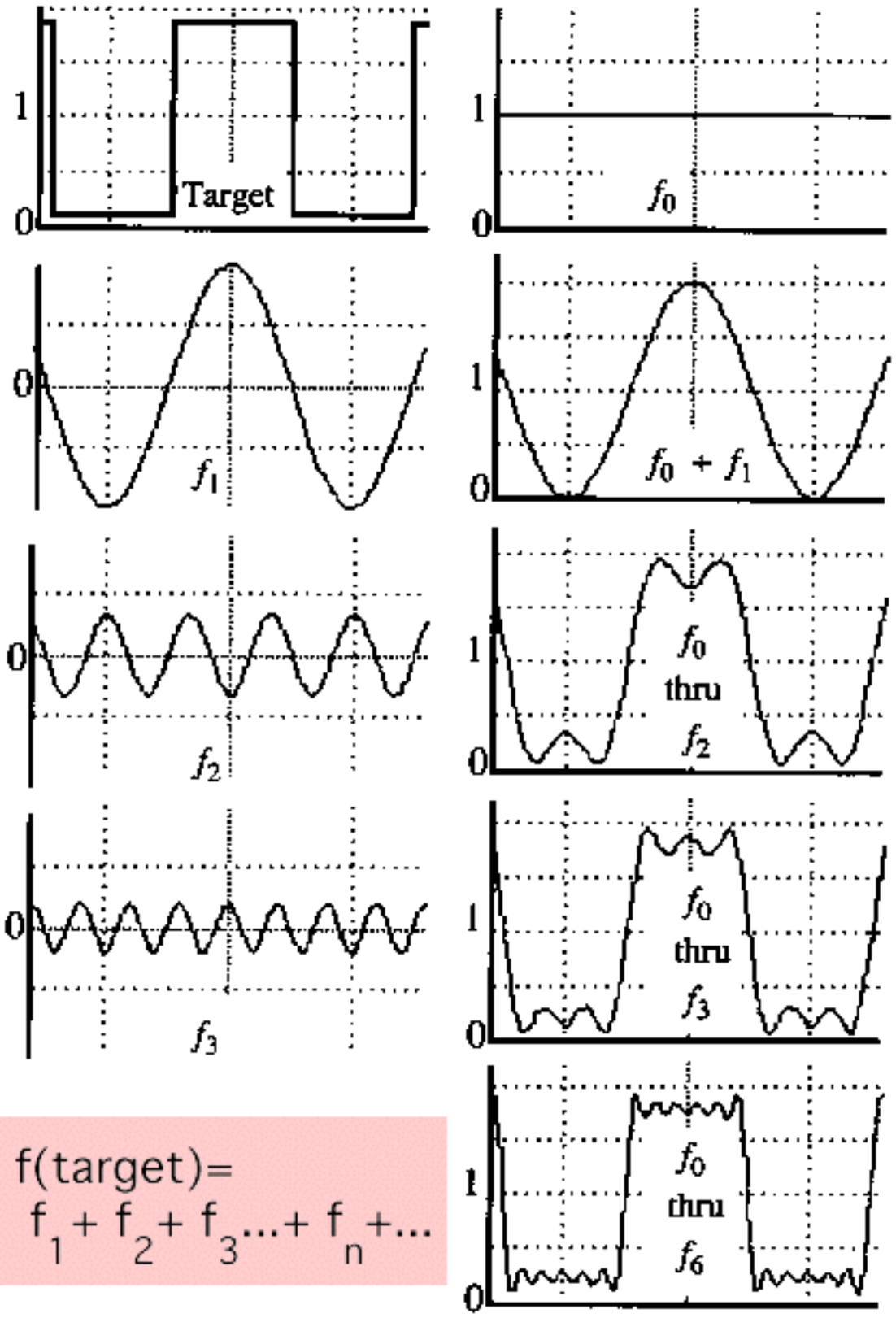
A sum of sines

The building block:

$$A\sin(\omega x + \phi)$$

Add enough of them to get *almost* any signal ***g(x)*** you want!

$$g(x) \approx \frac{A_0}{2} + \sum_{k=1}^{n} A_k \sin(2\pi k x/P + \phi_k)$$

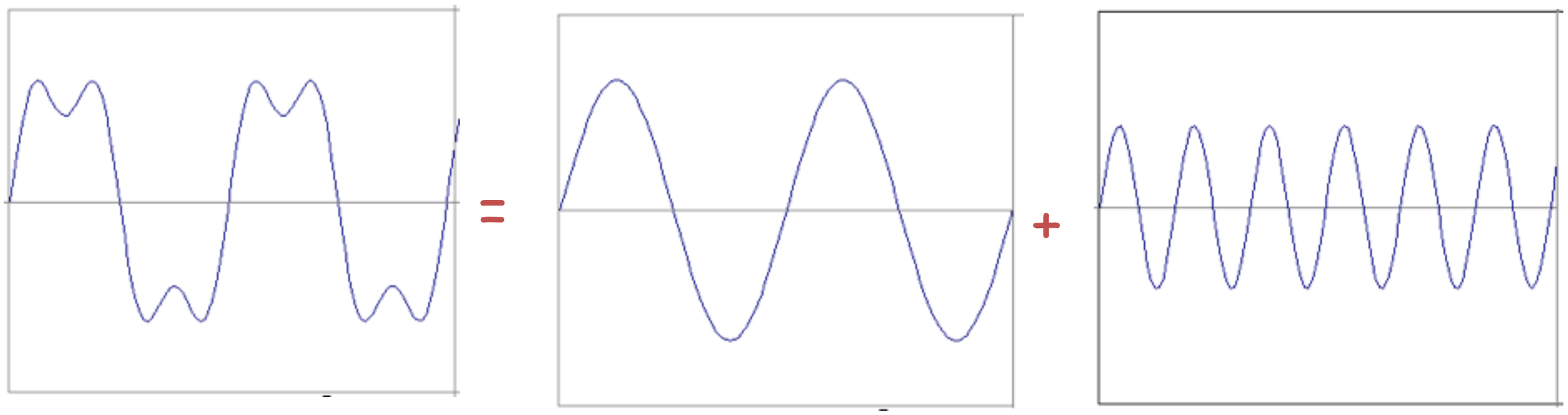


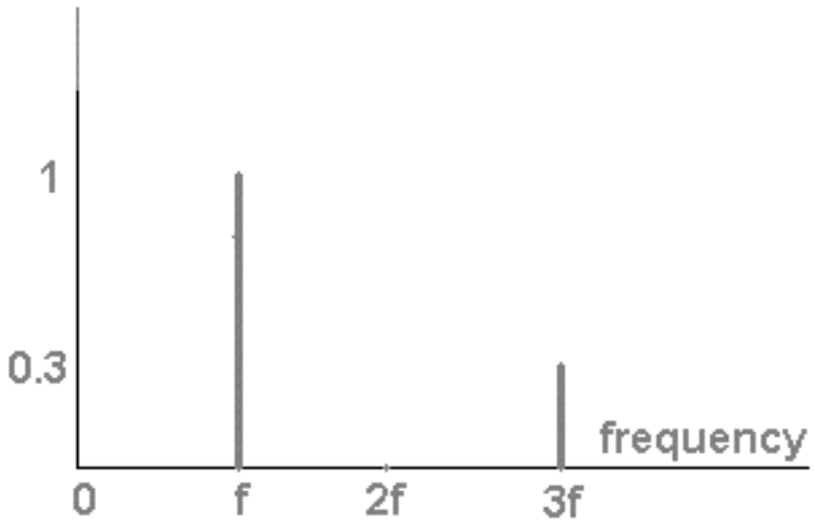
f(target)=
$f_1 + f_2 + f_3 ... + f_n + ...$

