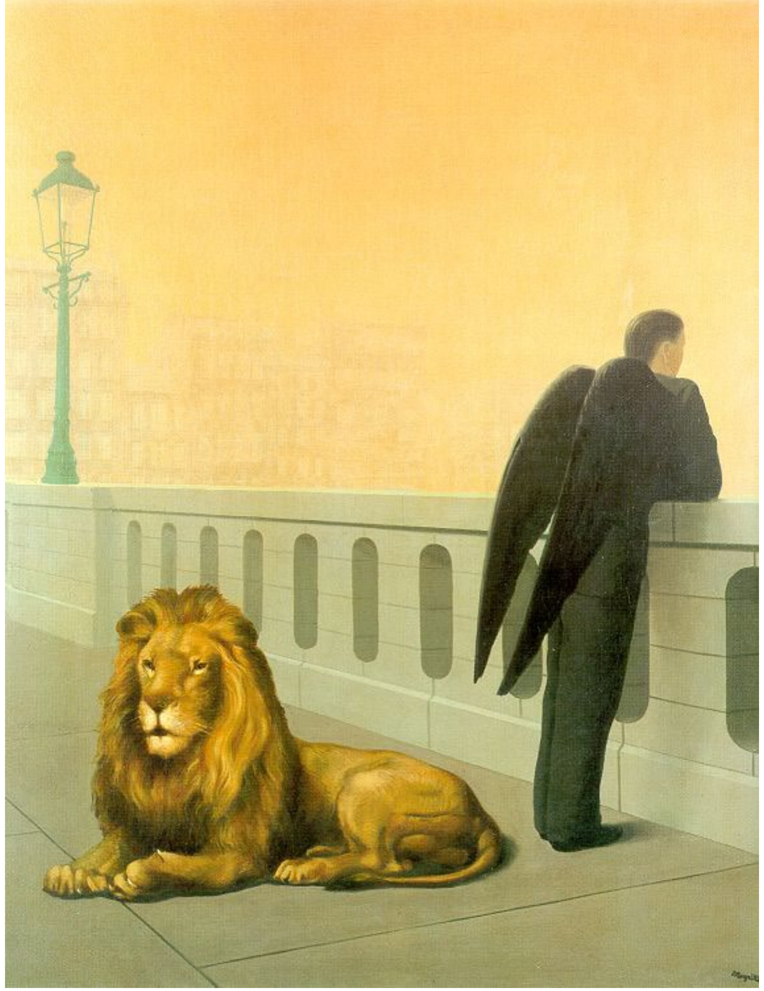
Midterm Review



Magritte, *Homesickness*

Computational Photography

Derek Hoiem, University of Illinois

Major Topics

- Linear Filtering
  - How it works
  - Template and Frequency interpretations
  - Image pyramids and their applications
  - Sampling (Nyquist theorem, application of low-pass filtering)
- Light and color
  - Lambertian shading, shadows, specularities
  - Color spaces (RGB, HSV, LAB)
  - Image-based lighting
- Techniques
  - Finding boundaries: intelligent scissors, graph cuts, where to cut and why
  - Texture synthesis: idea of sampling patches to synthesize, filling order, PatchMatch optimization
  - Compositing and blending: alpha compositing, Laplacian blending, Poisson editing

Major Topics

- Warping
  - Transformation matrices, homogeneous coordinates, solving for parameters via system of linear equations
- Modeling shape
  - Averaging and interpolating sets of points
- Camera models and Geometry
  - Pinhole model: diagram, intrinsic/extrinsic matrices, camera center (or center of projection), image plane
  - Focal length, depth of field, field of view, aperture size
  - Vanishing points and vanishing lines (what they are, how to find them)
  - Measuring relative lengths based on vanishing points and horizon

Preparing for the Exam

1. Review the slides briefly

2. Do the practice questions without looking at anything else

3. Carefully review material related to any that you miss or feel unsure about

Today’s review

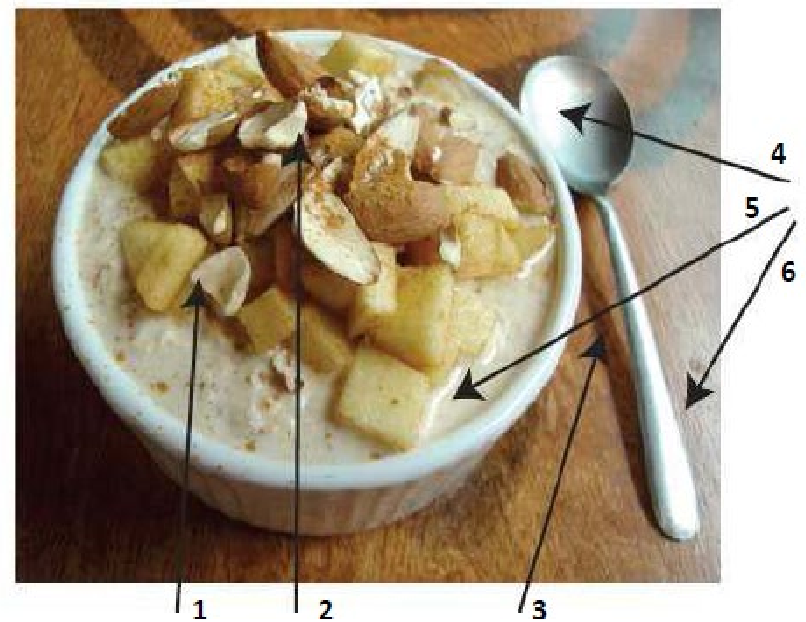
1. Light
2. Camera capture and geometry
3. Image filtering
4. Region selection and compositing
5. Solving for transformations

Purposes

- Remind you of key concepts
- Chance for you to ask questions

1. Light and color

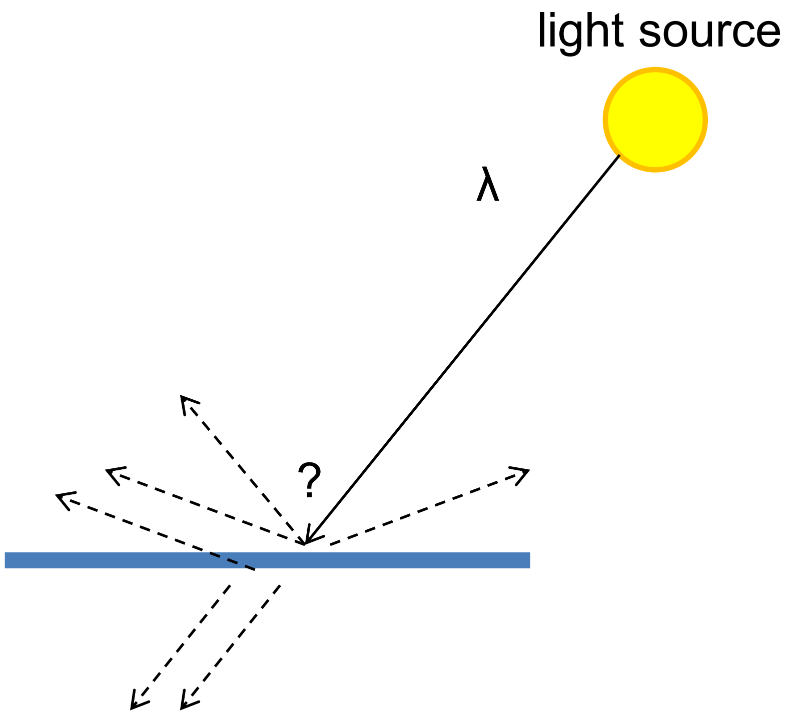
- Lighting
  - Lambertian shading, shadows, specularities
  - Color spaces (RGB, HSV, LAB)



How is light reflected from a surface?

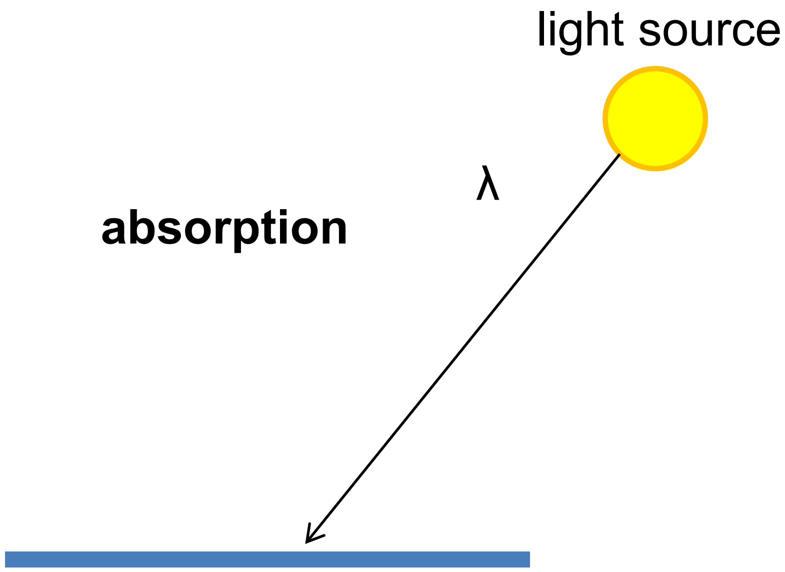
Depends on

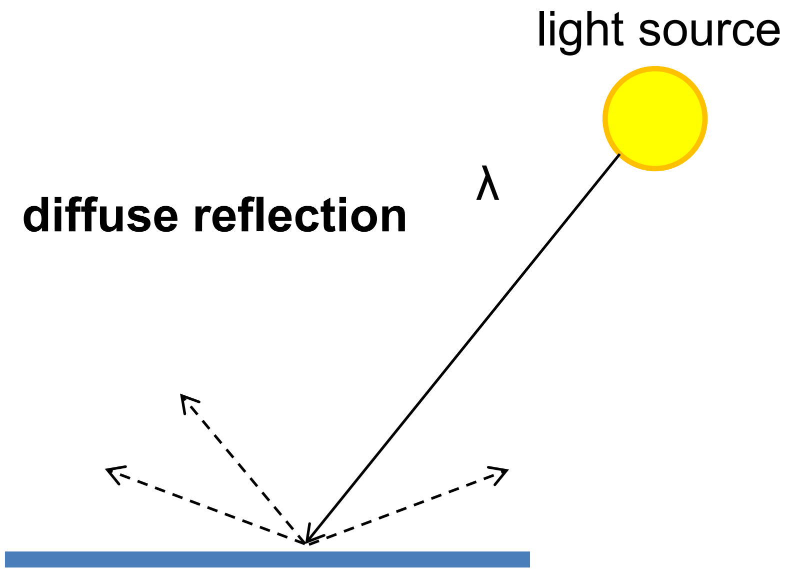
- Illumination properties: wavelength, orientation, intensity
- Surface properties: material, surface orientation, roughness, etc.