Slide: J. Hays

Frequency Spectra

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

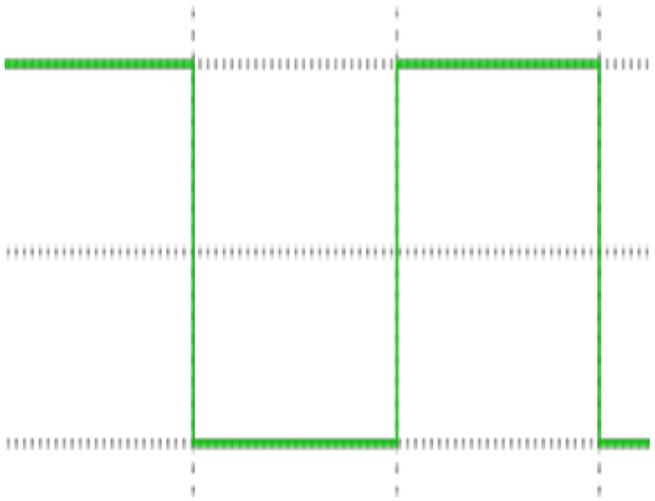




Slide: J. Hays

Slides: Efros

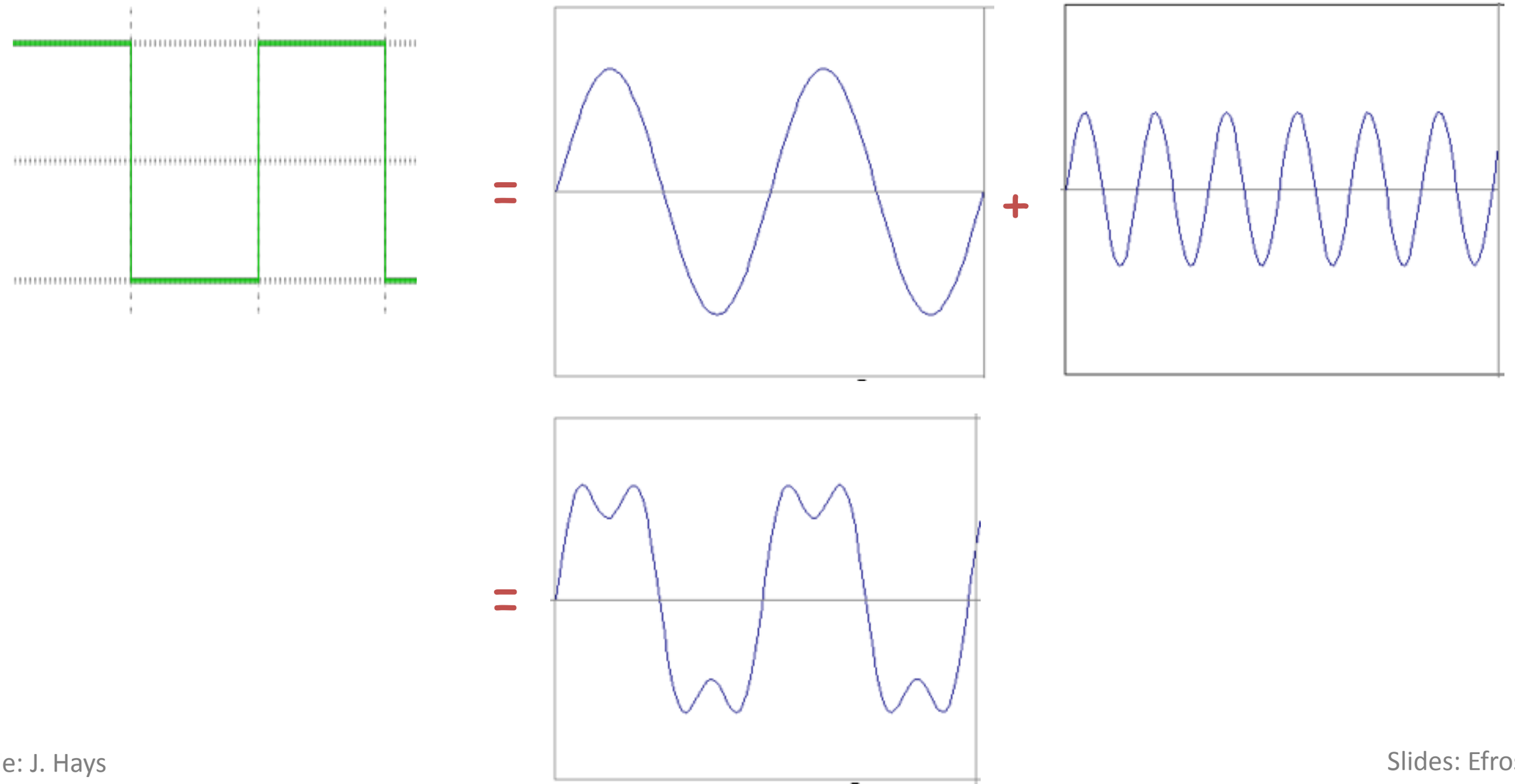
Frequency Spectra



Slide: J. Hays

Slides: Efros

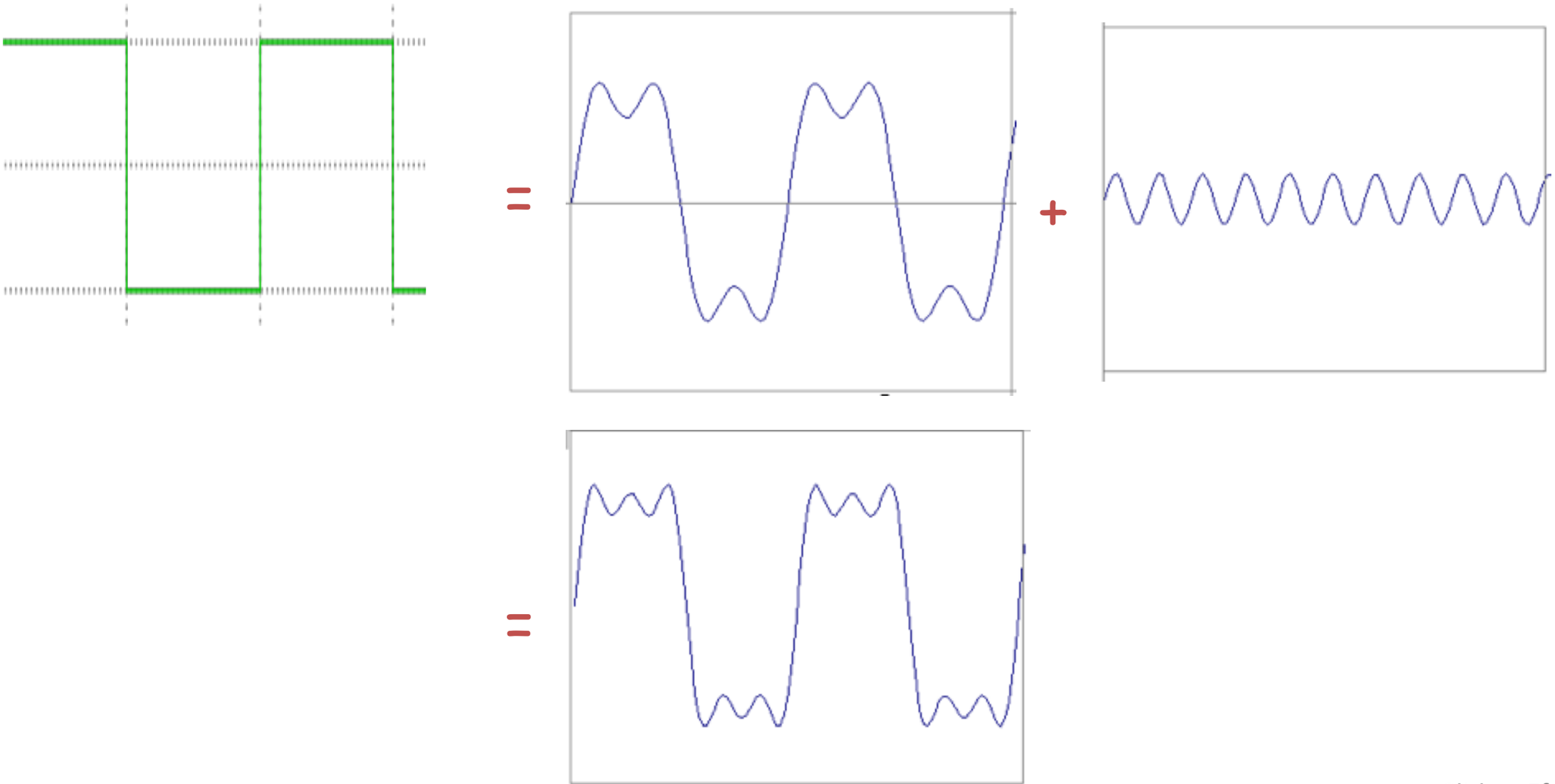
Frequency Spectra



Slide: J. Hays

Slides: Efros

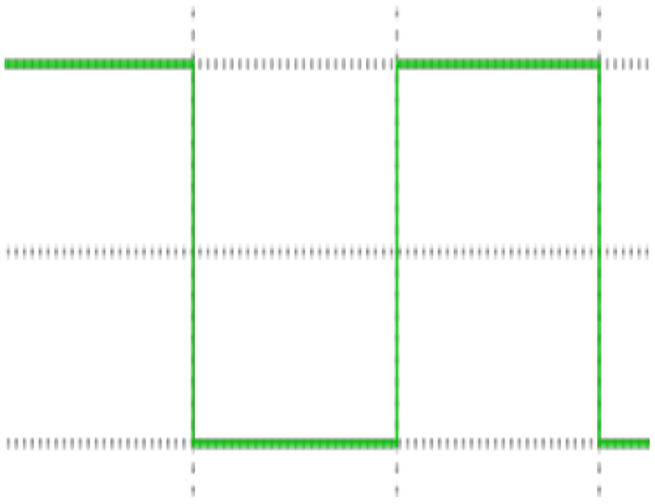
Frequency Spectra



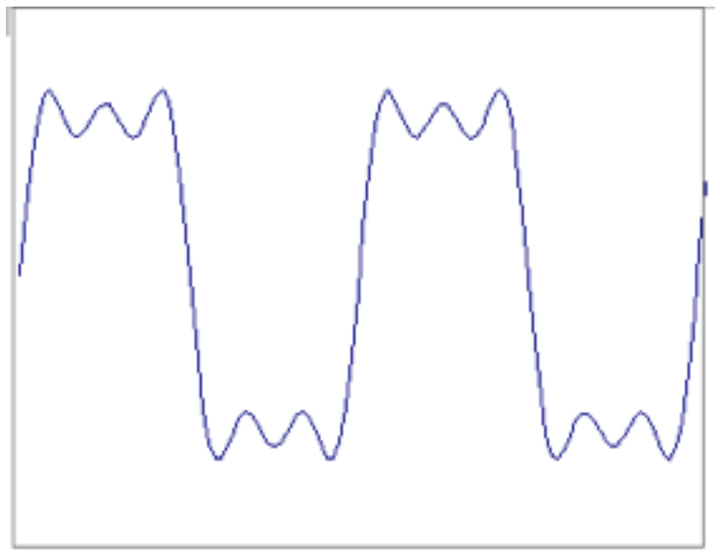
Slide: J. Hays

Slides: Efros

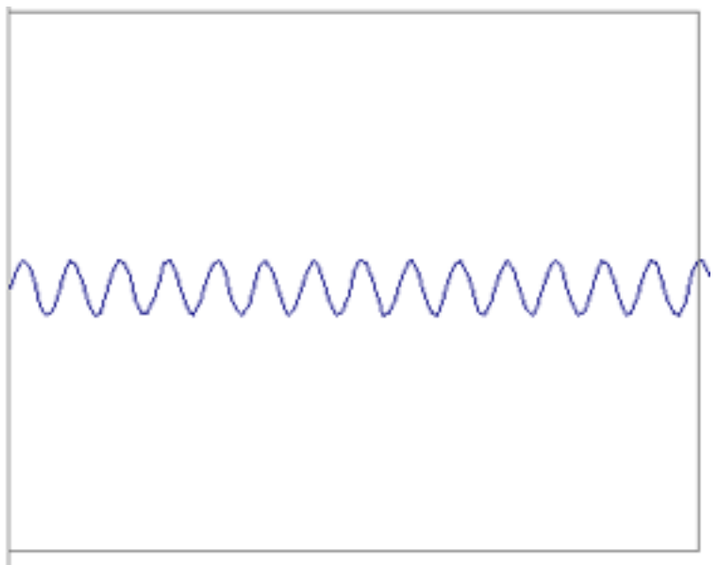
Frequency Spectra



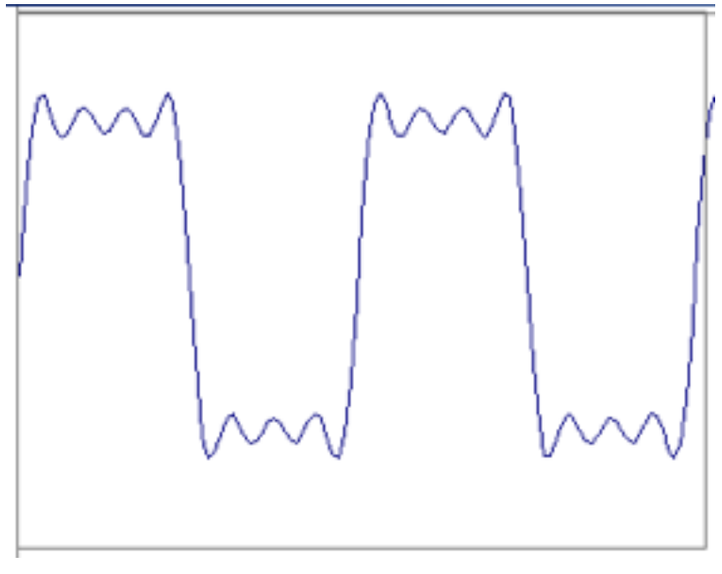
=



+



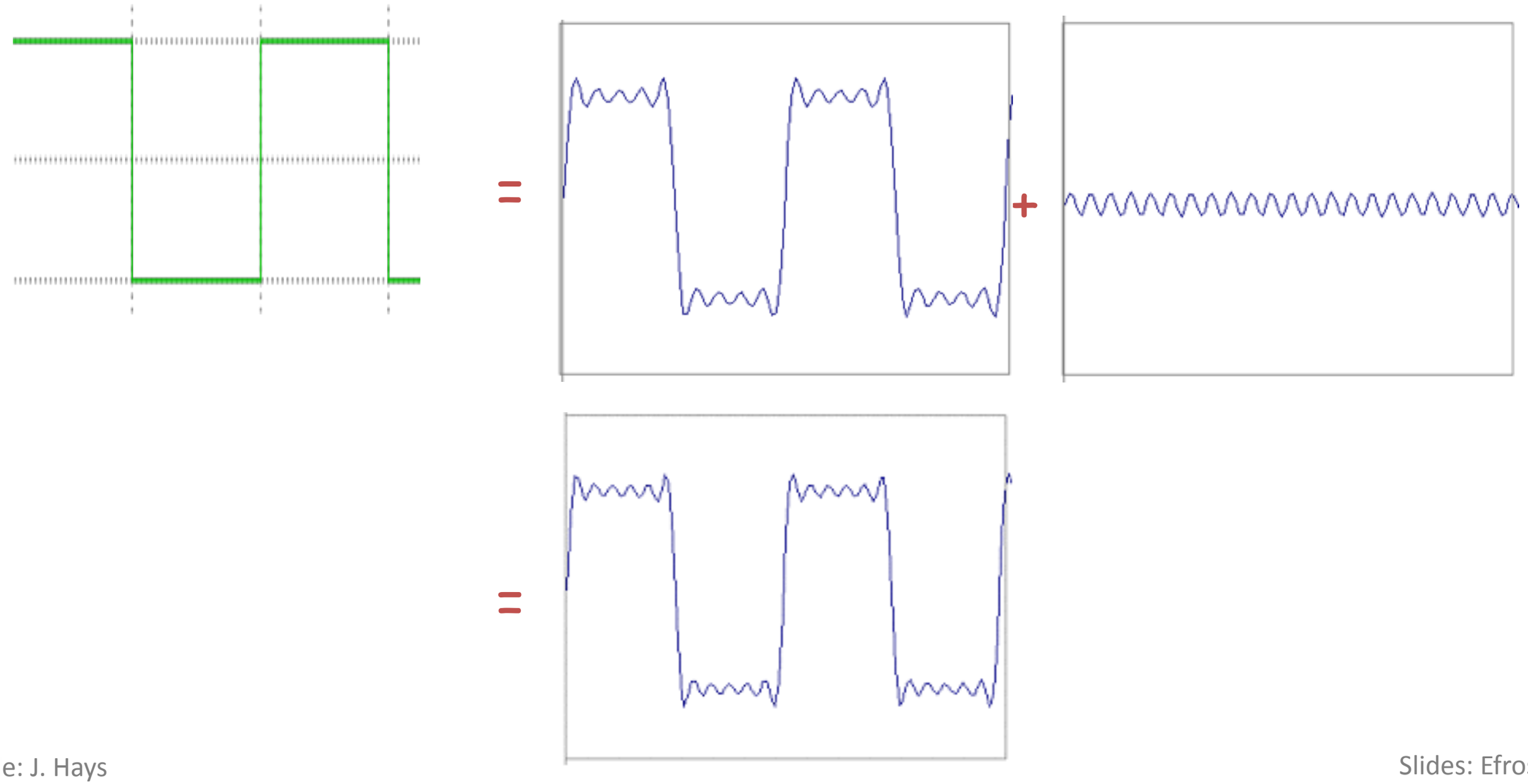
=



Slide: J. Hays

Slides: Efros

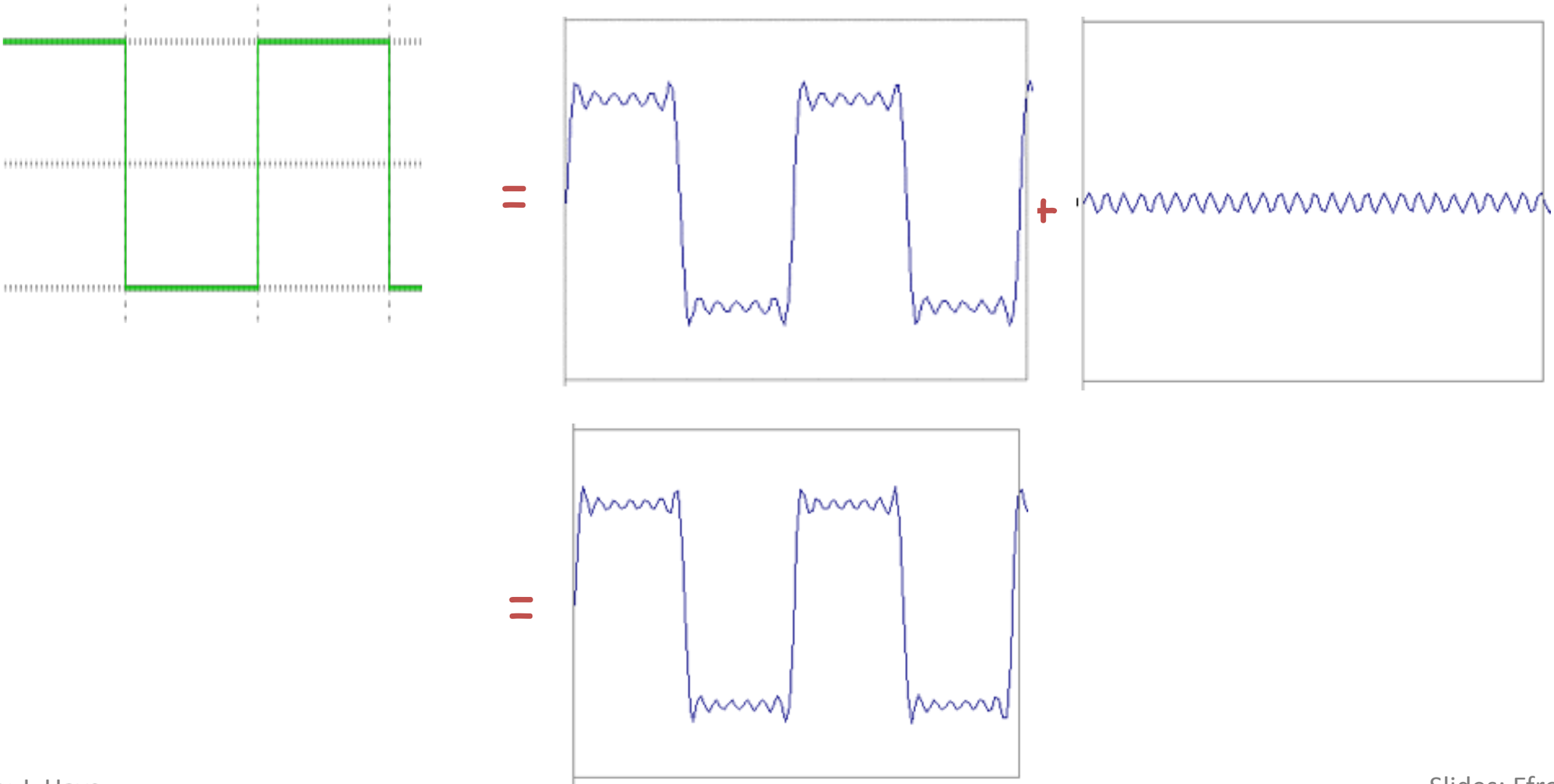
Frequency Spectra



Slide: J. Hays

Slides: Efros

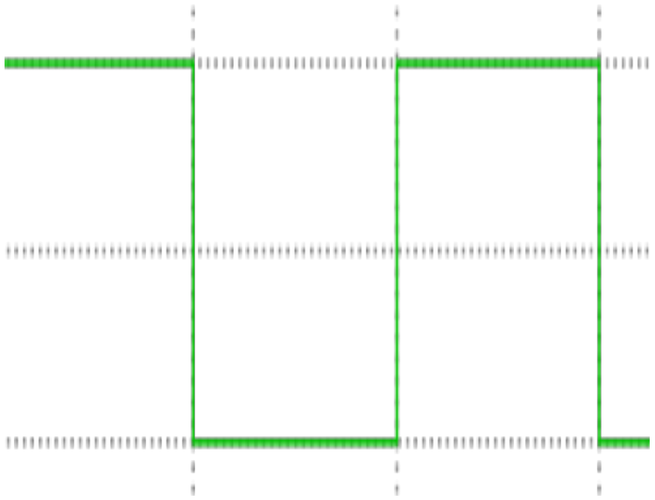
Frequency Spectra



Slide: J. Hays

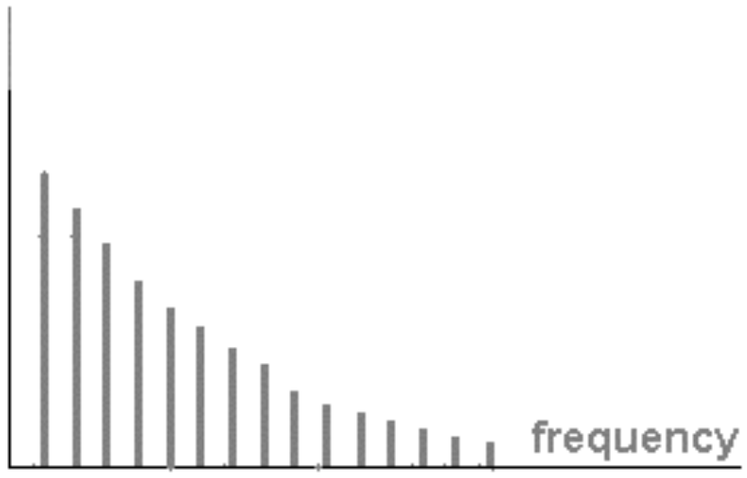
Slides: Efros

Frequency Spectra



=

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

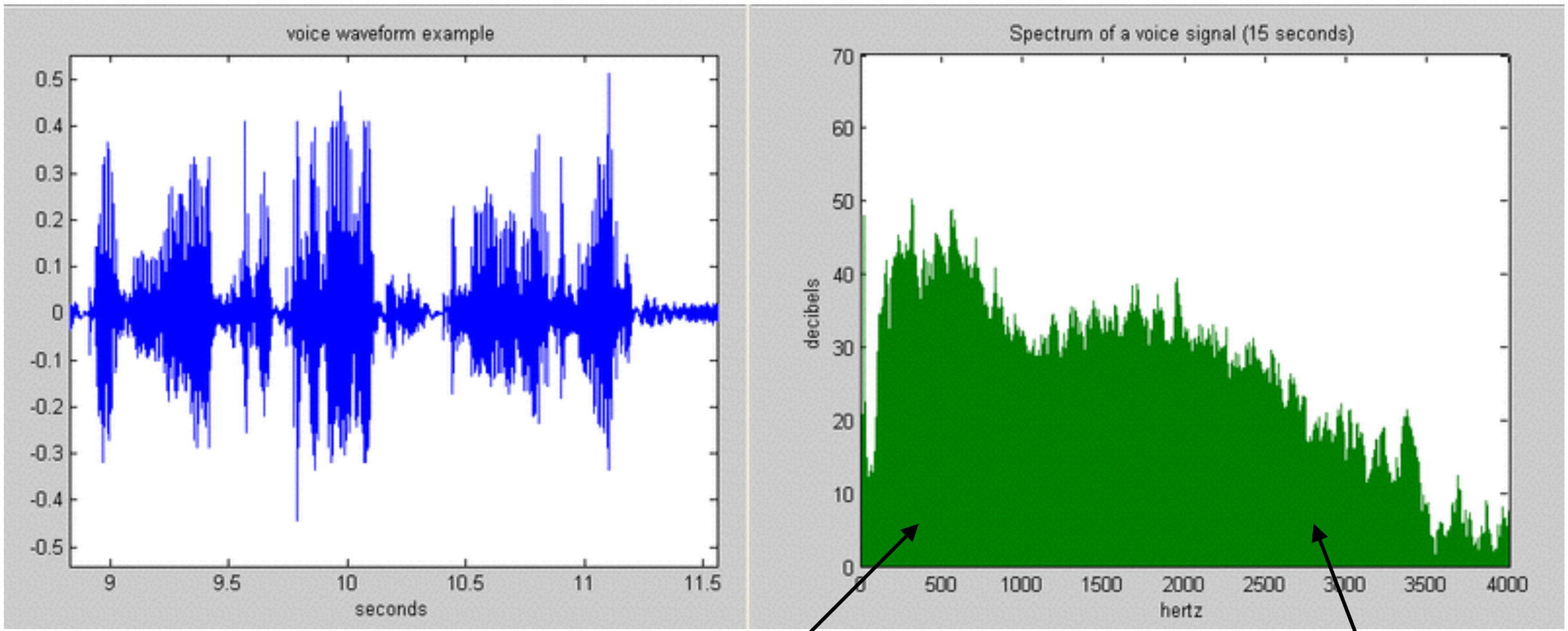


Slide: J. Hays

Slides: Efros

Example: Music

- We think of music in terms of frequencies at different L



Low frequencies

High frequencies

Slide: Hoiem

2-D Fourier transforms