Lambertian surface

- Some light is absorbed (function of albedo)
- Remaining light is reflected equally in all directions (diffuse reflection)
- Examples: soft cloth, concrete, matte paints





Diffuse reflection

Intensity does depend on illumination angle because less light comes in at oblique angles.

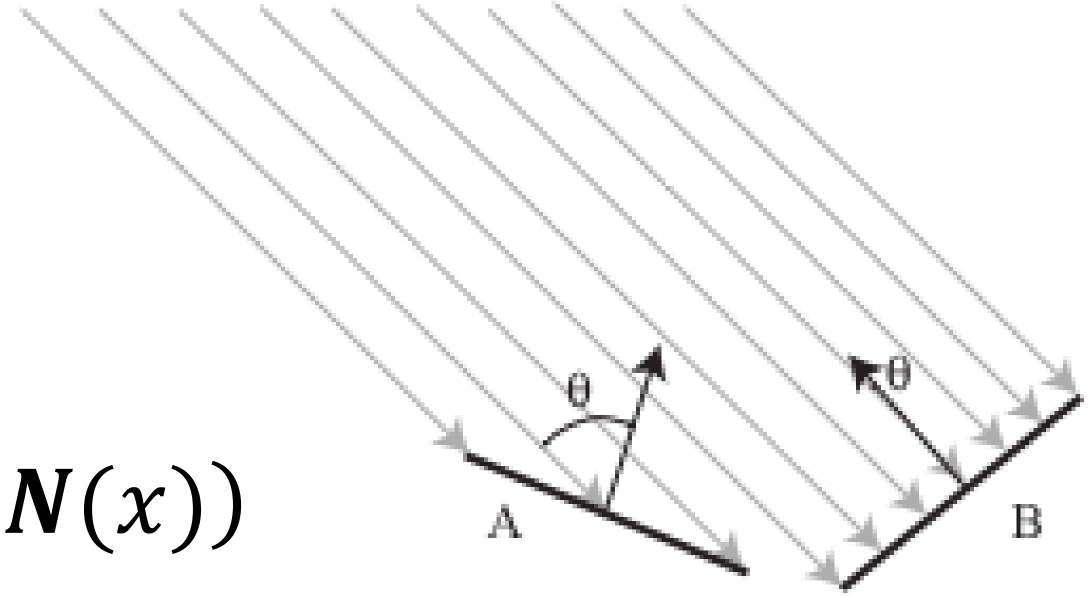
$\rho =$ albedo

$\boldsymbol{S} =$ directional source

$\boldsymbol{N} =$ surface normal

$\mathrm{I} =$ image intensity

$$I(x) = \rho(x)(\boldsymbol{S} \cdot \boldsymbol{N}(x))$$



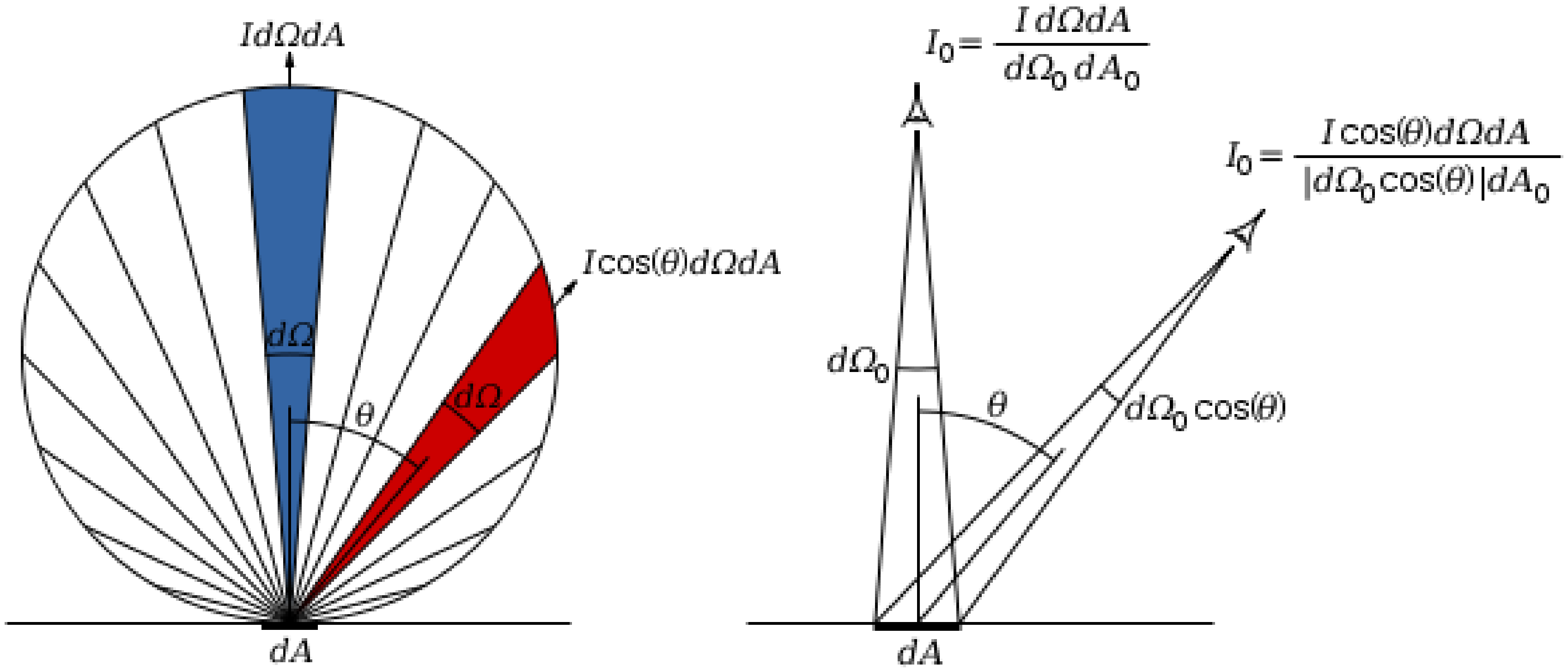
Slide: Forsyth


1
2

Diffuse reflection

Intensity does not depend on viewer angle.

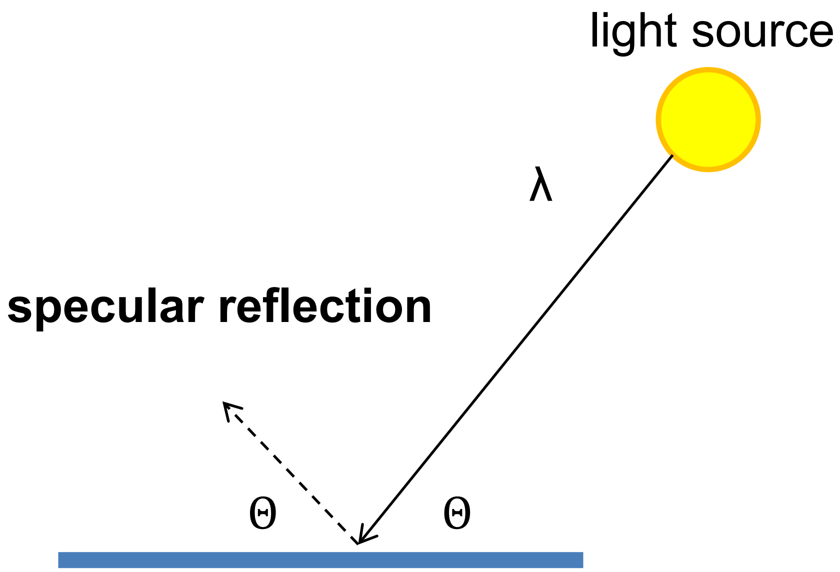
- Amount of reflected light proportional to cos(theta)
- Visible solid angle also proportional to cos(theta)



http://en.wikipedia.org/wiki/Lambert%27s_cosine_law

Specular Reflection

- Reflected direction depends on light orientation and surface normal
- E.g., mirrors are mostly specular



Flickr, by suzysputnik

Flickr, by piratejohnny

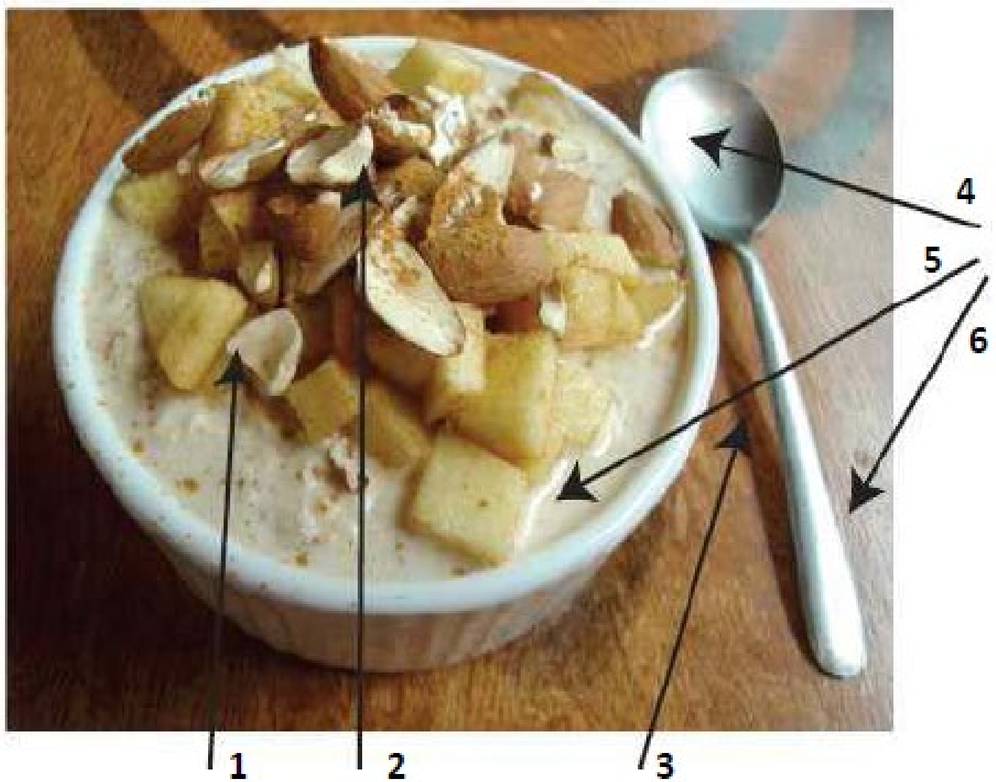
Many surfaces have both specular and diffuse components

- Specularity = spot where specular reflection dominates (typically reflects light source)

Photo: northcountryhardwoodfloors.com

Questions

1.

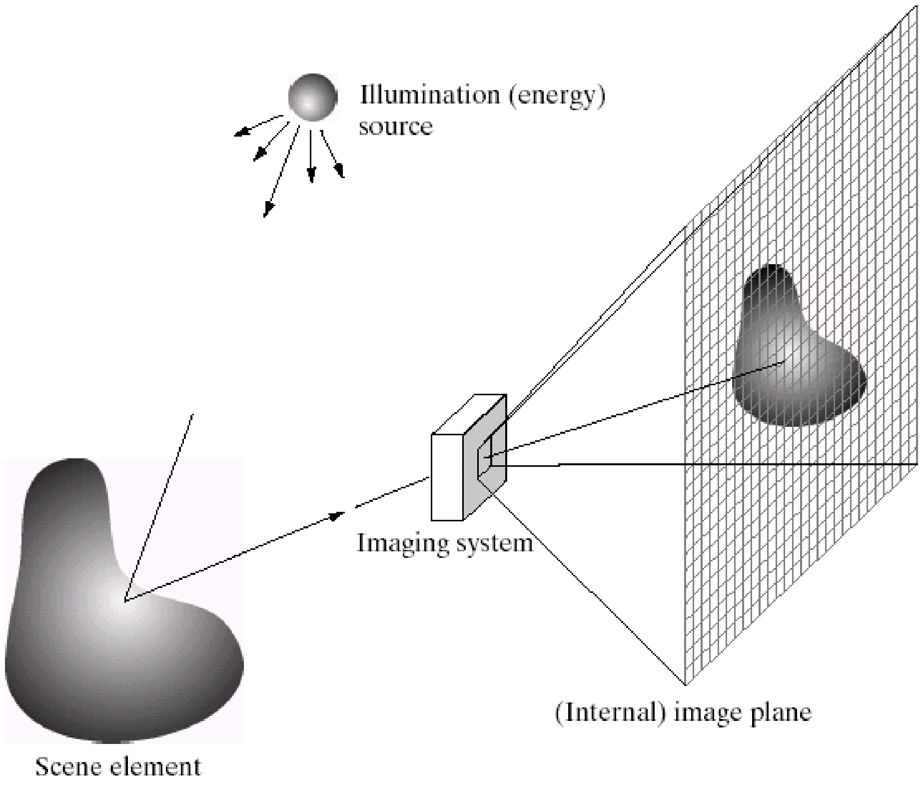


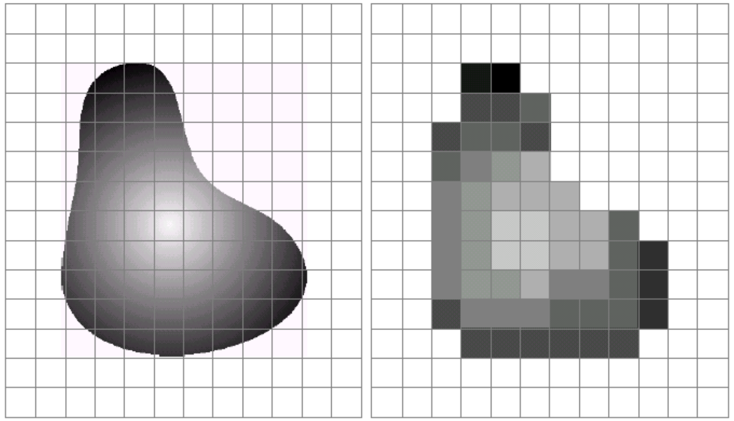
A. For each of the arrows in the above image, name the reasons the pixel near the end of the arrow has its brightness value and explain very briefly. The arrow pointing to milk is pointing to the thin bright line at the edge of the piece of apple; the arrow pointing to the spoon handle is pointing to the bright area on the handle.

Possible factors: albedo, shadows, texture, specularities, curvature, lighting direction

Discretization

- Because pixel grid is discrete, pixel intensities are determined by a range of scene points

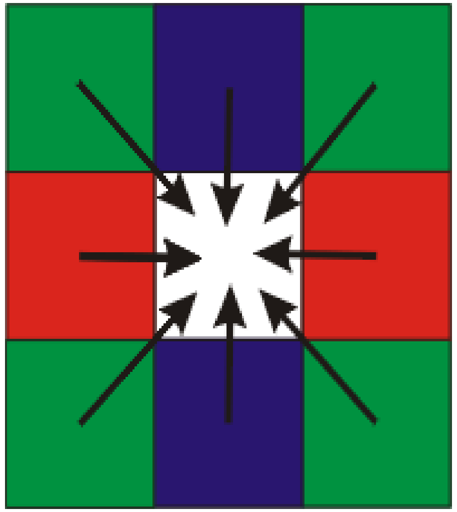




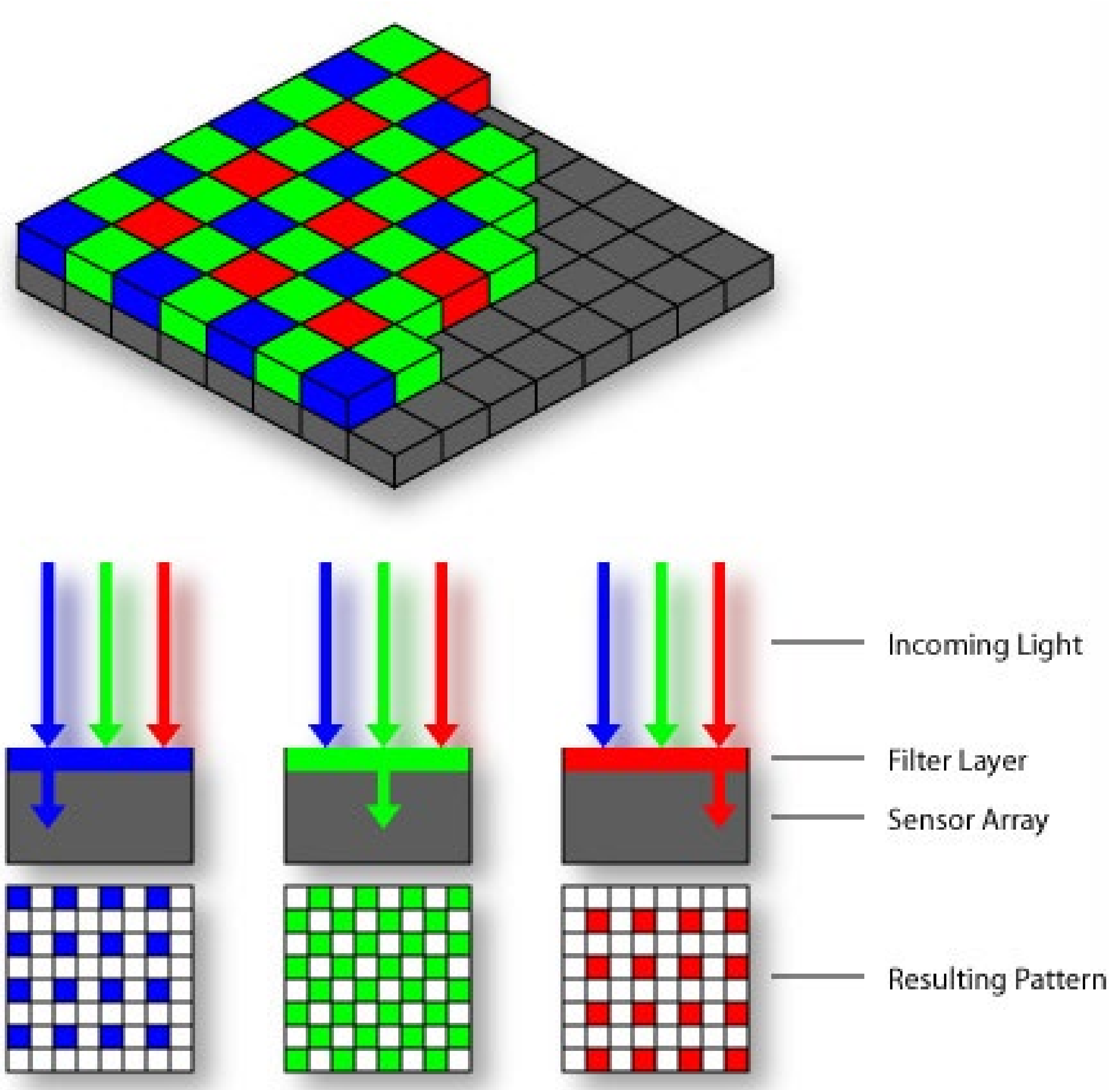
a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