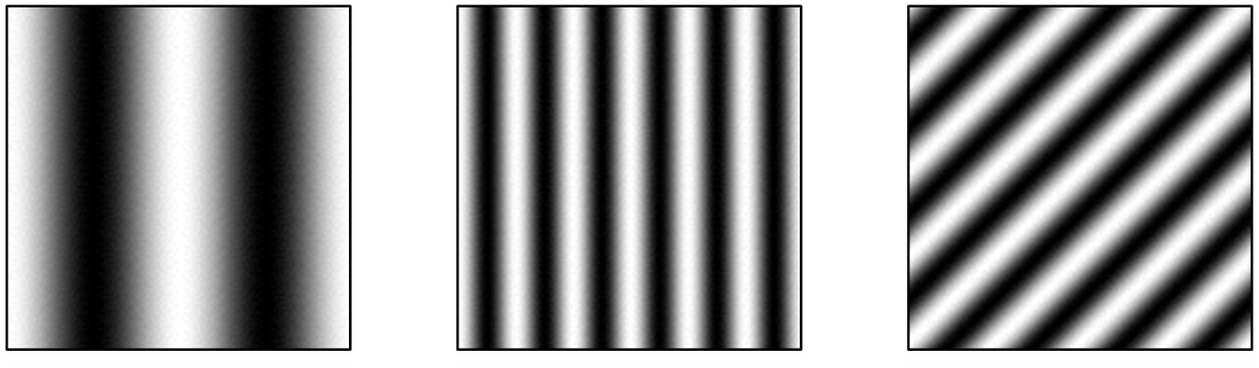
- Discrete 2-D Fourier pair
  - Forward transform $$F(u,v) = \sum_{x=0,y=0}^{N-1} f(x,y)e^{-2\pi\imath(ux+vy)/N}$$
  - Inverse transform $$f(x,y) = \sum_{u=0,v=0}^{N-1} F(u,v)e^{2\pi\imath(ux+vy)/N}$$

- Forward 2-D transform in Matlab

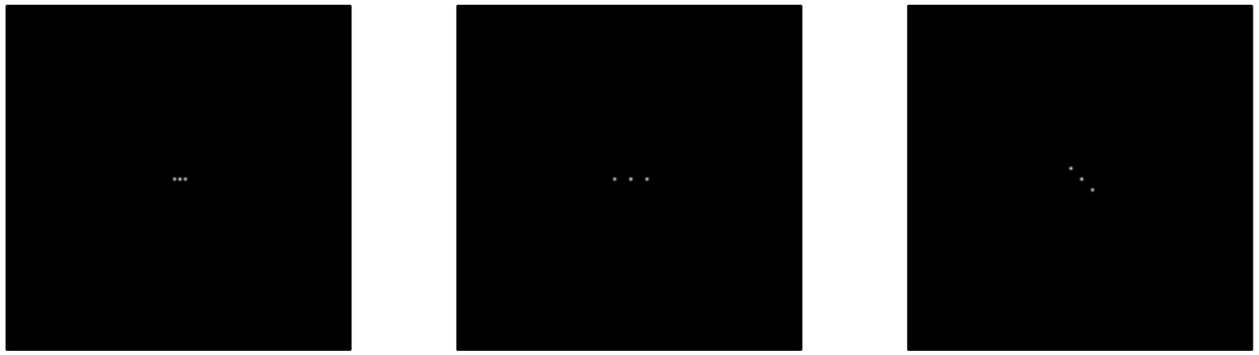
```
>> g = imread('puppy.jpg');
>> g = double(rgb2gray(g));
>> G = fft2(g);
```

Fourier analysis in images

Intensity Image

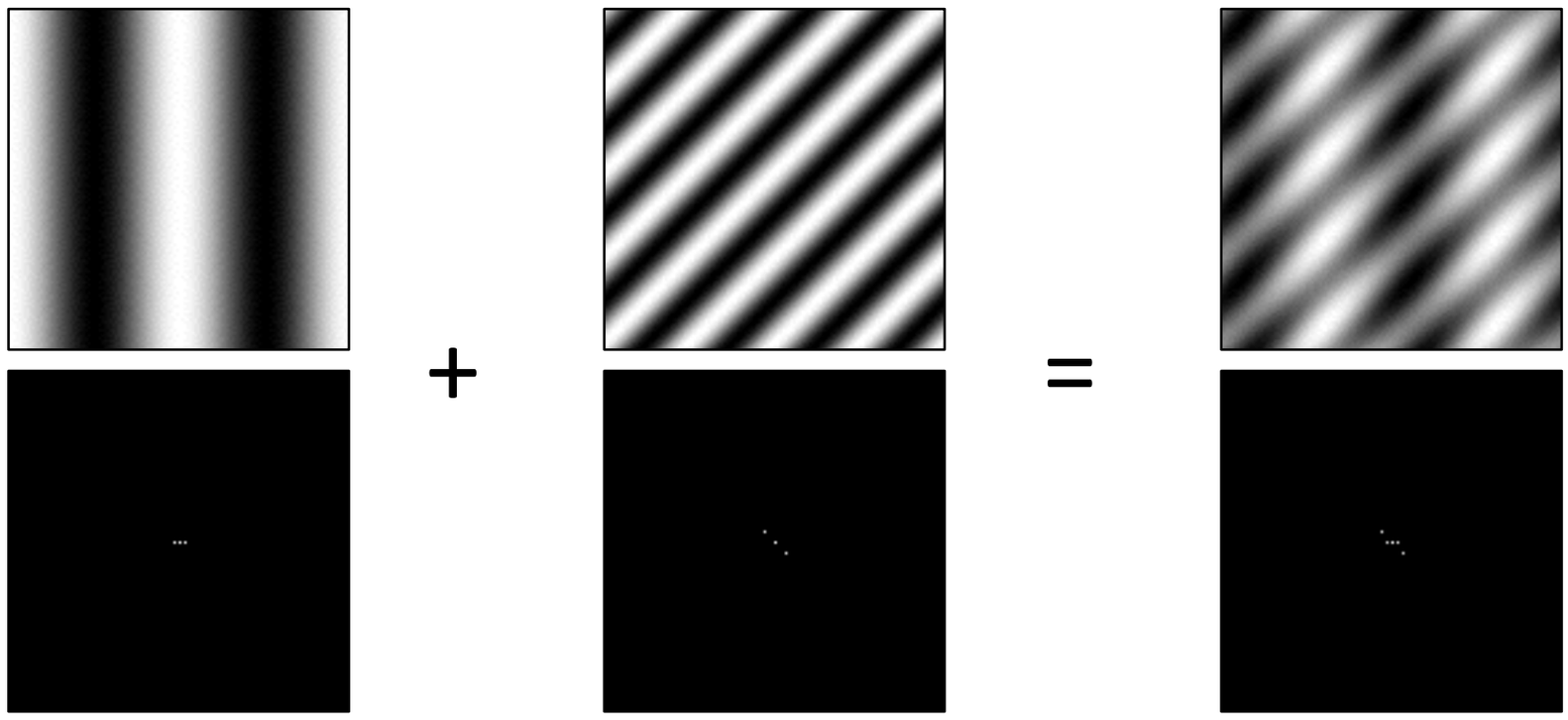


Fourier Image



http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Signals can be composed



http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html

2-D Fourier Transform

- The 2-D Fourier transform stores the magnitude and phase at each frequency

$$A(u,v) = \sqrt{Real(G(u,v))^2 + Imag(G(u,v))^2}$$

$$\Phi(u,v) = \tan^{-1}\frac{Imag(G(u,v))}{Real(G(u,v))}$$

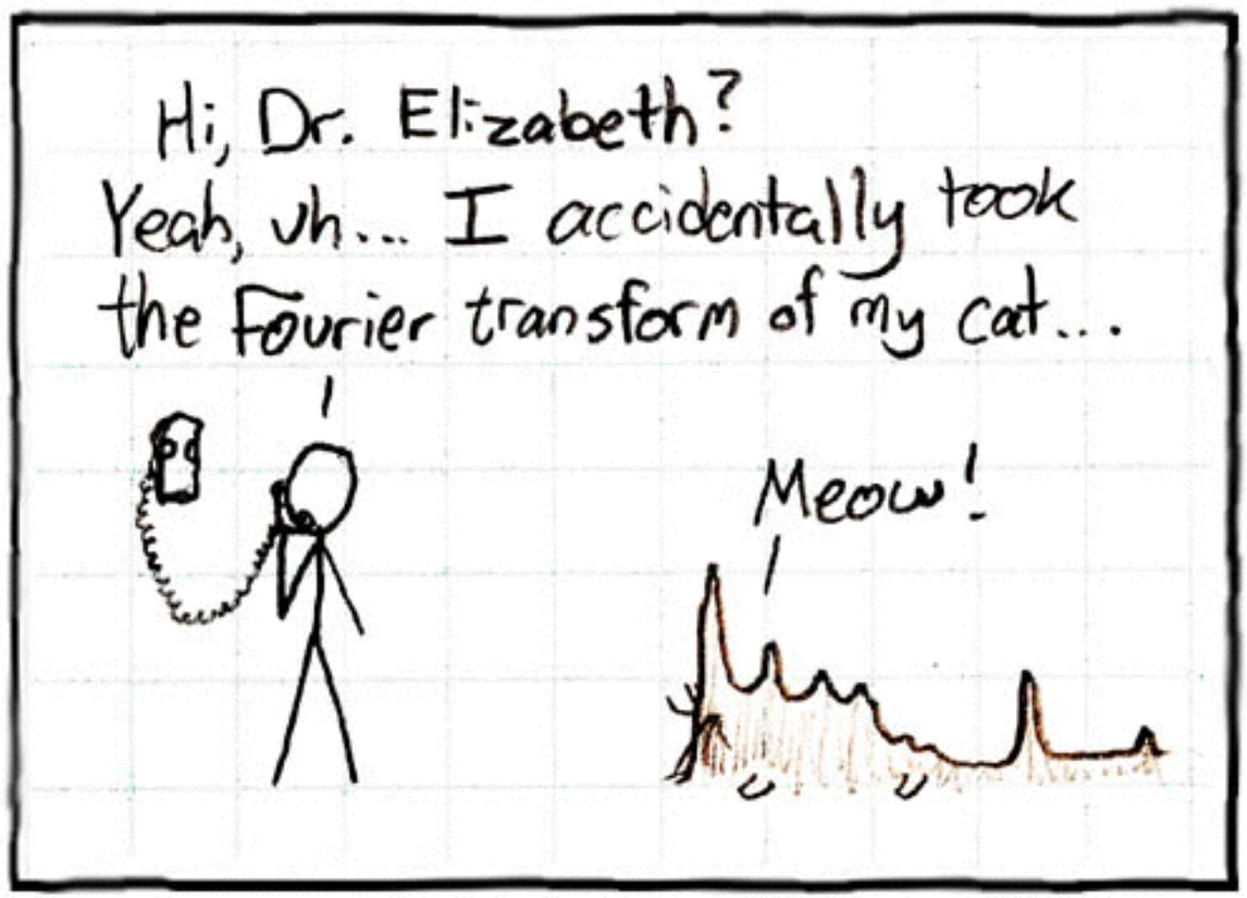
  - Magnitude encodes how much signal there is at a particular frequency
  - Phase encodes direction information of the image content

- In Matlab

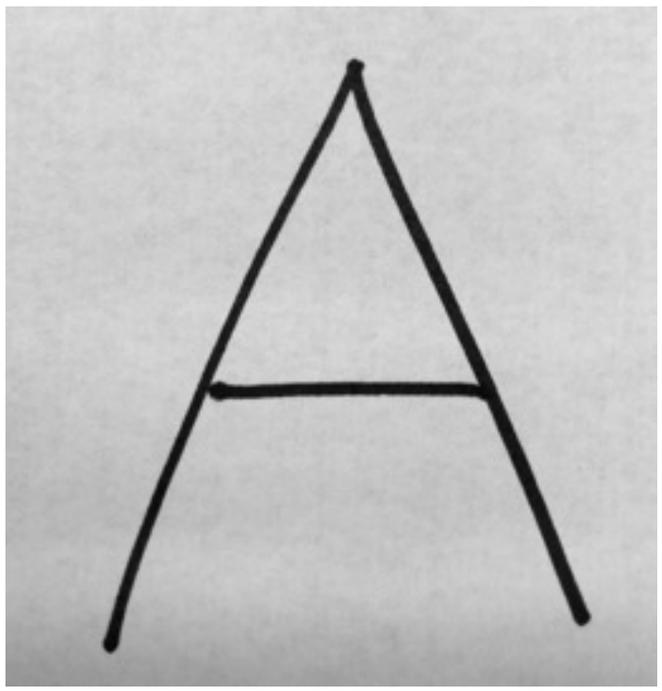
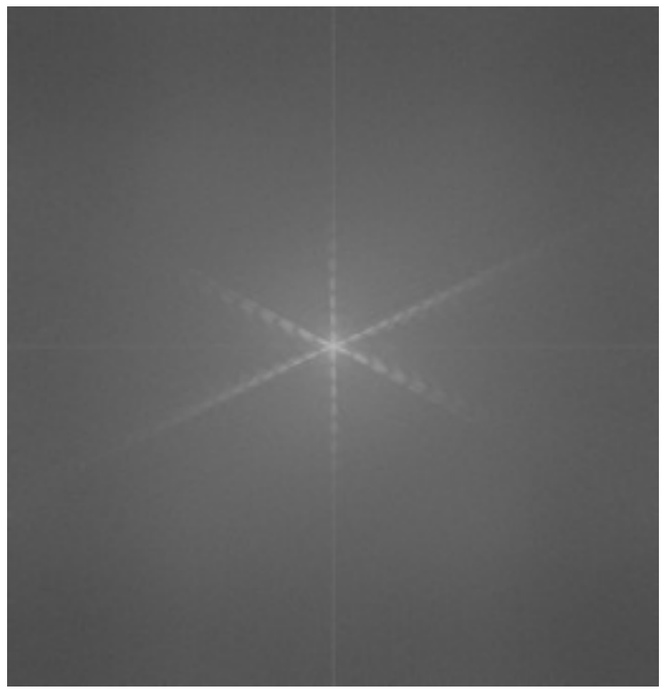
```
>> A = abs(G);
>> Phase = atan(imag(G)./real(G));
```