Color Sensing: Bayer Grid



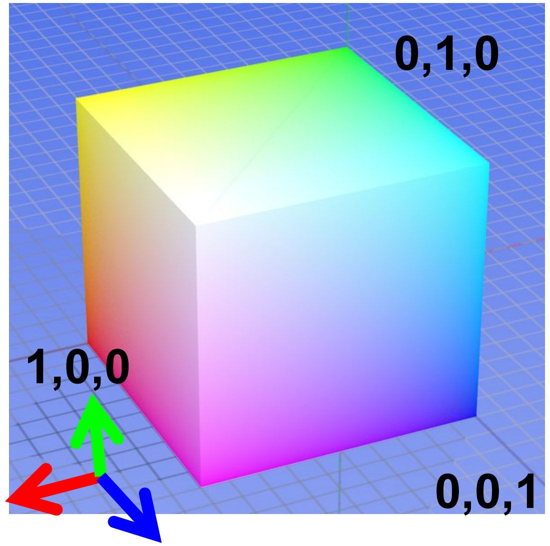
Estimate RGB at each cell from neighboring values



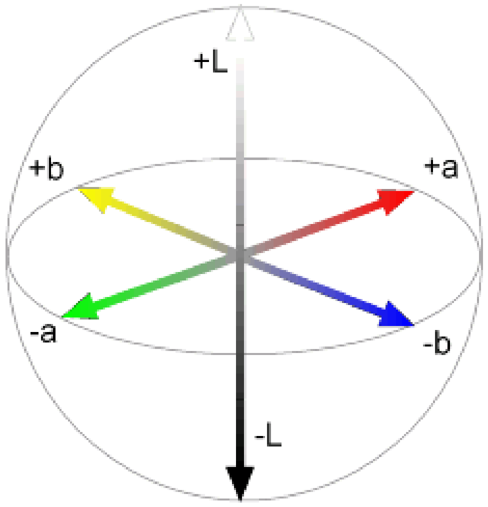
http://en.wikipedia.org/wiki/Bayer_filter

Color spaces

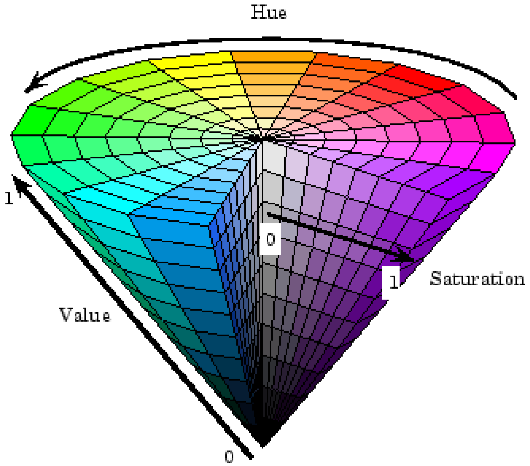
RGB



LAB



HSV



YCbCr

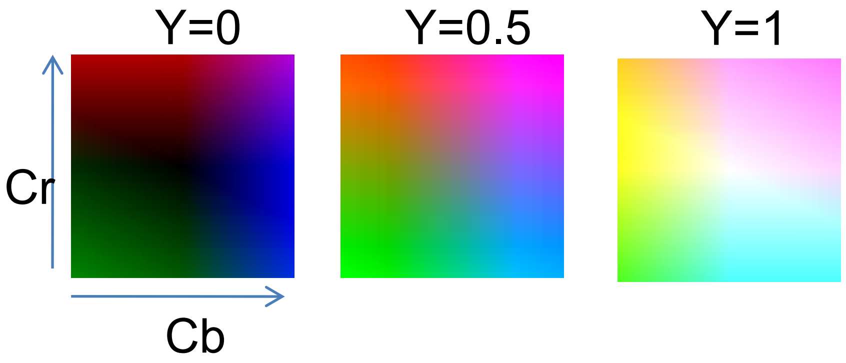
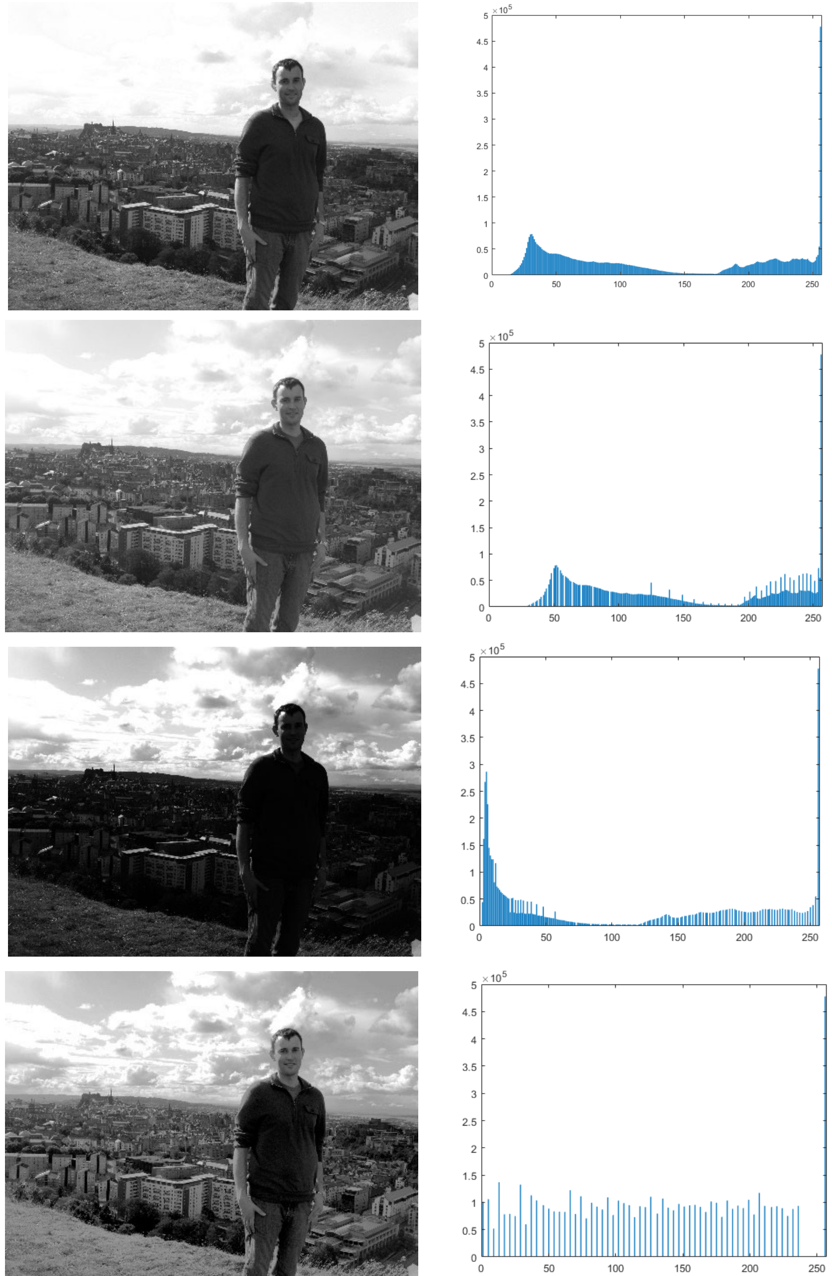
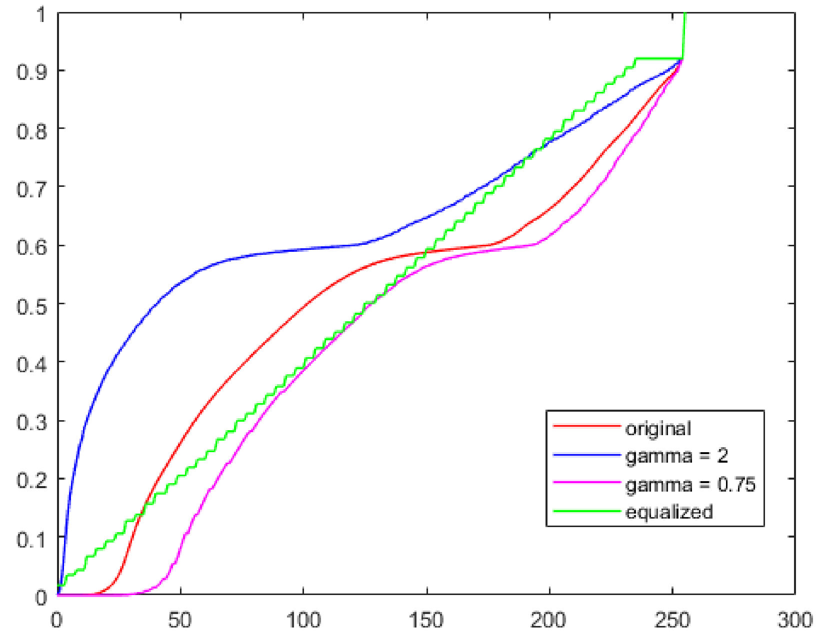


Image Histograms

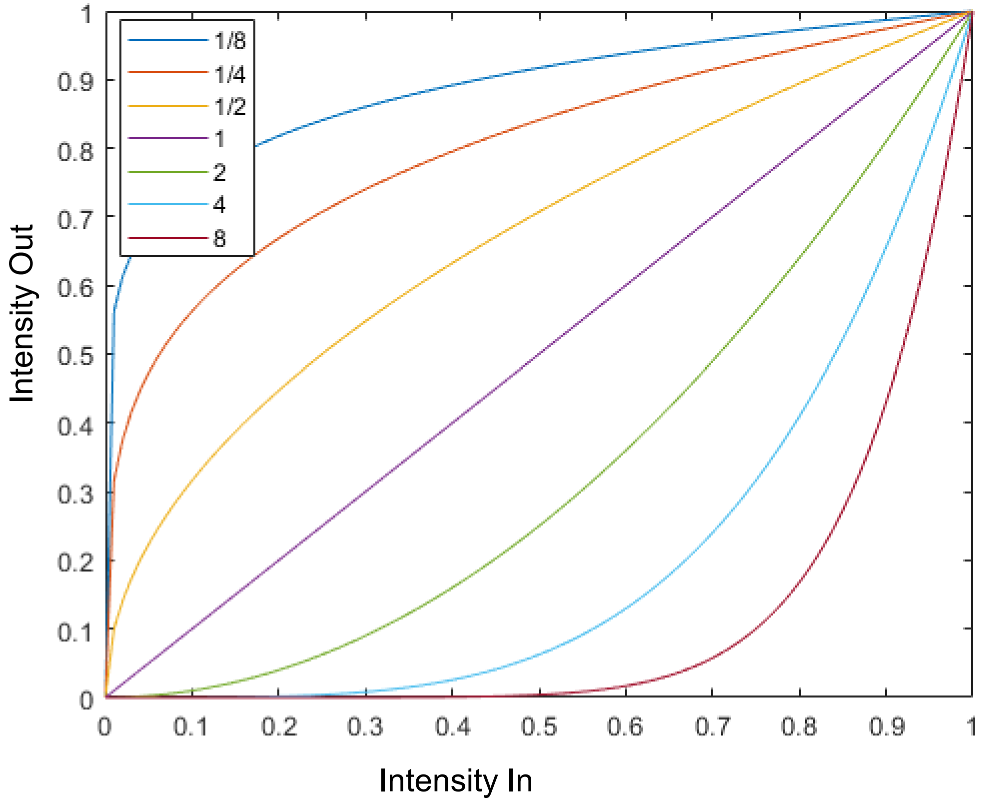




Cumulative Histograms

Gamma adjustment

$$i_{out} = {i_{in}}^{\gamma}$$

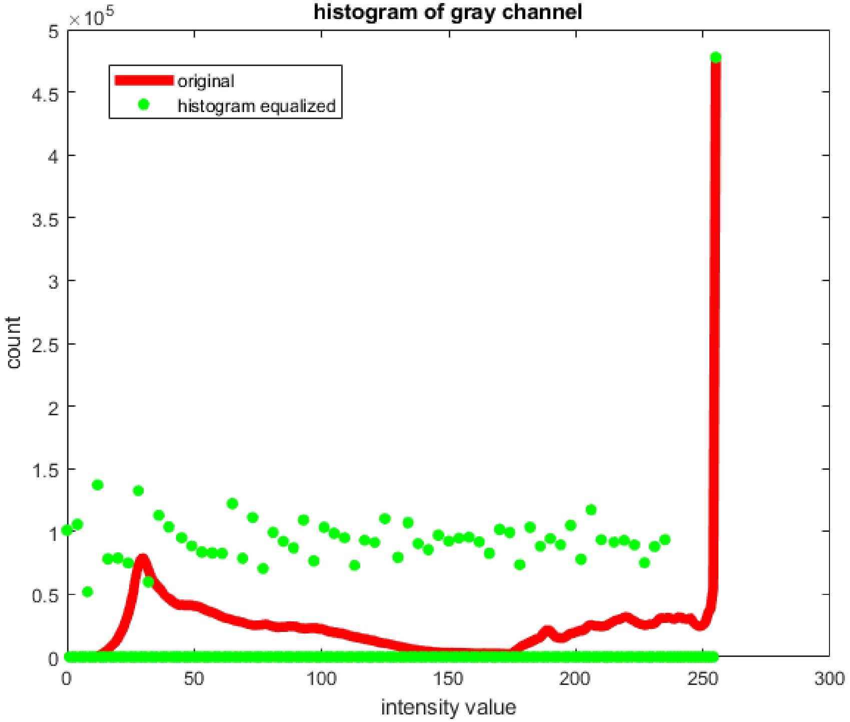


$\gamma = 0.5$

$\gamma = 1$

$\gamma = 2$

Histogram Equalization

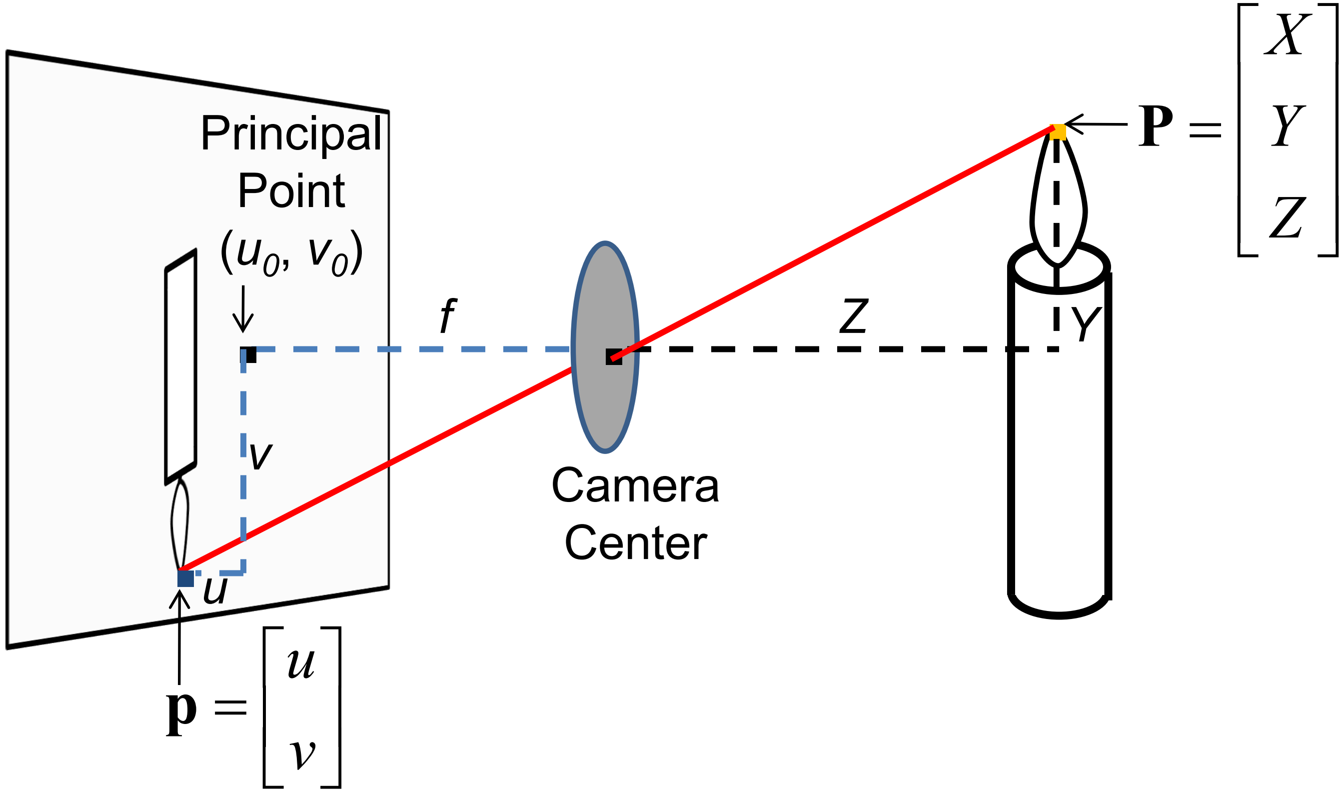


Practice question

19. For each material, indicate whether its reflection can be modeled as Lambertian, specular, or mixed:
    a. Balloon
    b. Cotton shirt
    c. Polished wood
    d. Mirror
    e. Cement block