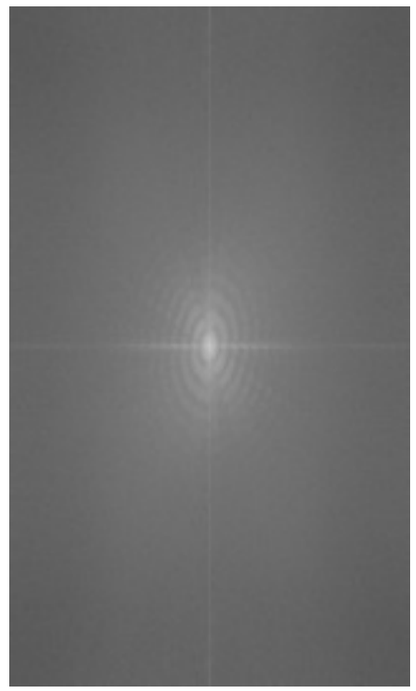
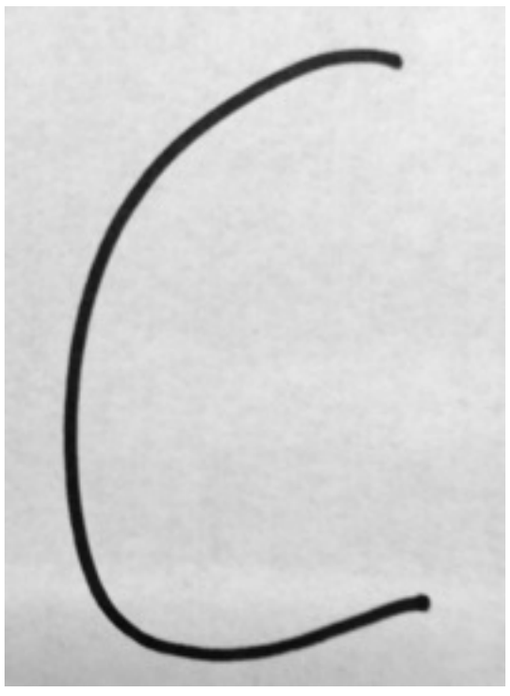
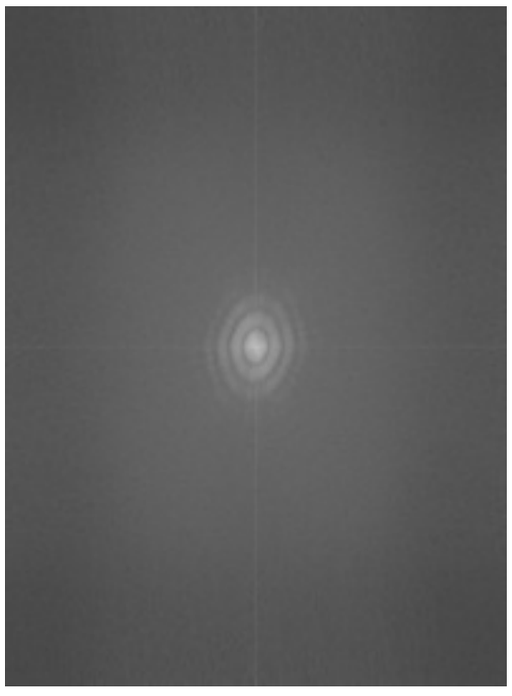
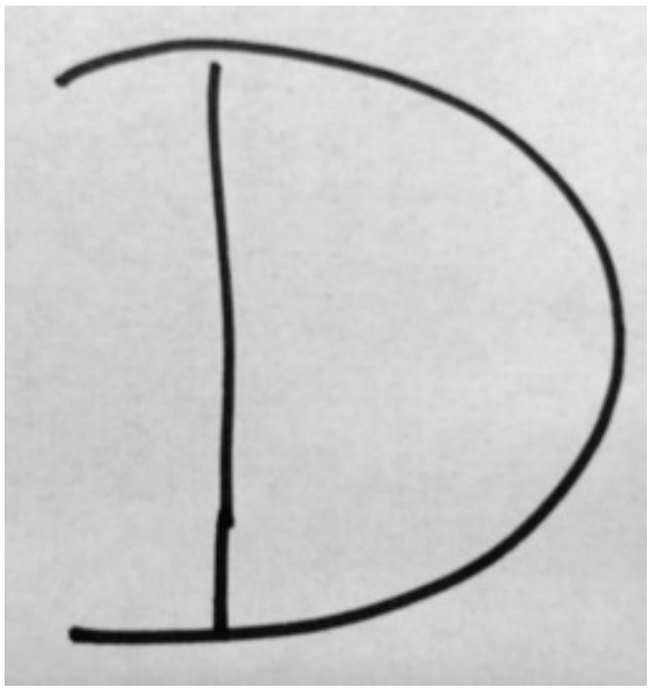
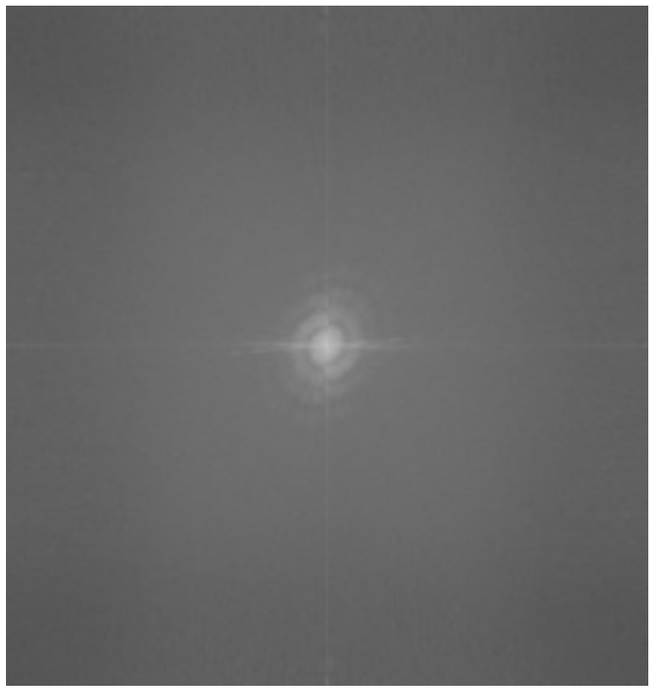
Slide: J. Hays