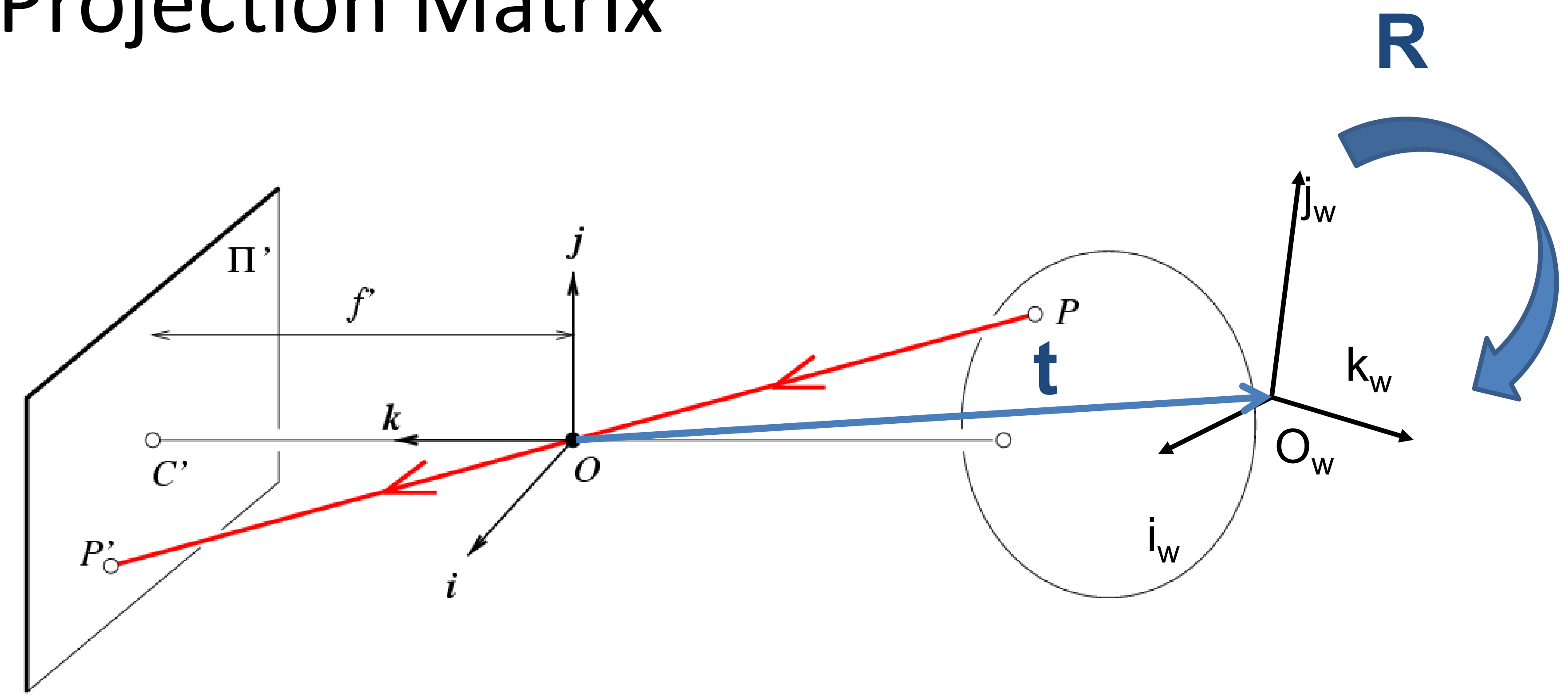
2. Camera Capture and Geometry

Pinhole Camera



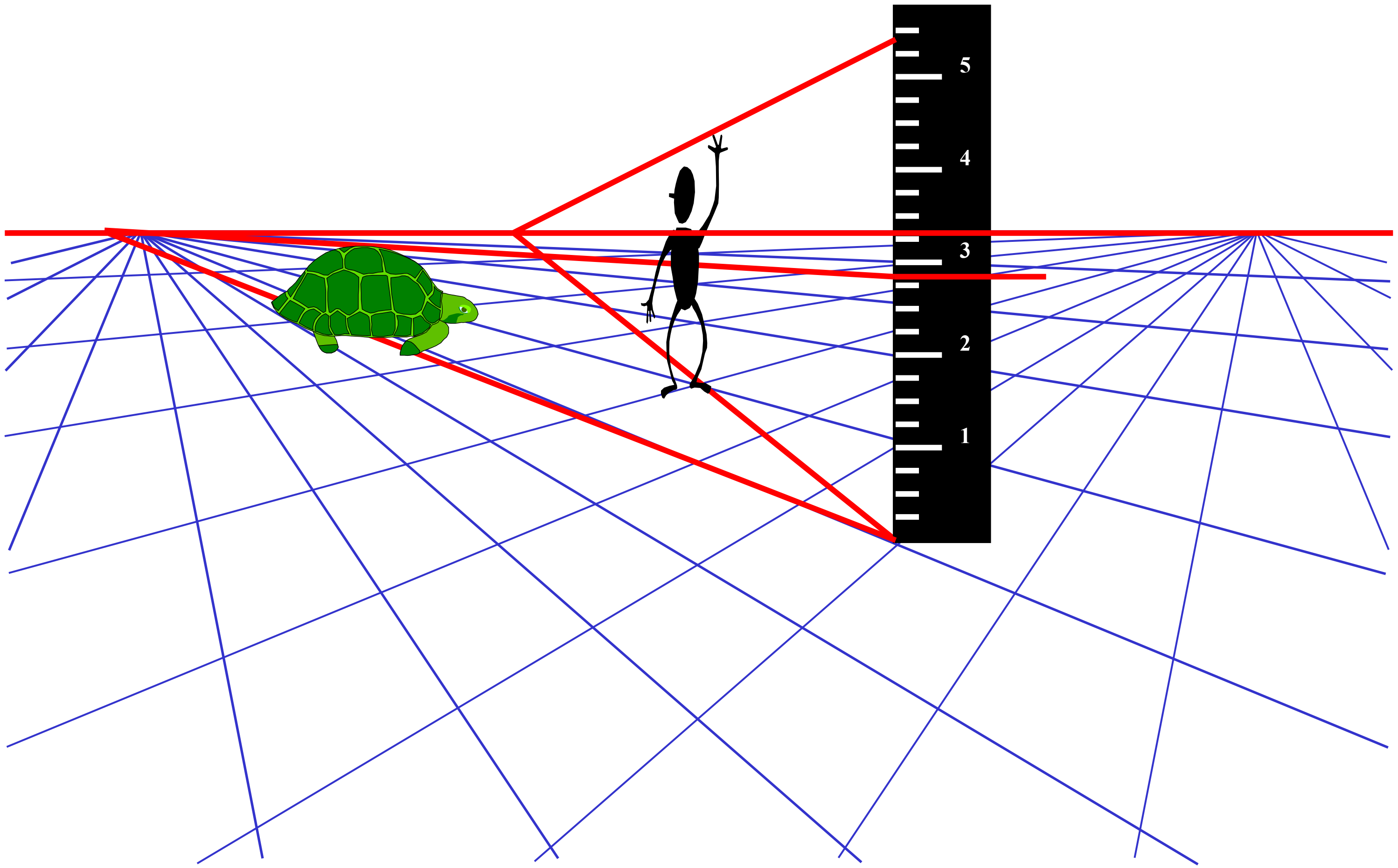
Useful figure to remember

Projection Matrix



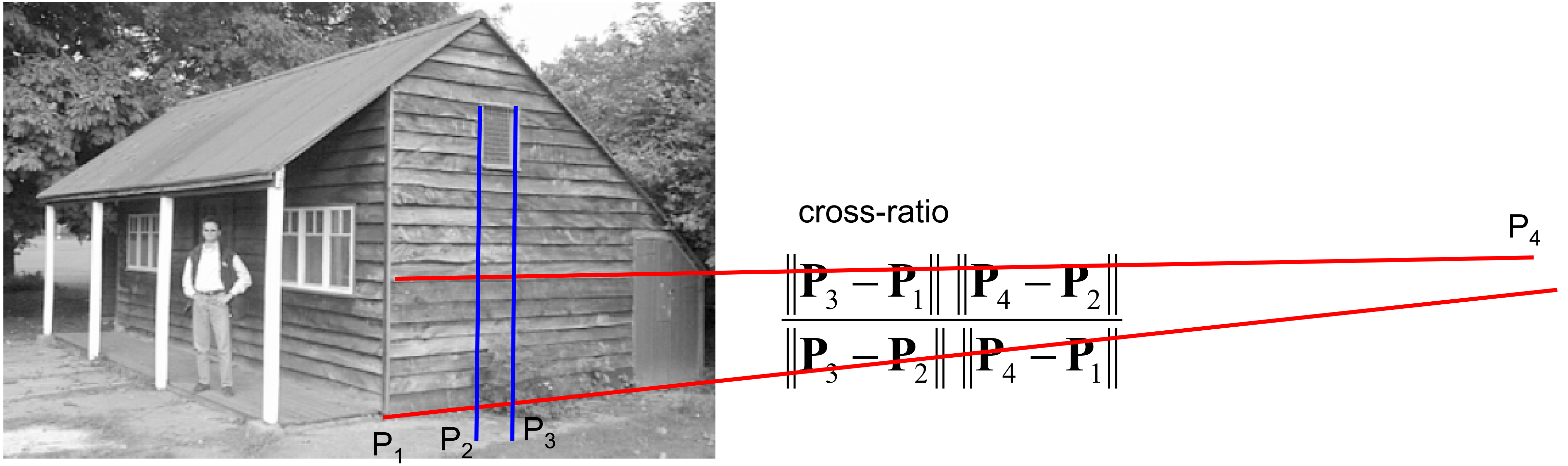
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f & 0 & u_0\\0 & f & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Single-view metrology



Assume the man is 6 ft tall.

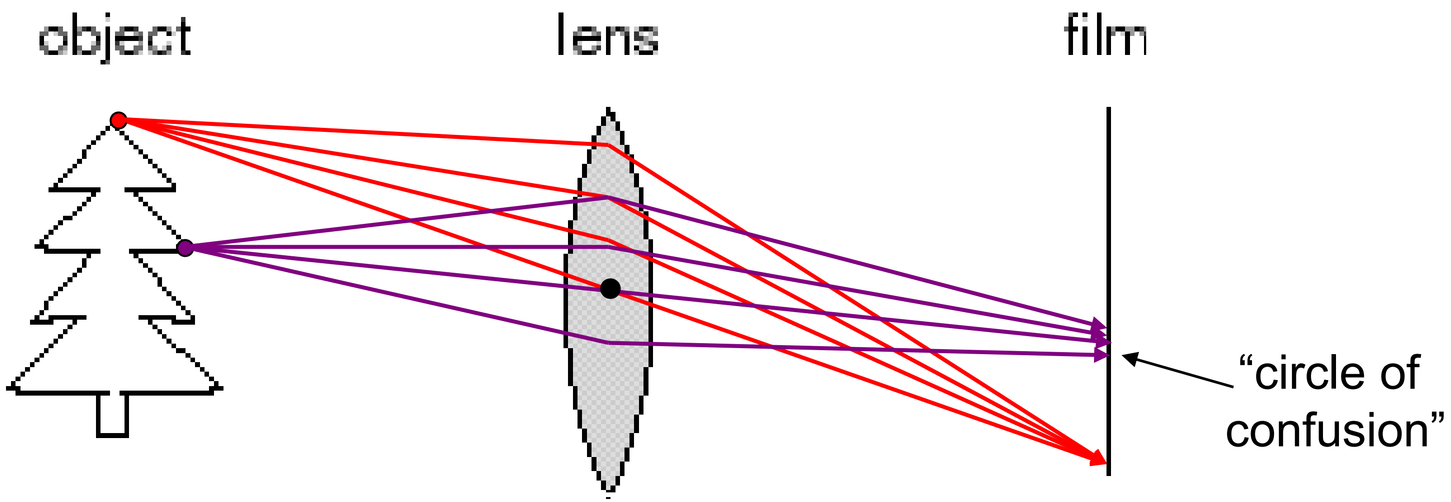
- What is the height of the building?
- How long is the right side of the building compared to the small window on the right side of the building?



cross-ratio

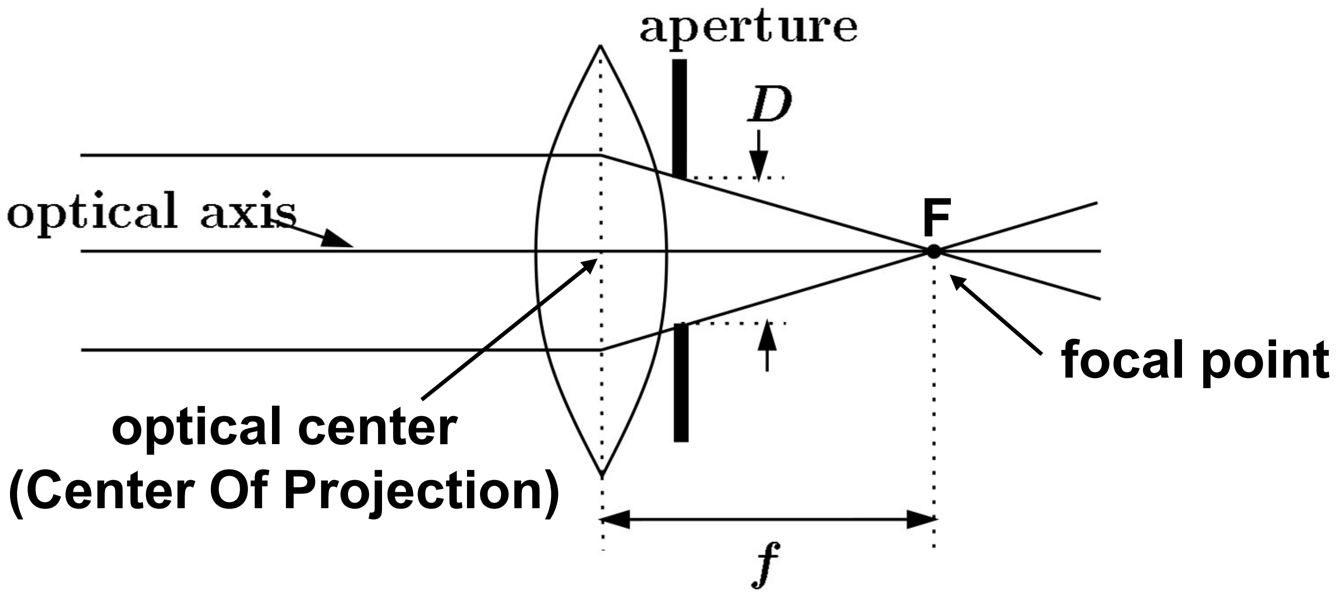
$$\frac{\|\mathbf{P}_3-\mathbf{P}_1\|\,\|\mathbf{P}_4-\mathbf{P}_2\|}{\|\mathbf{P}_3-\mathbf{P}_2\|\,\|\mathbf{P}_4-\mathbf{P}_1\|}$$

Adding a lens



- A lens focuses light onto the film
  - There is a specific distance at which objects are “in focus”
    - other points project to a “circle of confusion” in the image
  - Changing the shape of the lens changes this distance

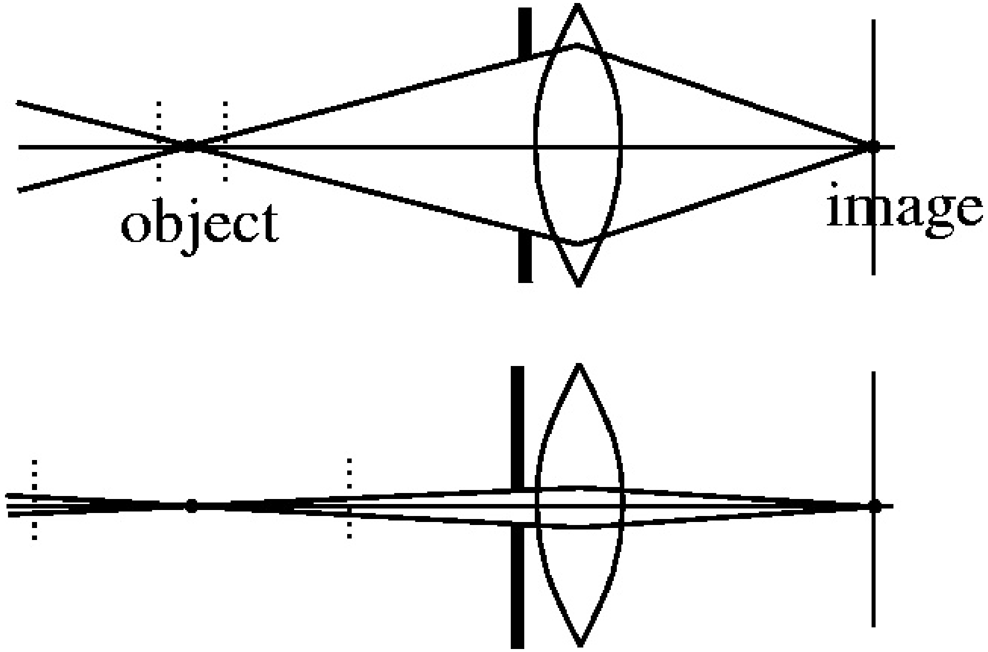
Focal length, aperture, depth of field



A lens focuses parallel rays onto a single focal point

- focal point at a distance *f* beyond the plane of the lens
- Aperture of diameter D restricts the range of rays

The aperture and depth of field



f / 5.6

f / 32

Main way to increase depth of field: Decrease aperture size

F-number (f/#) =focal_length / aperture_diameter

- E.g., f/16 means that the focal length is 16 times the diameter
- When you set the f-number of a camera, you are setting the aperture

Flower images from Wikipedia http://en.wikipedia.org/wiki/Depth_of_field

The Photographer's Great Compromise

What we want	How we get it	Cost
More spatial resolution	Increase focal length	Light, FOV
Broader field of view	Decrease focal length	DOF
More depth of field	Decrease aperture	Light
	Increase aperture	DOF
More temporal resolution	Shorten exposure	Light
More light	Lengthen exposure	Temporal Res

Practice questions

- 39, 44, 45

3. Linear filtering

- Can think of filtering as
  - A function in the spatial domain (e.g., compute average of each 3x3 window)
  - Template matching
  - Modifying the frequency of the image

Filtering in spatial domain

Slide filter over image and take dot product at each position

$g[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
				?					

1.Filters

a) Compute the center four values of the filter response, using the filter and image given below.