Examples of 2-D Fourier representation

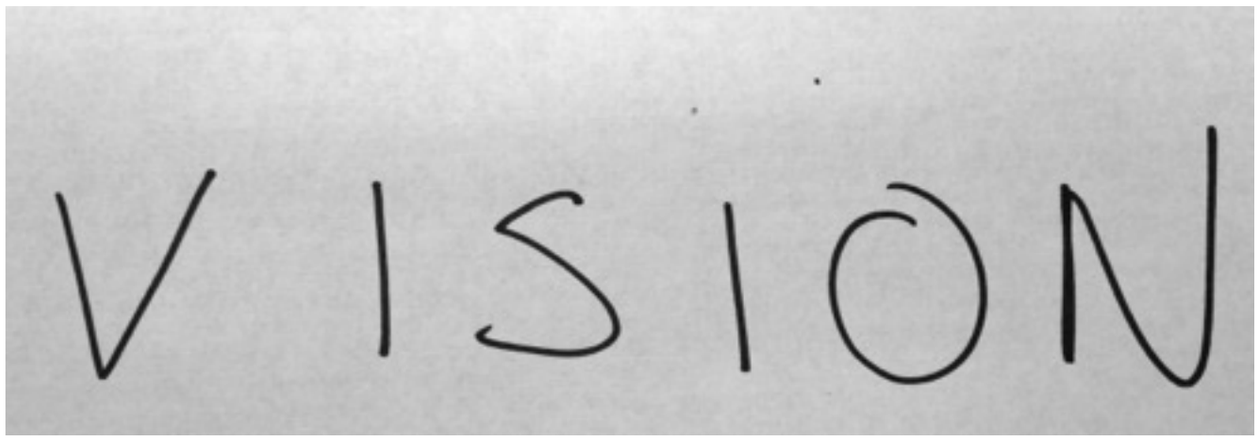


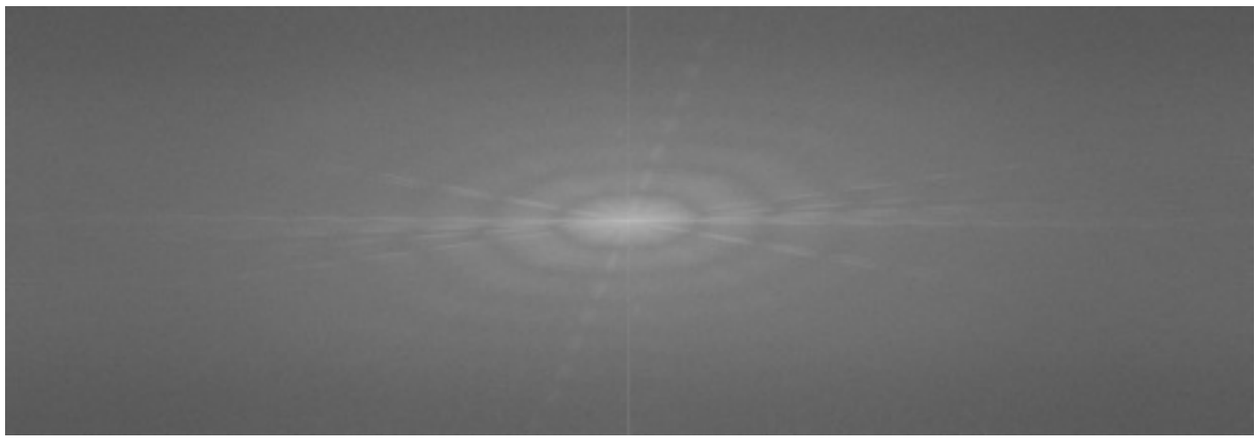
Single Letters

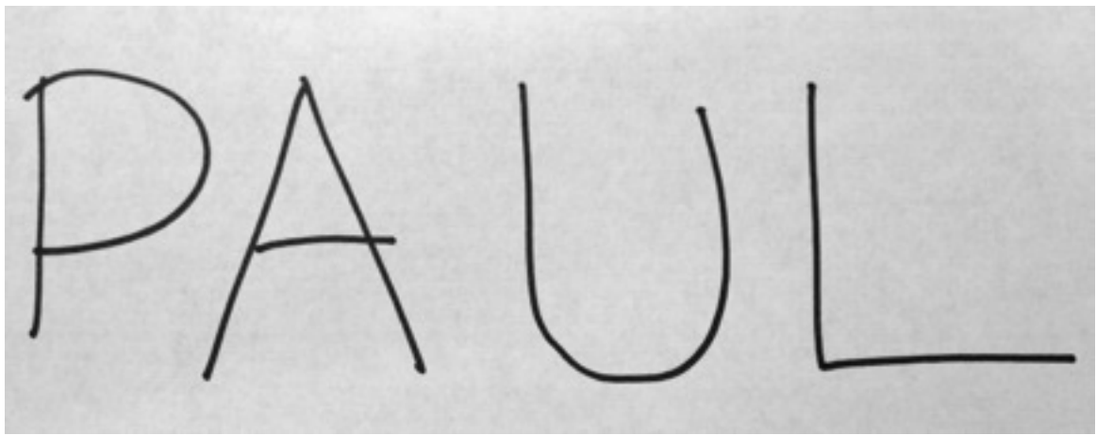



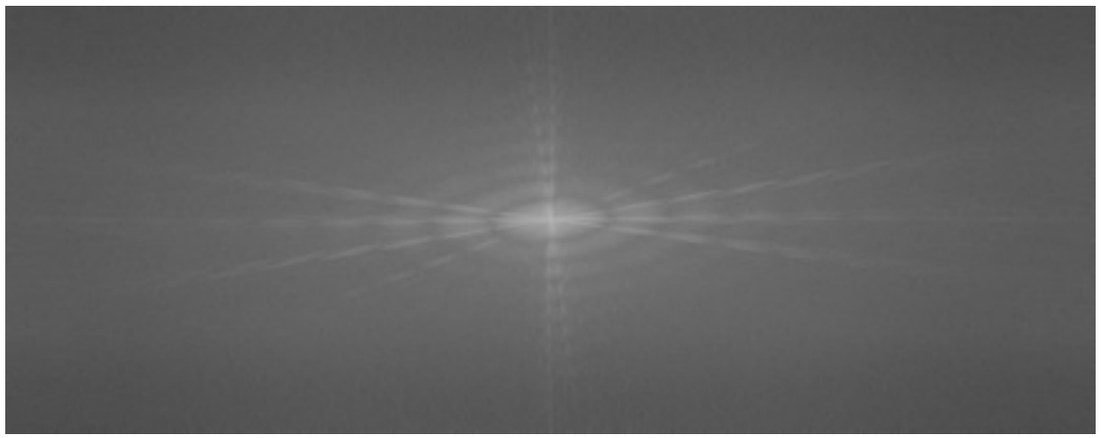






Words

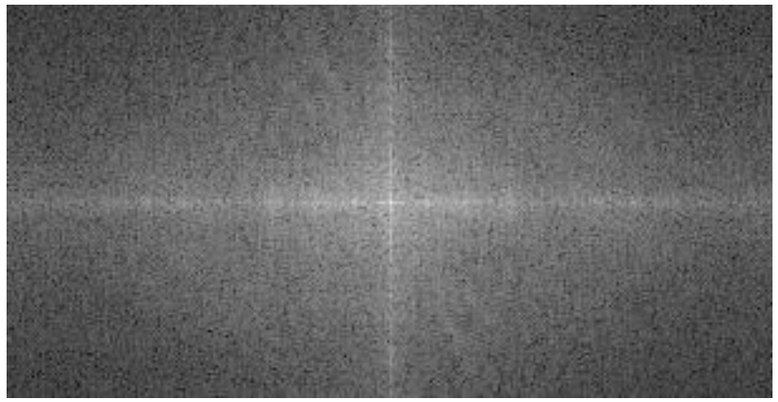


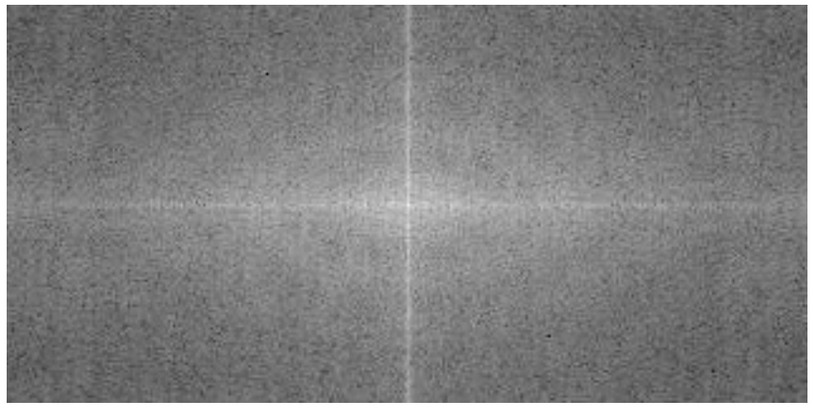


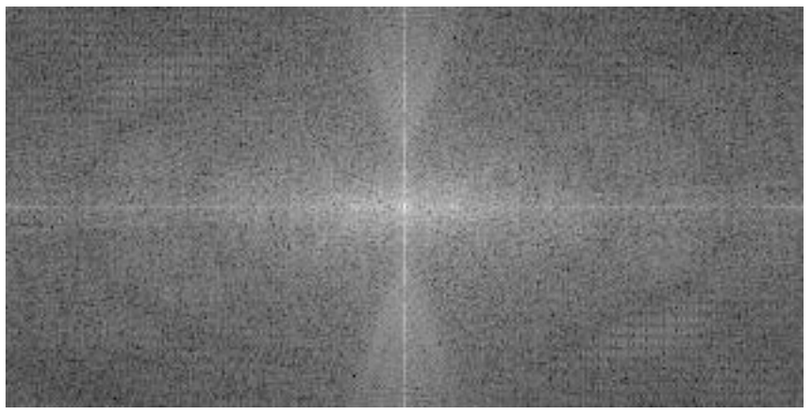




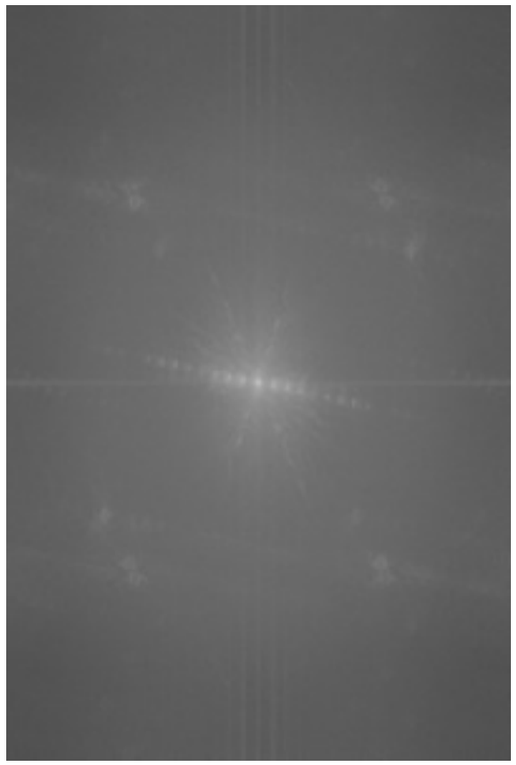
License plates

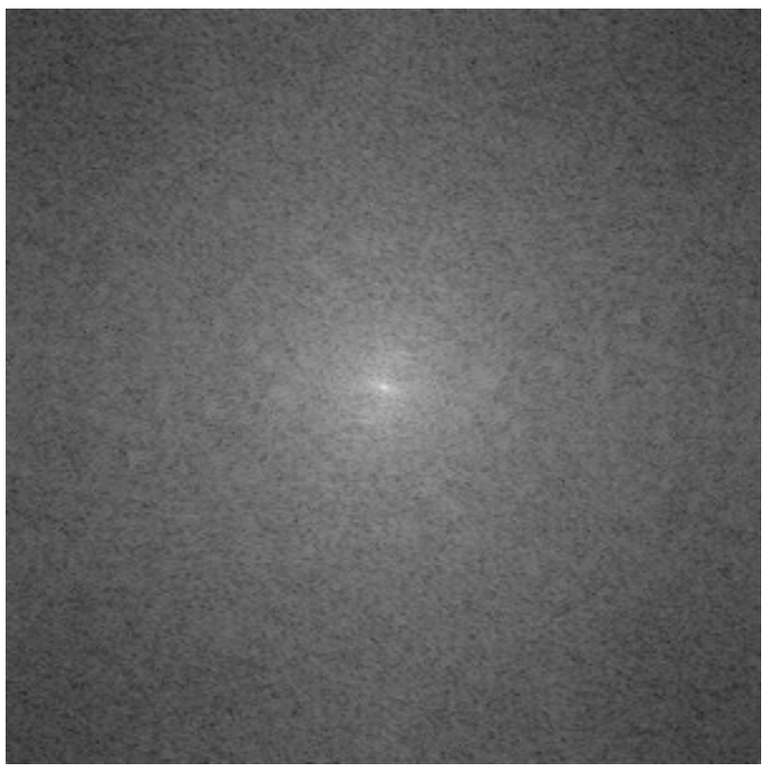




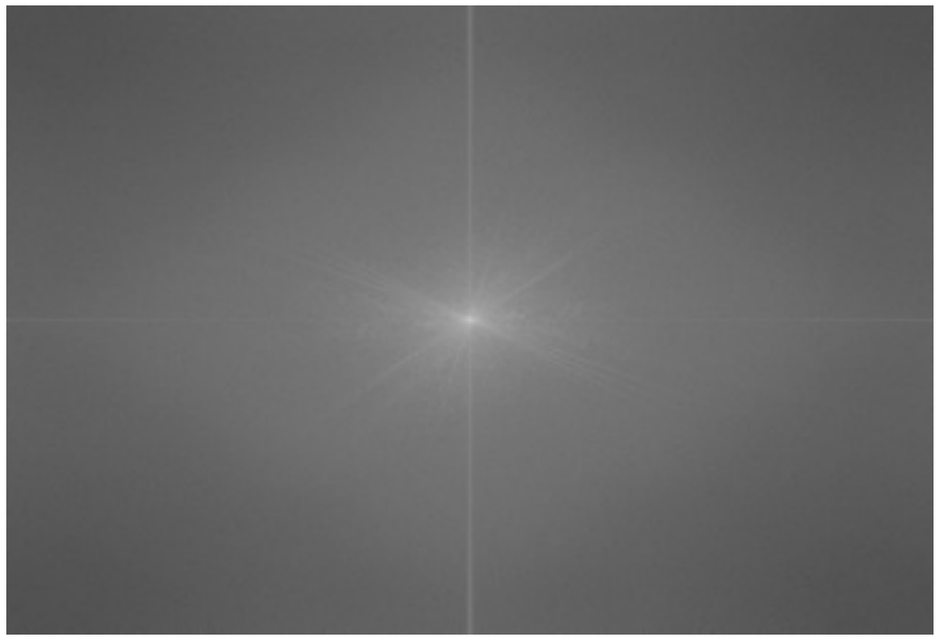


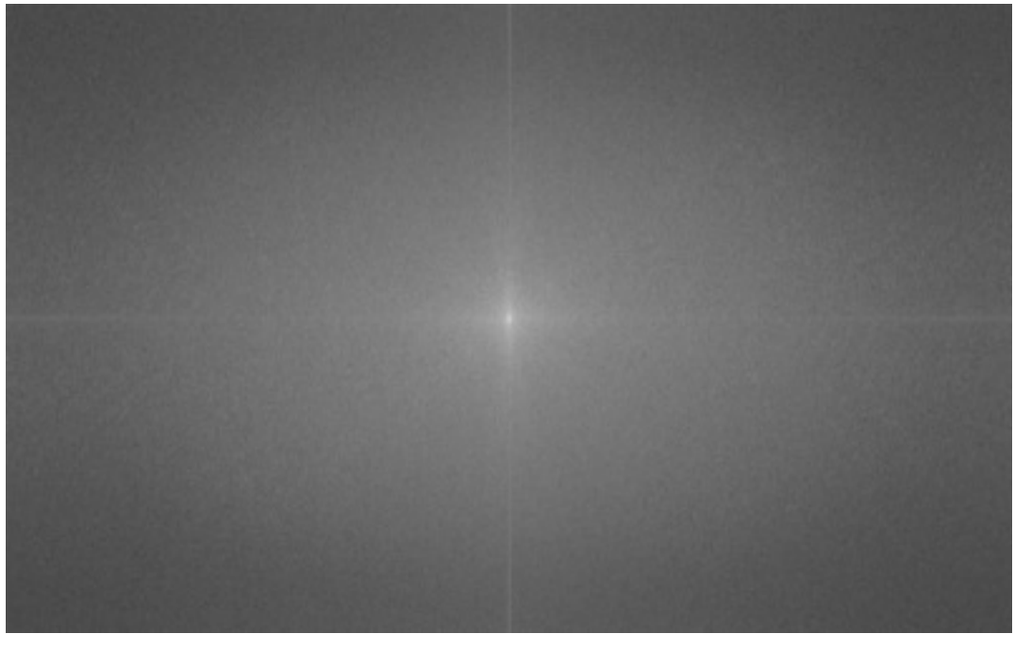
People and Animals





Man-made and natural objects





The Convolution Theorem

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

$$g * h = \sum_k \sum_\ell g(i-k, j-\ell) h(k, \ell)$$

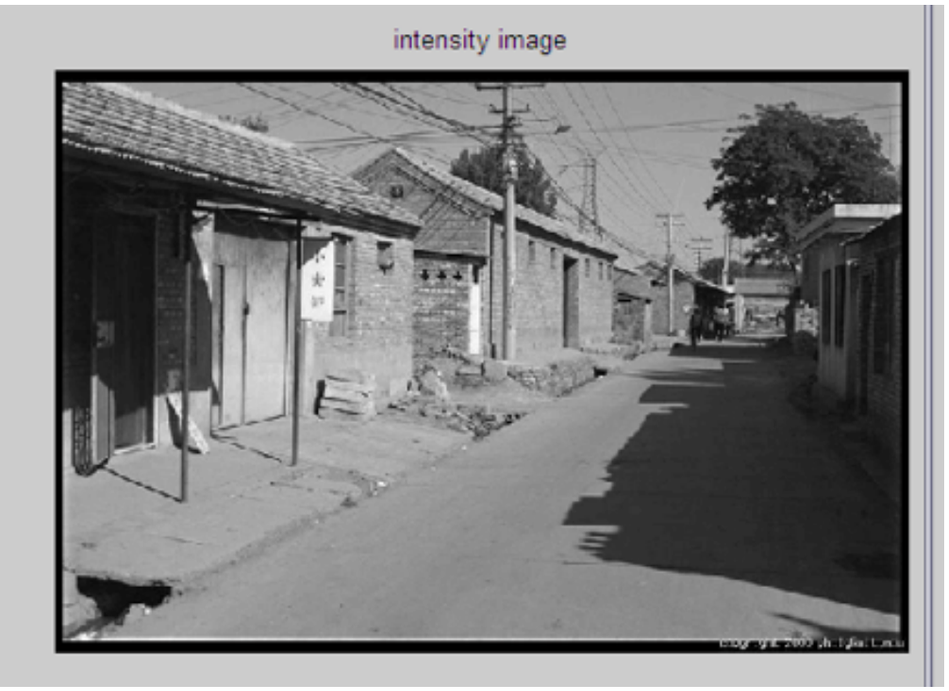
$$\mathcal{F}\{g * h\} = \mathcal{F}\{g\}\mathcal{F}\{h\}$$

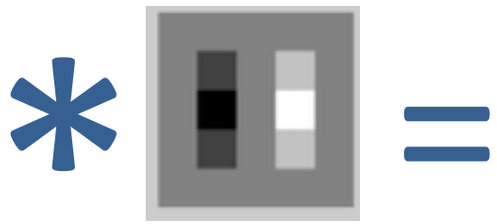
- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!
- For large filter functions h, convolution via the Fourier transform is a lot faster than computing it in the spatial domain

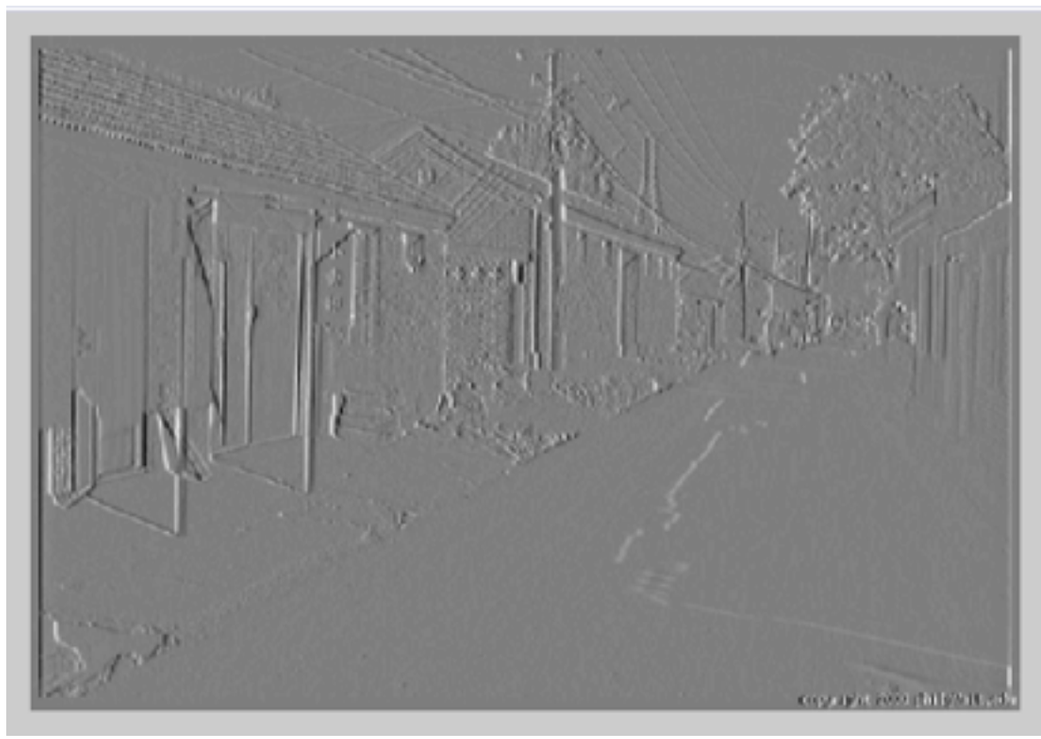
Slide: J. Hays

Filtering in spatial domain

1	0	-1
2	0	-2
1	0	-1

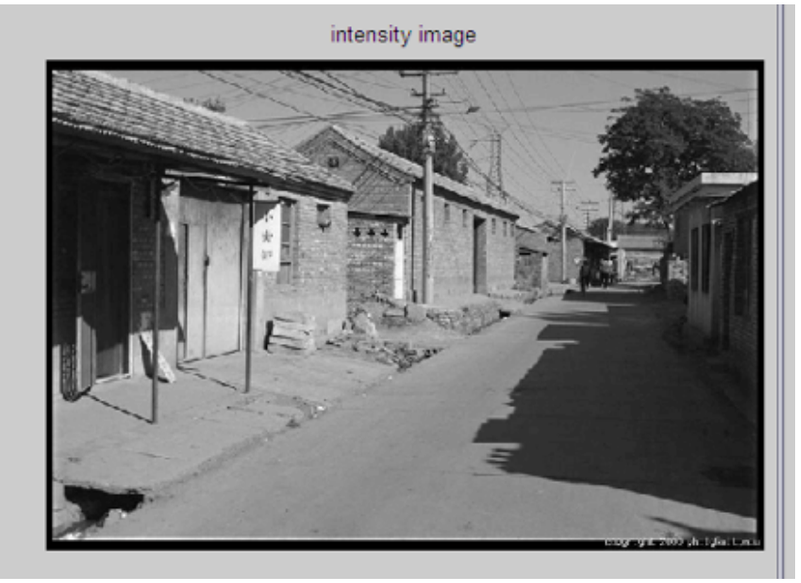




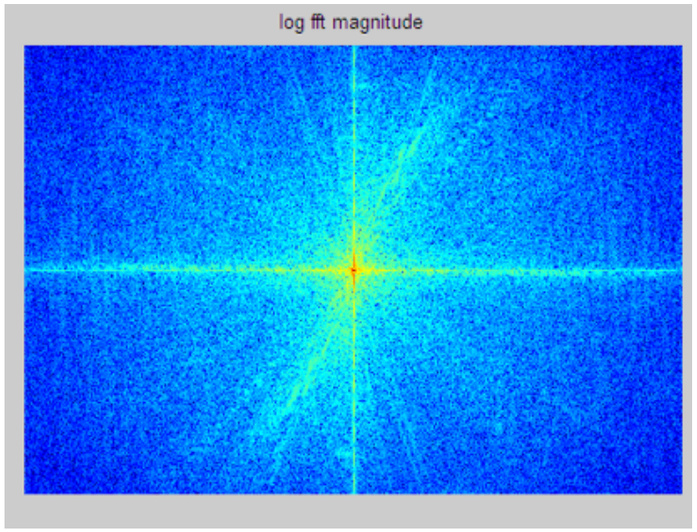


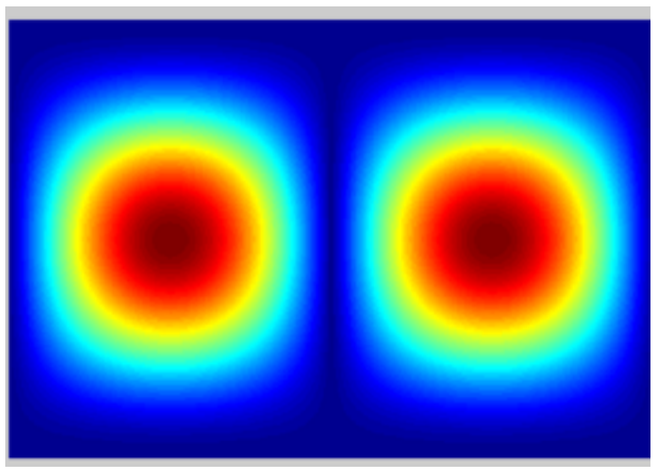
Filtering in frequency domain

FFT

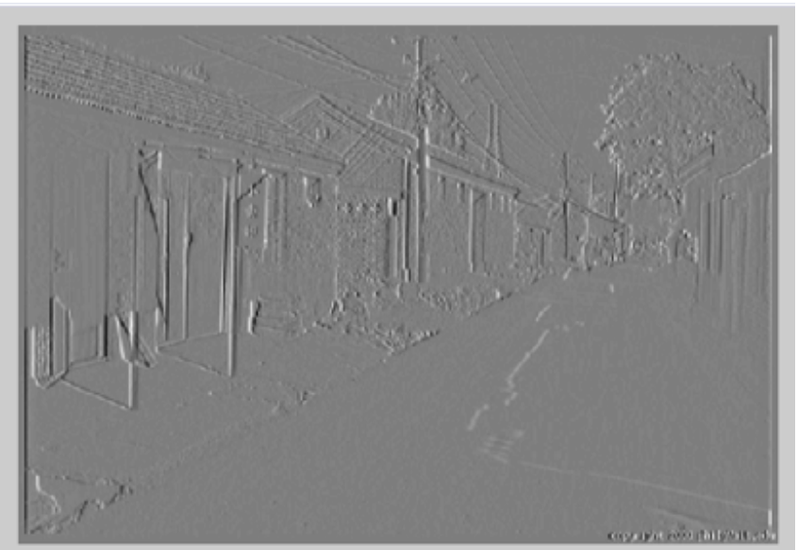


FFT

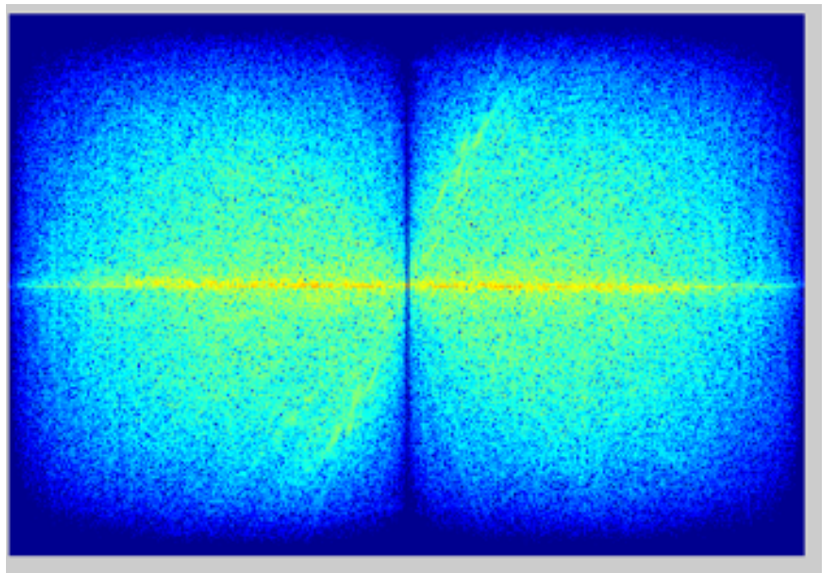




=



Inverse FFT



Slide: Hoiem

Frequency filtering in Matlab

- Filtering with fft

```
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

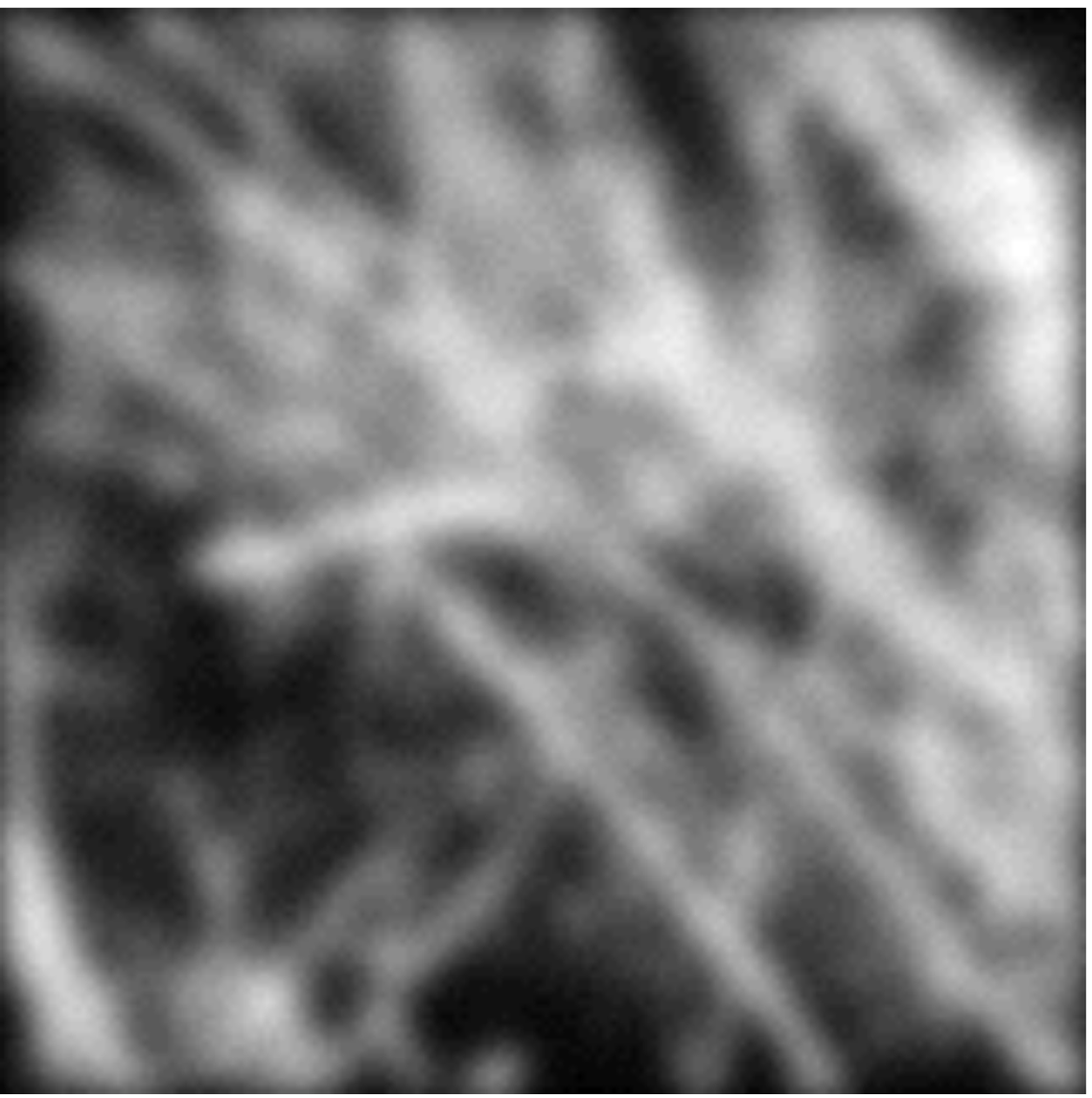
```
figure(1), imagesc(log(abs(fftshift(im_fft)))), axis image, colormap jet
```

Slide: Hoiem

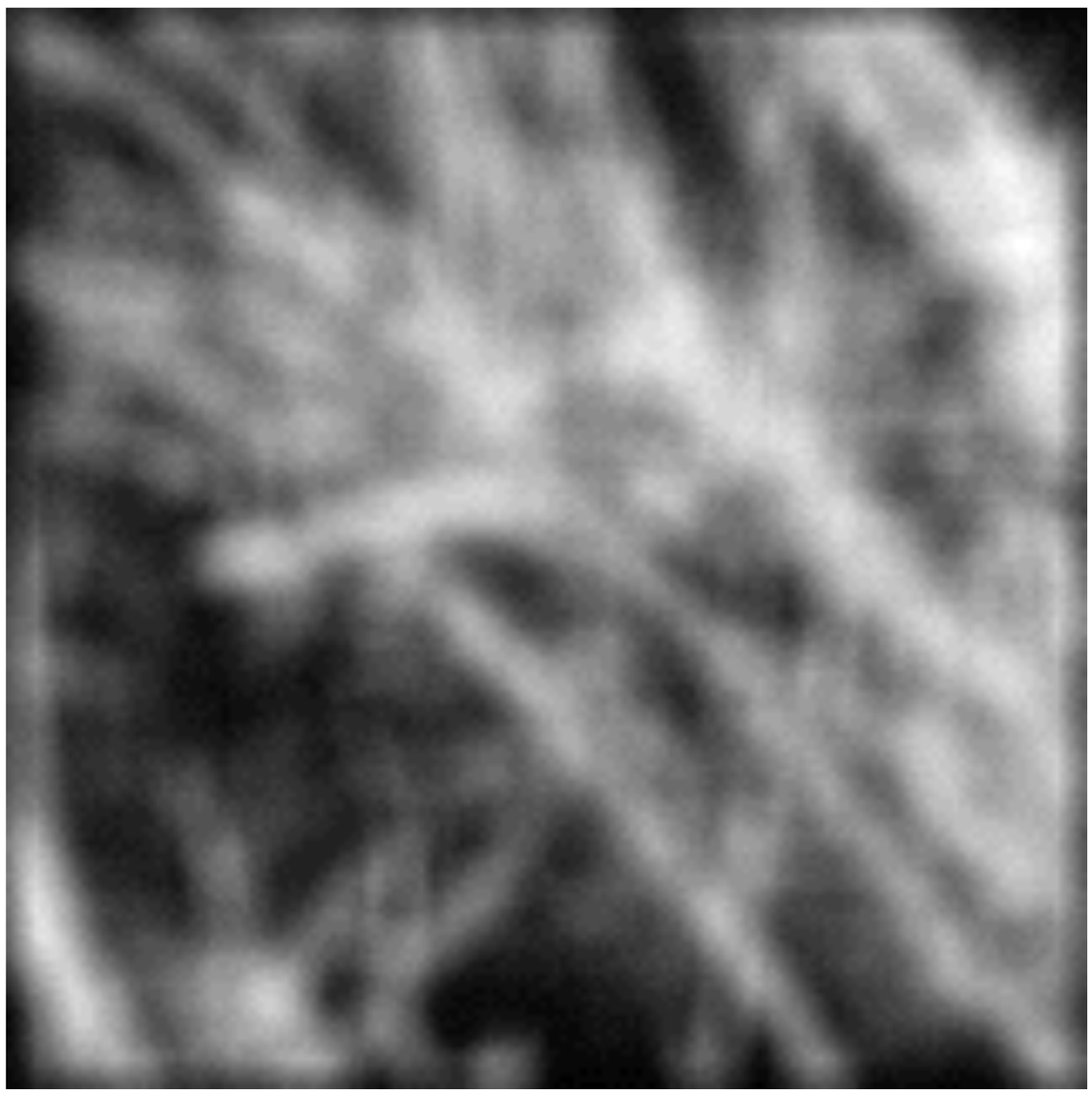
Filtering

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

Gaussian 

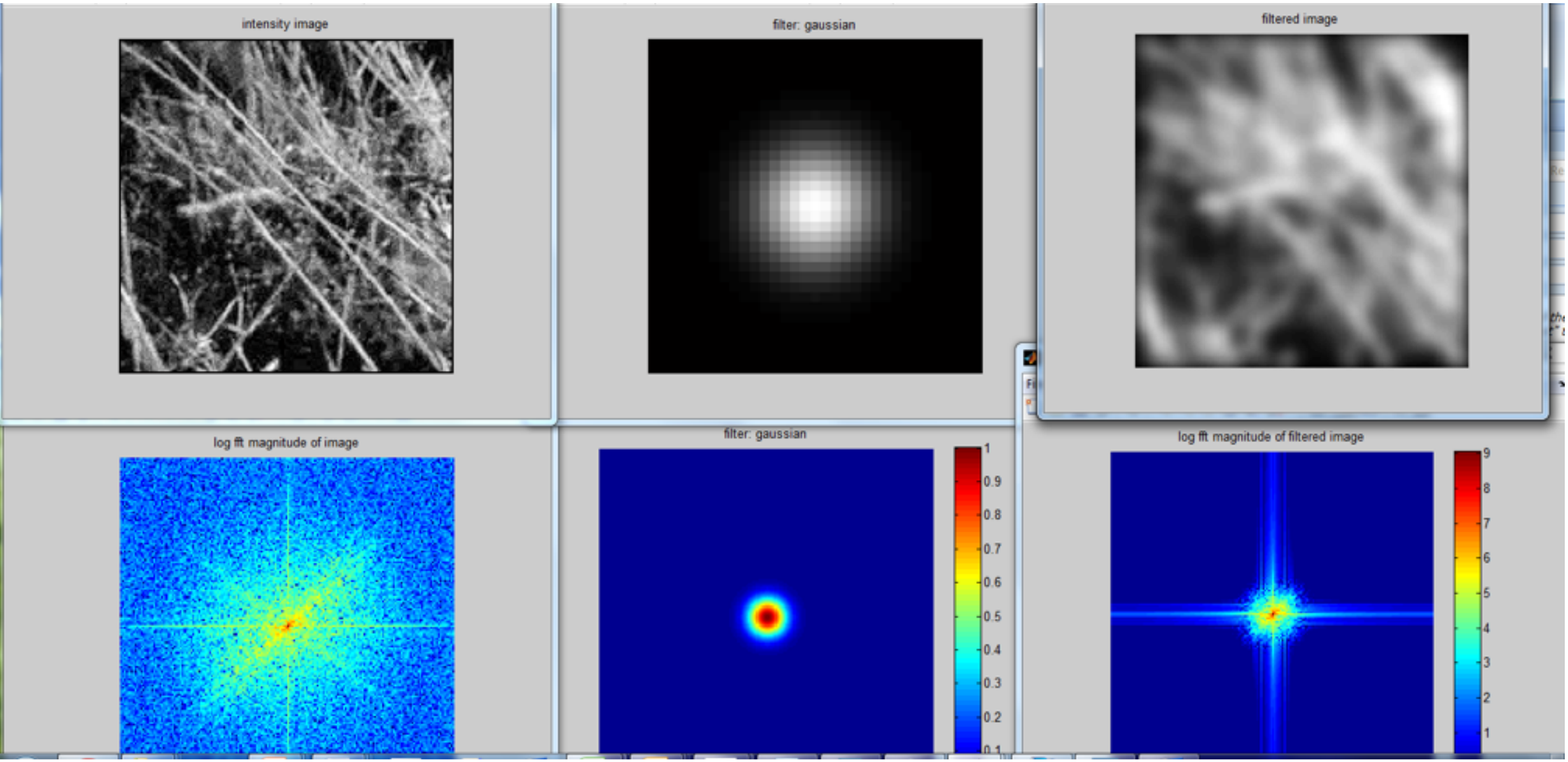


Box filter 



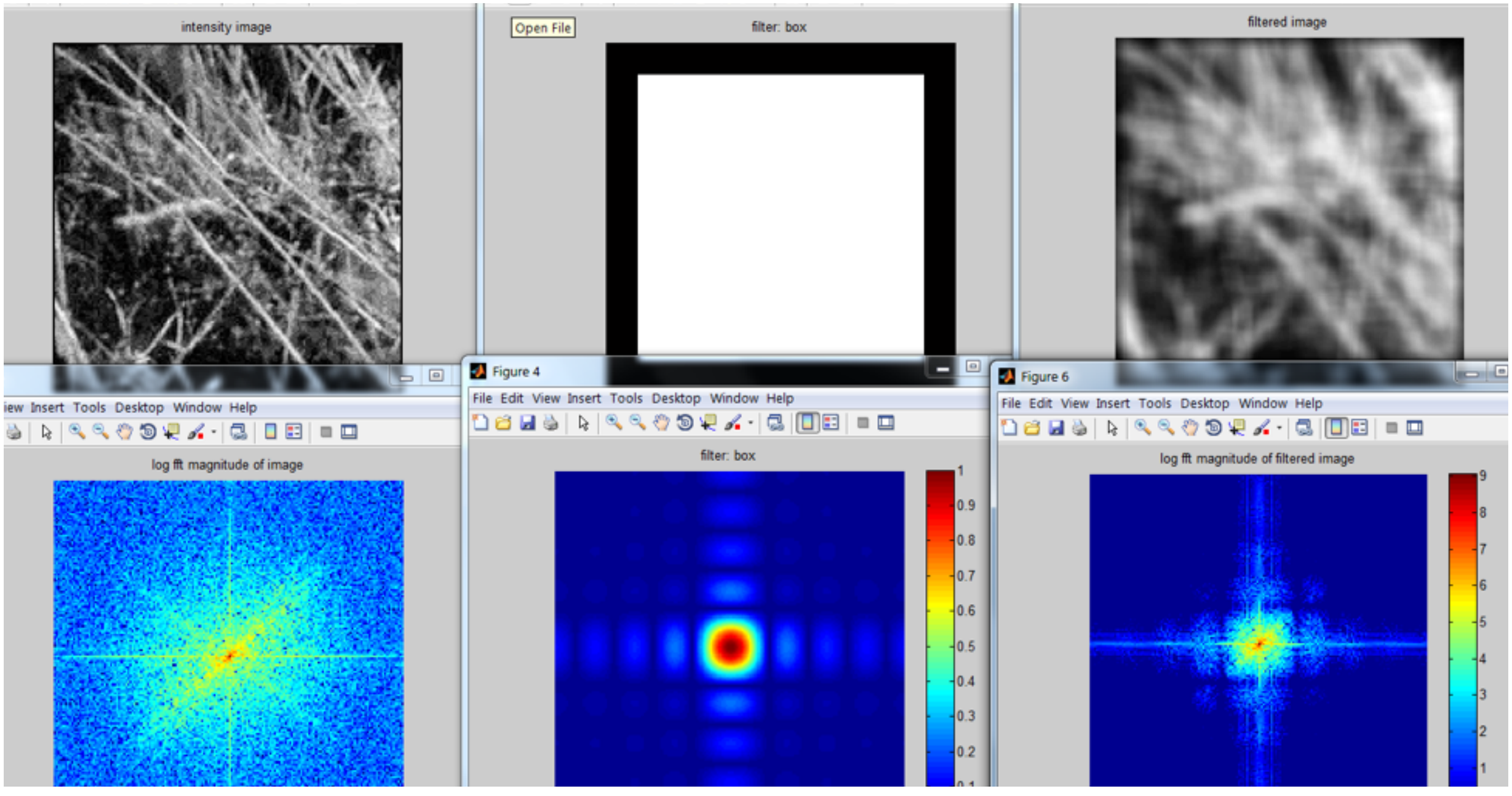
Slide: J. Hays

Gaussian



Slide: J. Hays

Box Filter



Slide: J. Hays

Sampling

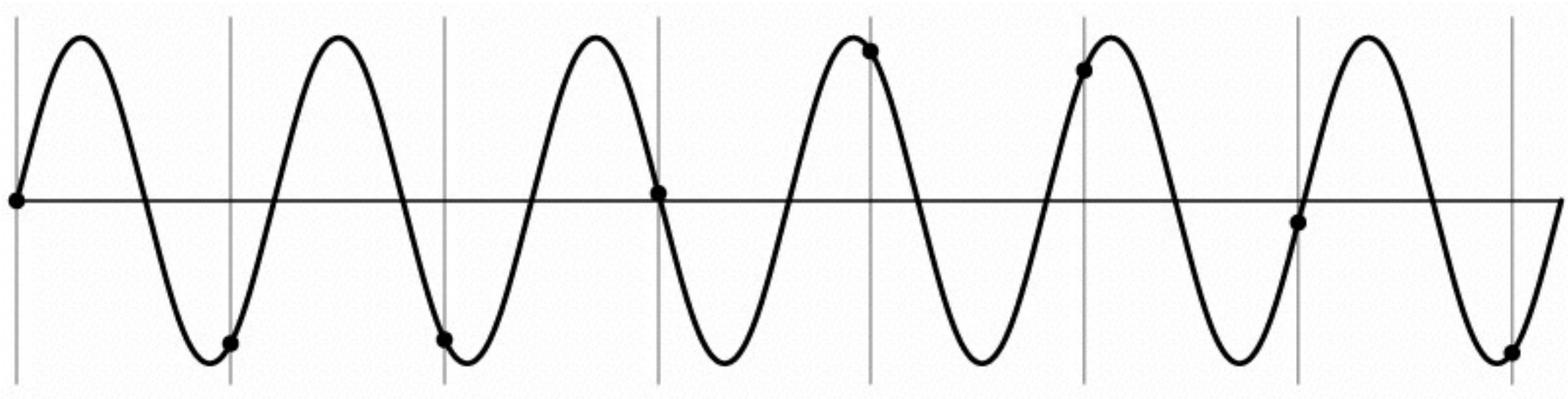
Why does a lower resolution image still make sense to us? What do we lose?

Slide: J. Hays

Image: http://www.flickr.com/photos/igorms/136916757/

Aliasing problem

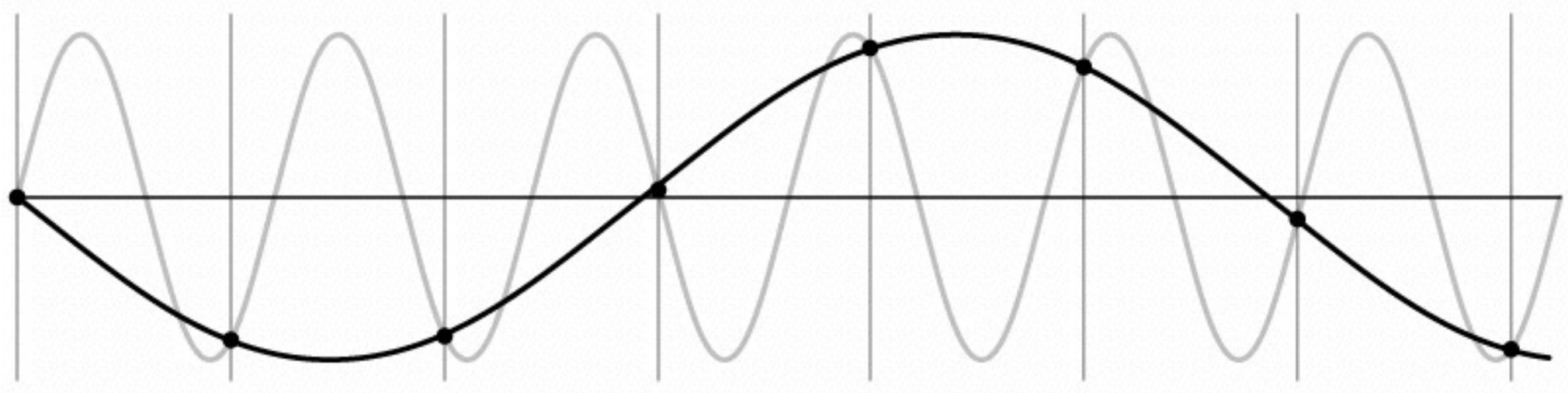
- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

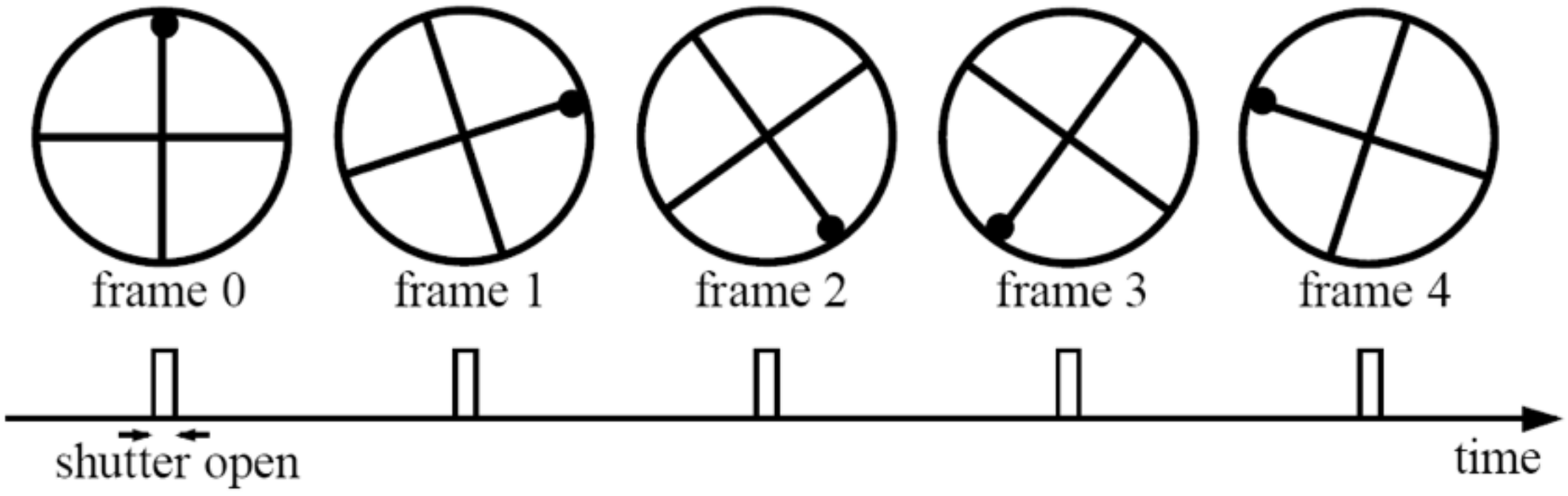
- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - “Wagon wheels rolling the wrong way in movies”
  - “Checkerboards disintegrate in ray tracing”
  - “Striped shirts look funny on color television”

Source: D. Forsyth

Aliasing in video

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

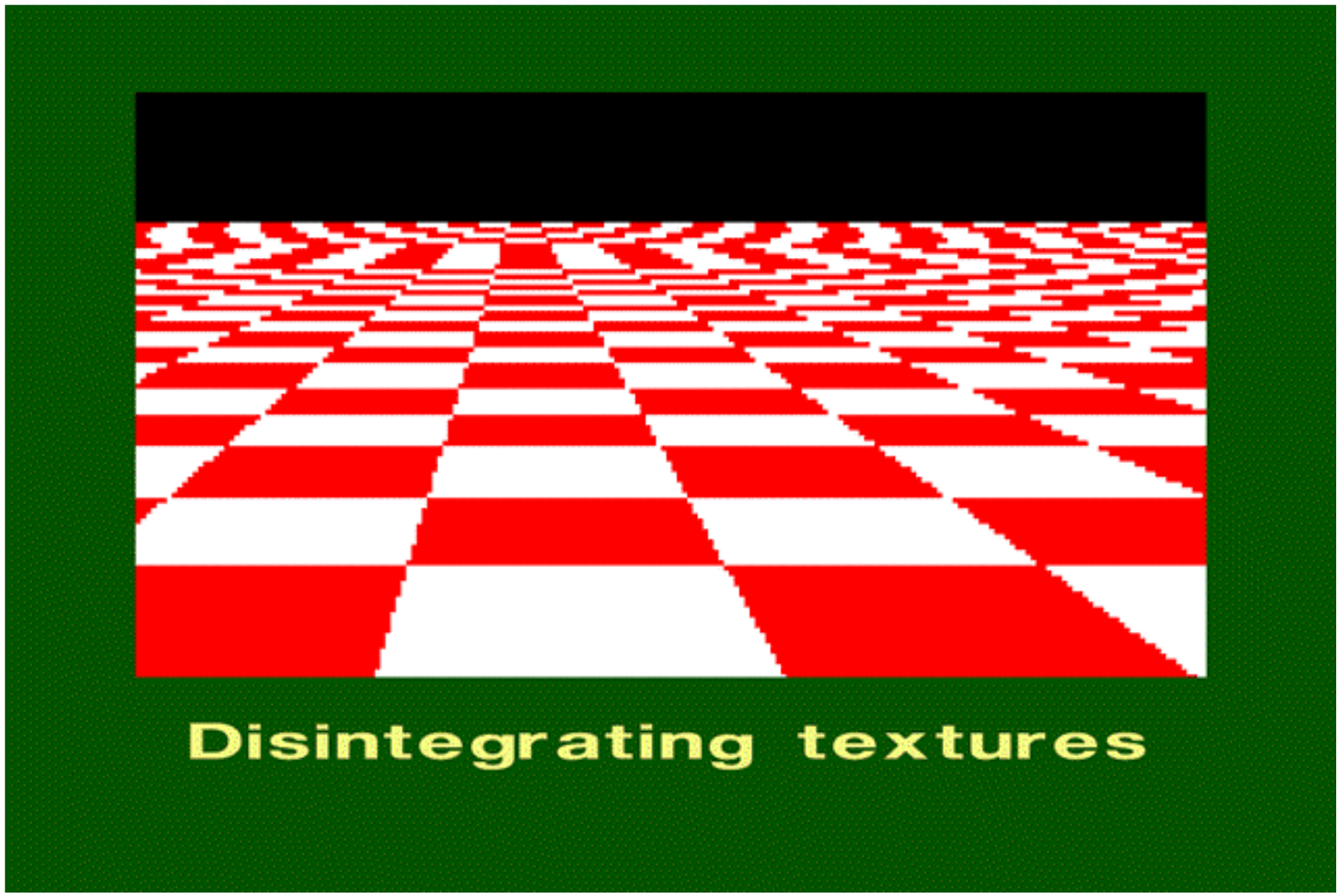
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



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

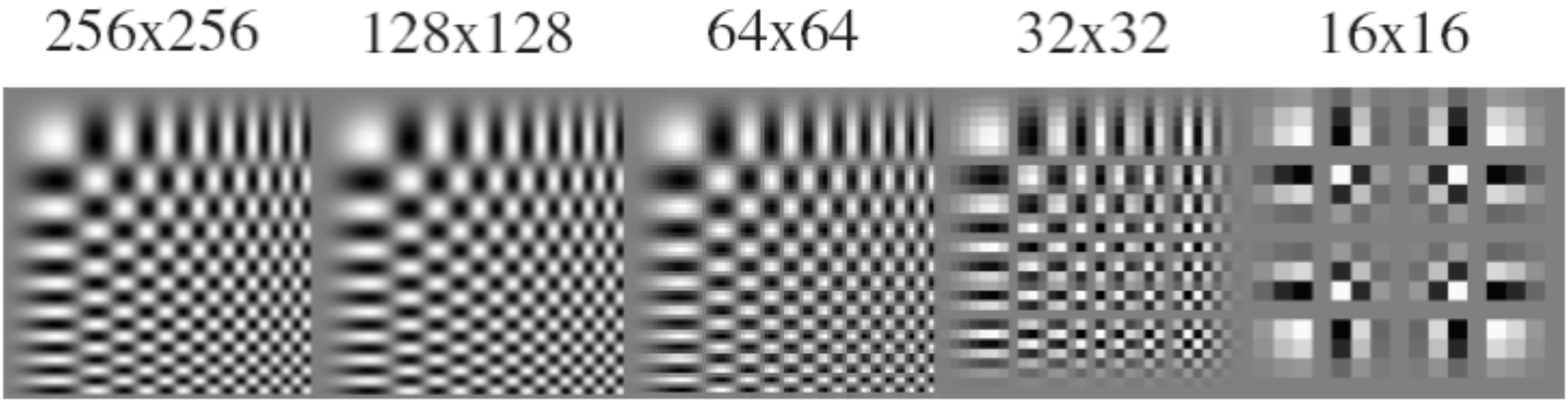
Slide by Steve Seitz

Aliasing in graphics



Source: A. Efros

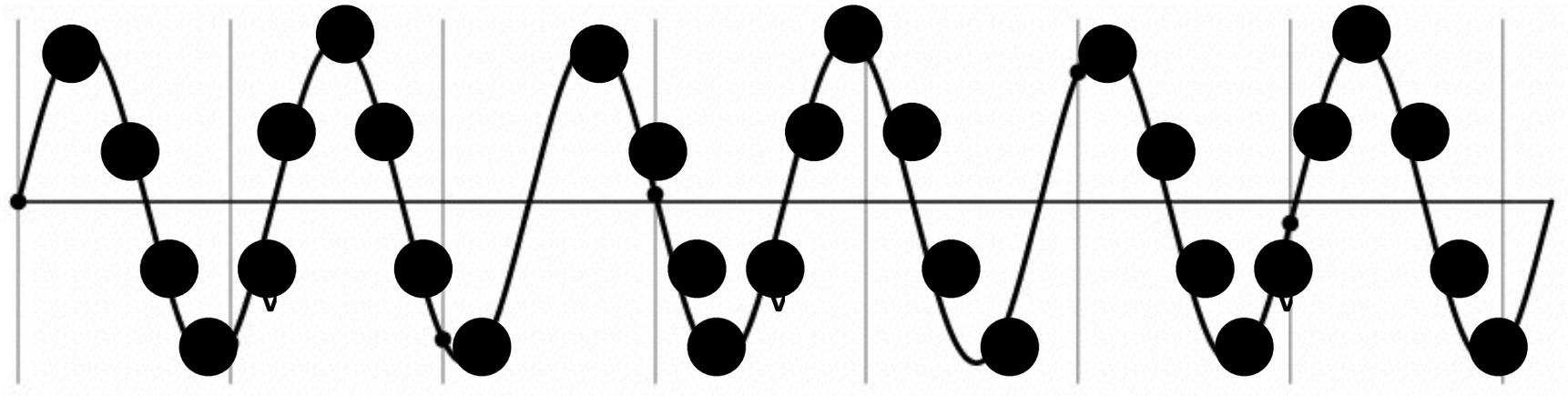
Sampling and aliasing



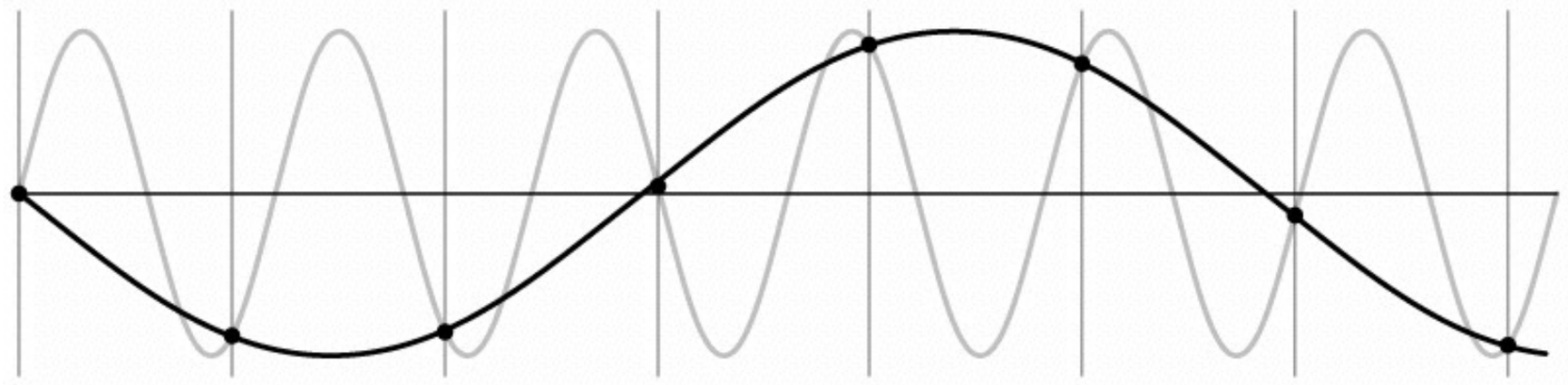
Slide: J. Hays

Nyquist-Shannon Sampling Theorem

- When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{max}$
- f_{max} = max frequency of the input signal
- This will allows to reconstruct the original perfectly from the sampled version



good



bad

Slide: J. Hays

Anti-aliasing

Solutions:

- Sample more often
- Get rid of all frequencies that are greater than half the new sampling frequency
  - Will lose information
  - But it’s better than aliasing
  - Apply a smoothing filter

Slide: J. Hays

Algorithm for downsampling by factor of 2

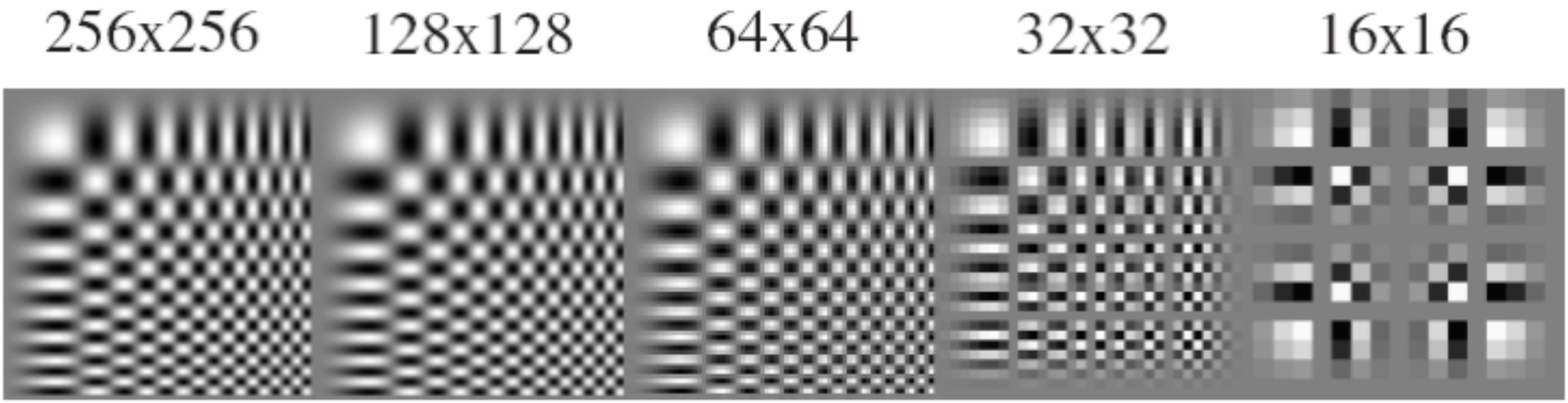
1. Start with image(h, w)
2. Apply low-pass filter

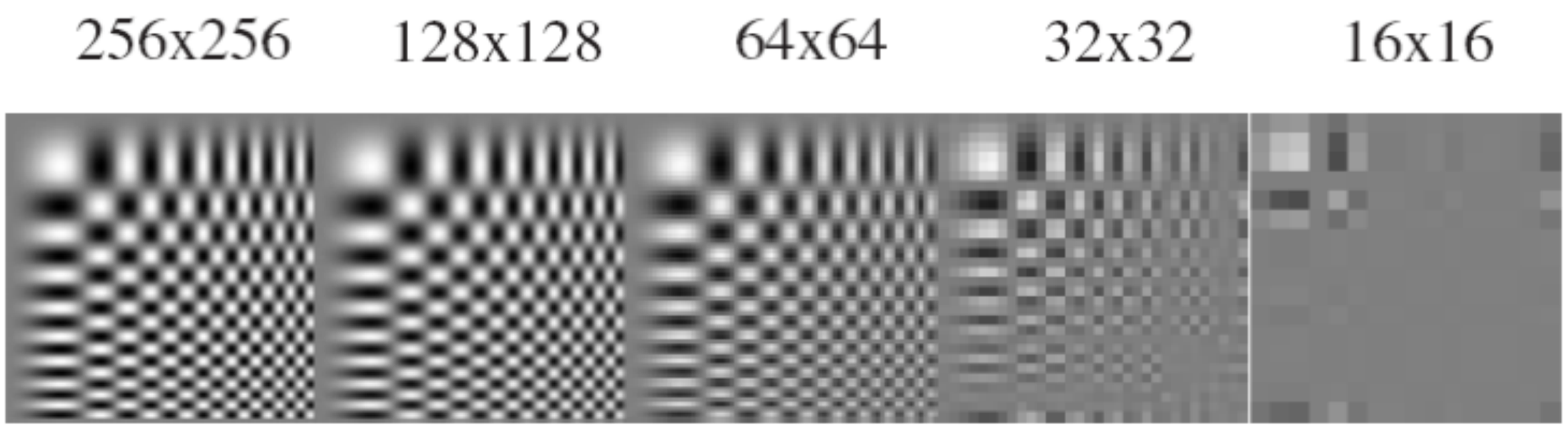
   im_blur = imfilter(image, fspecial('gaussian', 7, 1))
3. Sample every other pixel

   im_small = im_blur(1:2:end, 1:2:end);

Slide: J. Hays

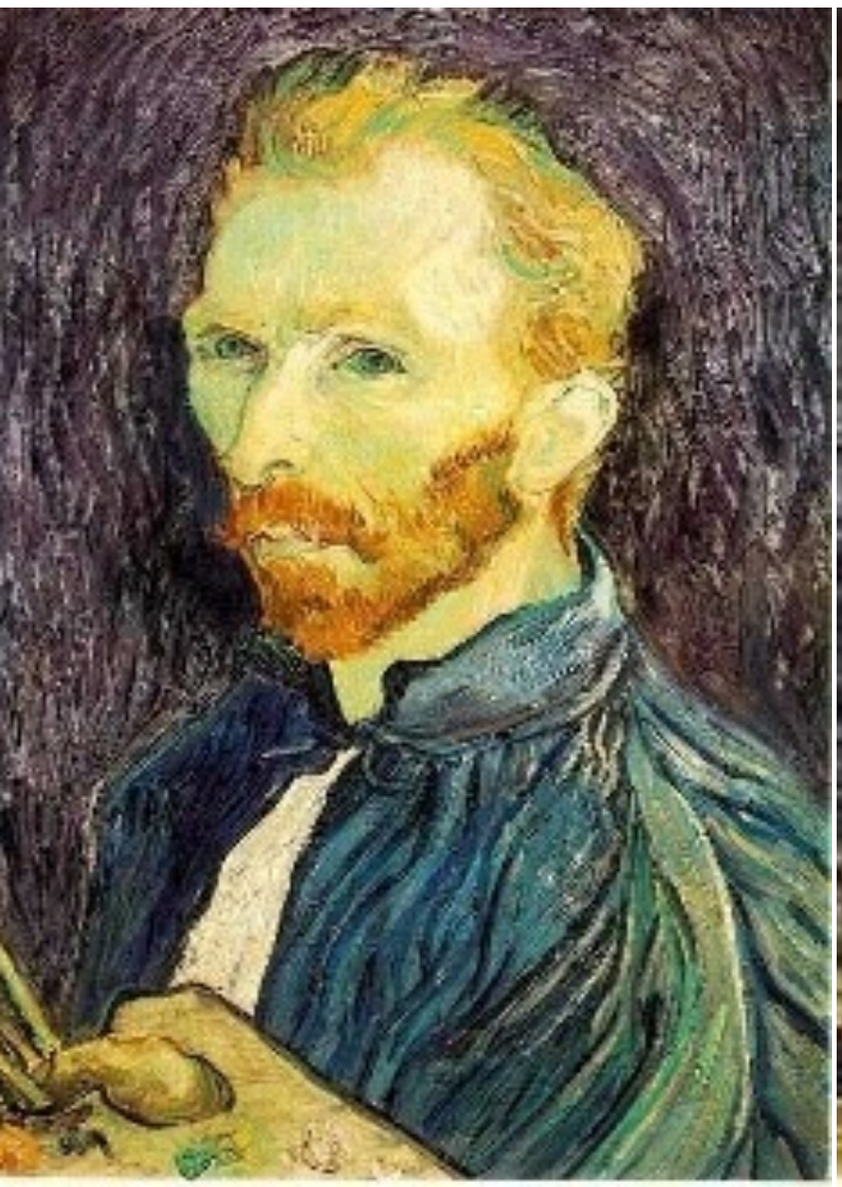
Anti-aliasing





Forsyth and Ponce 2002

Subsampling without pre-filtering



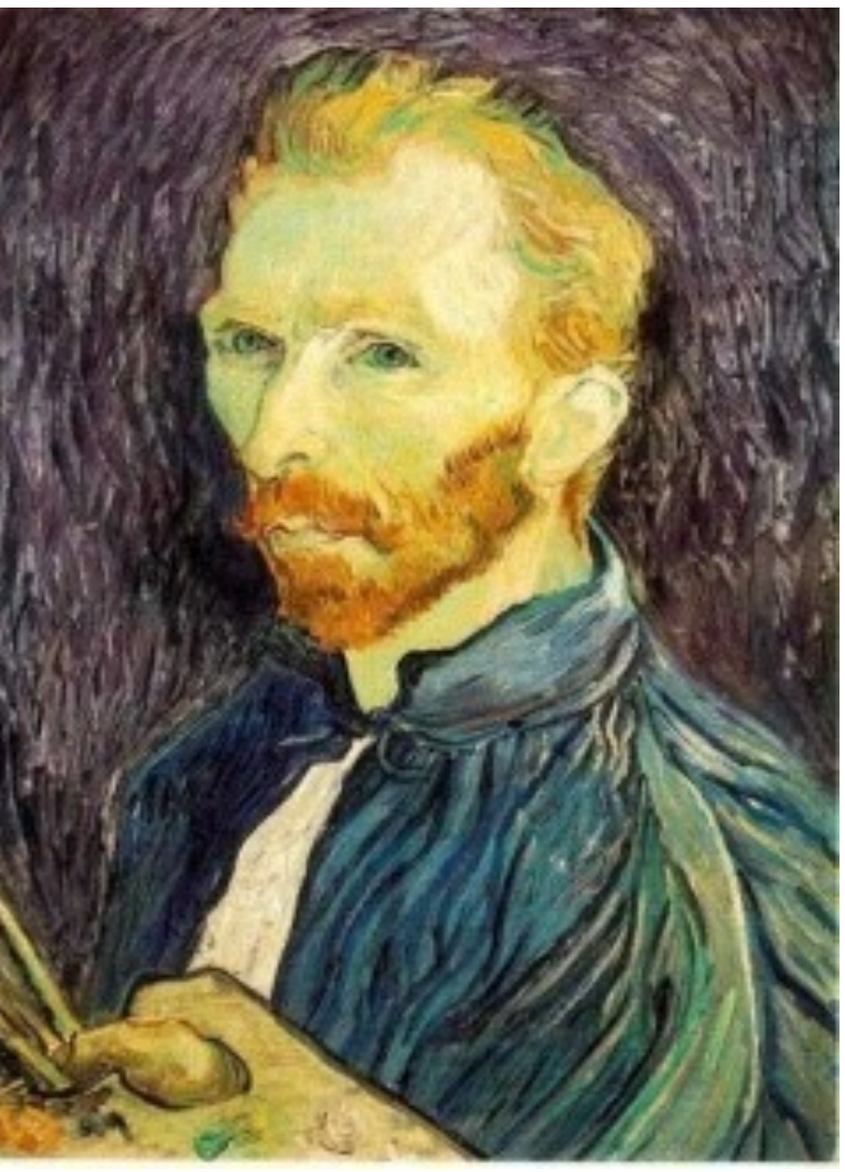
1/2

1/4 (2x zoom)

1/8 (4x zoom)

Slide by Steve Seitz

Subsampling with Gaussian pre-filtering



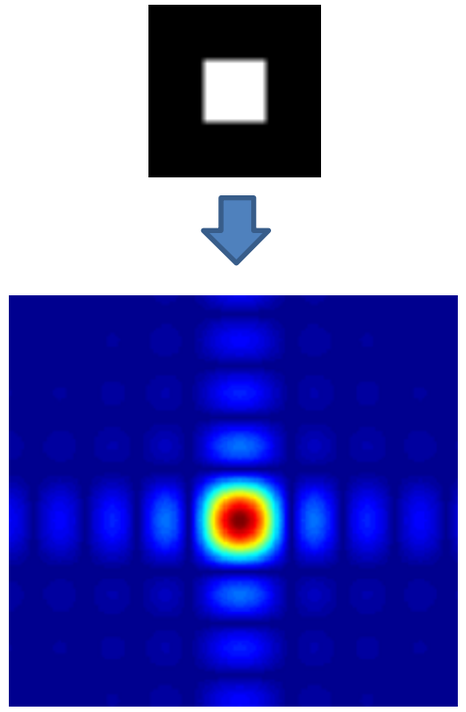
Gaussian 1/2

G 1/4

G 1/8

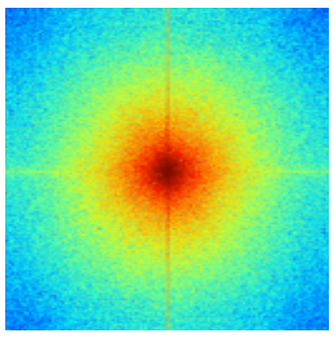
Slide by Steve Seitz

Things to Remember



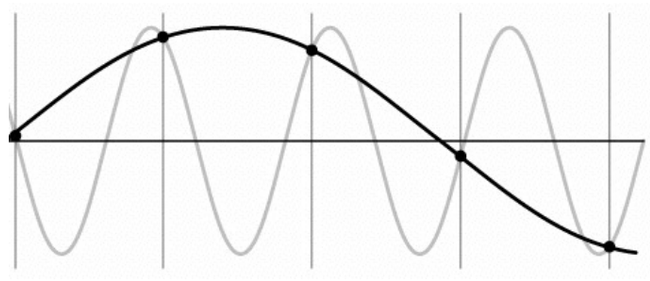
- Sometimes it makes sense to think of images and filtering in the frequency domain
  - Fourier analysis

- Can be faster to filter using FFT for large images (N logN vs. N^2 for auto-correlation)



- Images are mostly smooth
  - Basis for compression

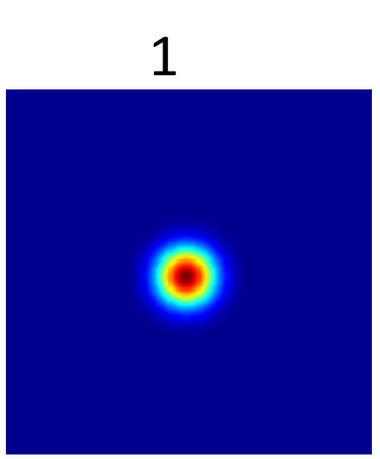
- Remember to low-pass before sampling

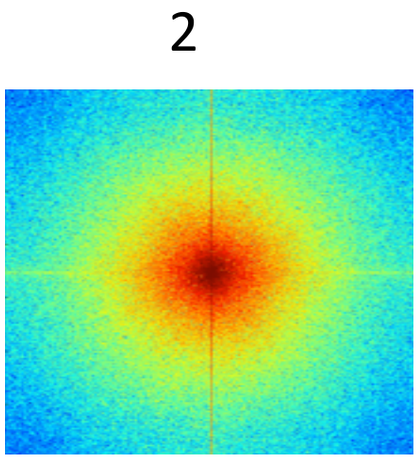


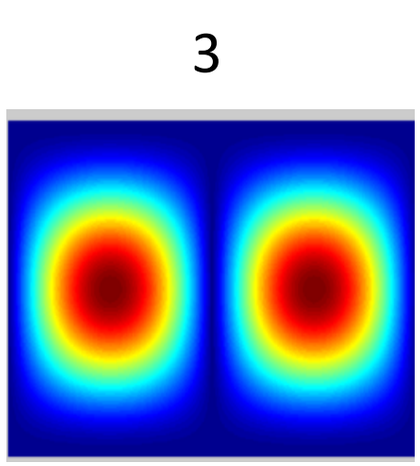
Slide: J. Hays

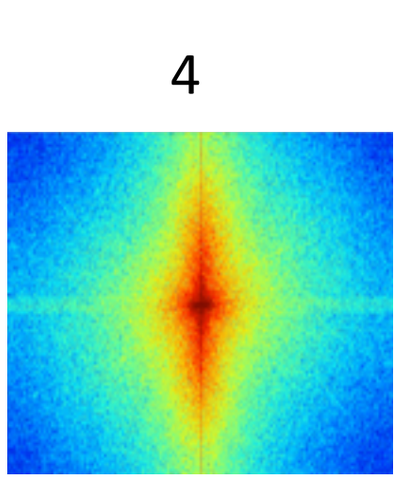
Practice question

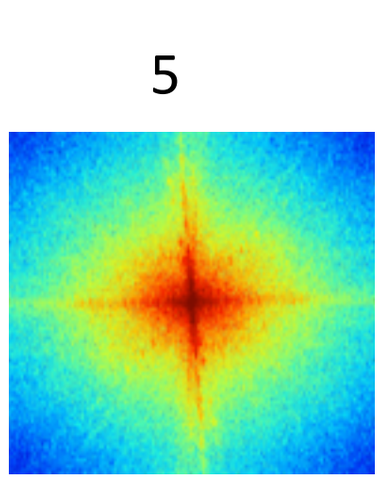
1. Match the spatial domain image to the Fourier magnitude image









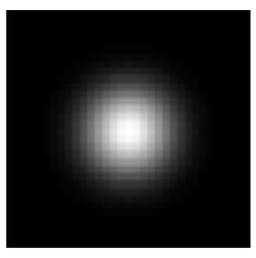


A

B

C

D



E

Slide: J. Hays

Credit:
Slide set developed by J. Hays, Brown University.