Filter

-1	0	-1
0	4	0
-1	0	-1

*

Input Image

0	1	0	0
0	2	6	0
5	3	2	1
1	2	1	0

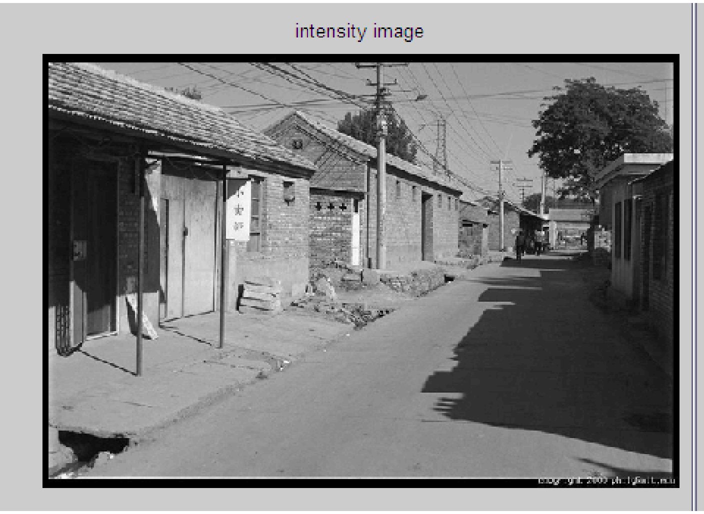
→

Filter Response

x	x	x	x
x			x
x			x
x	x	x	x

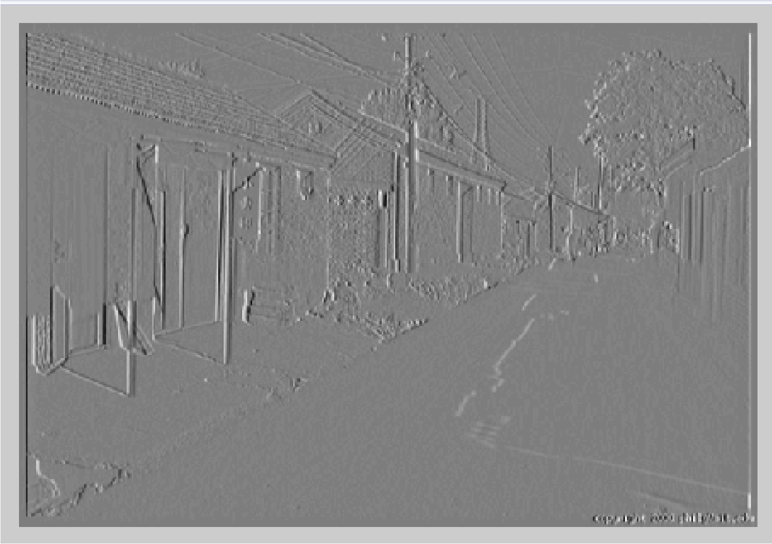
Filtering in spatial domain

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



*

=



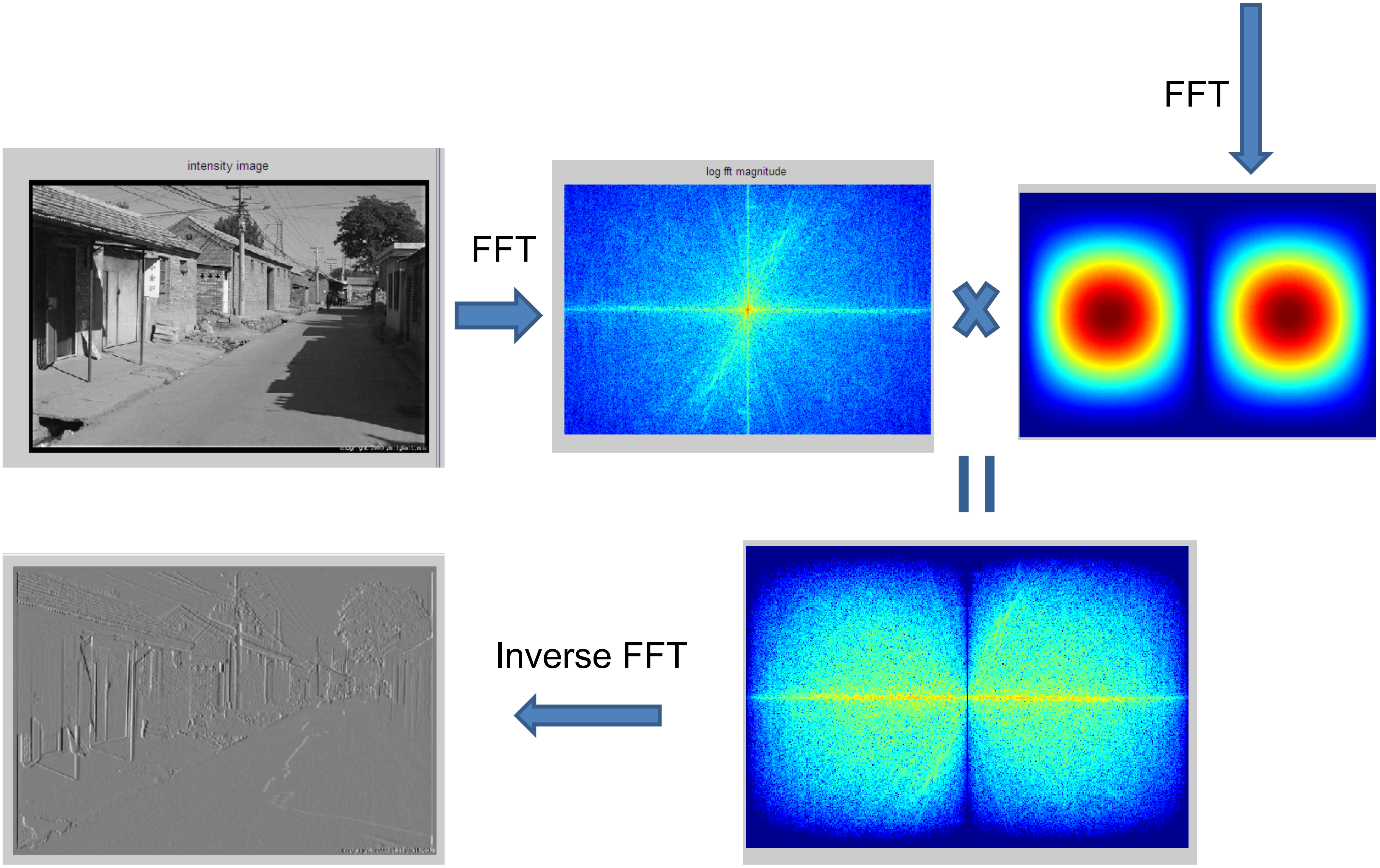
Filtering in frequency domain

- Can be faster than filtering in spatial domain (for large filters)

- Can help understand effect of filter

- Algorithm:
  1. Convert image and filter to fft (fft2 in matlab)
  2. Pointwise-multiply ffts
  3. Convert result to spatial domain with ifft2

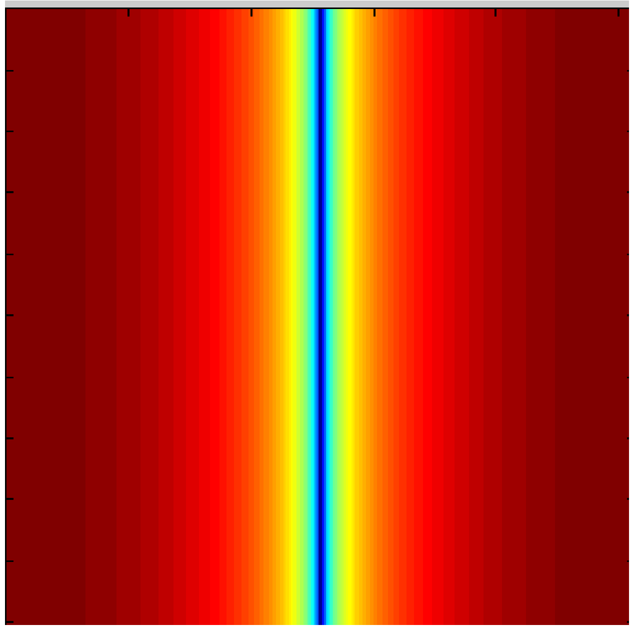
Filtering in frequency domain



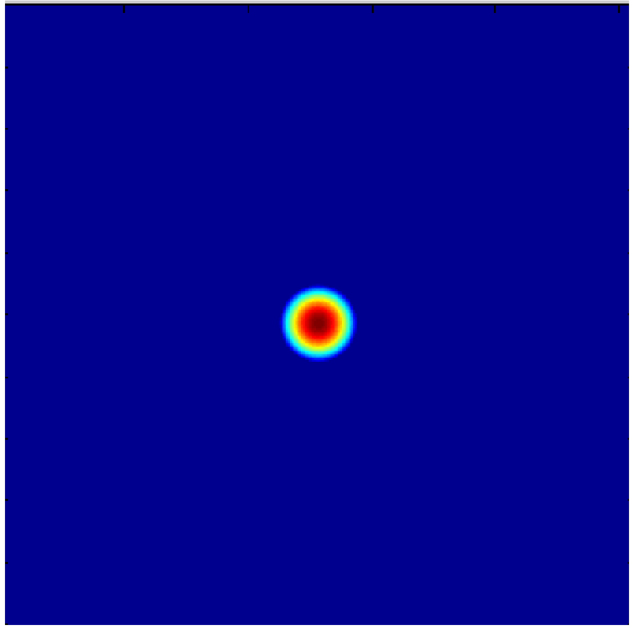
Filtering in frequency domain

- Linear filters for basic processing
  - Edge filter (high-pass)
  - Gaussian filter (low-pass)

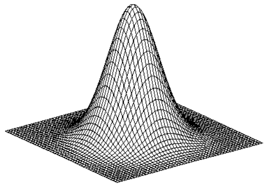
[-1 1]



FFT of Gradient Filter



FFT of Gaussian

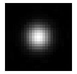


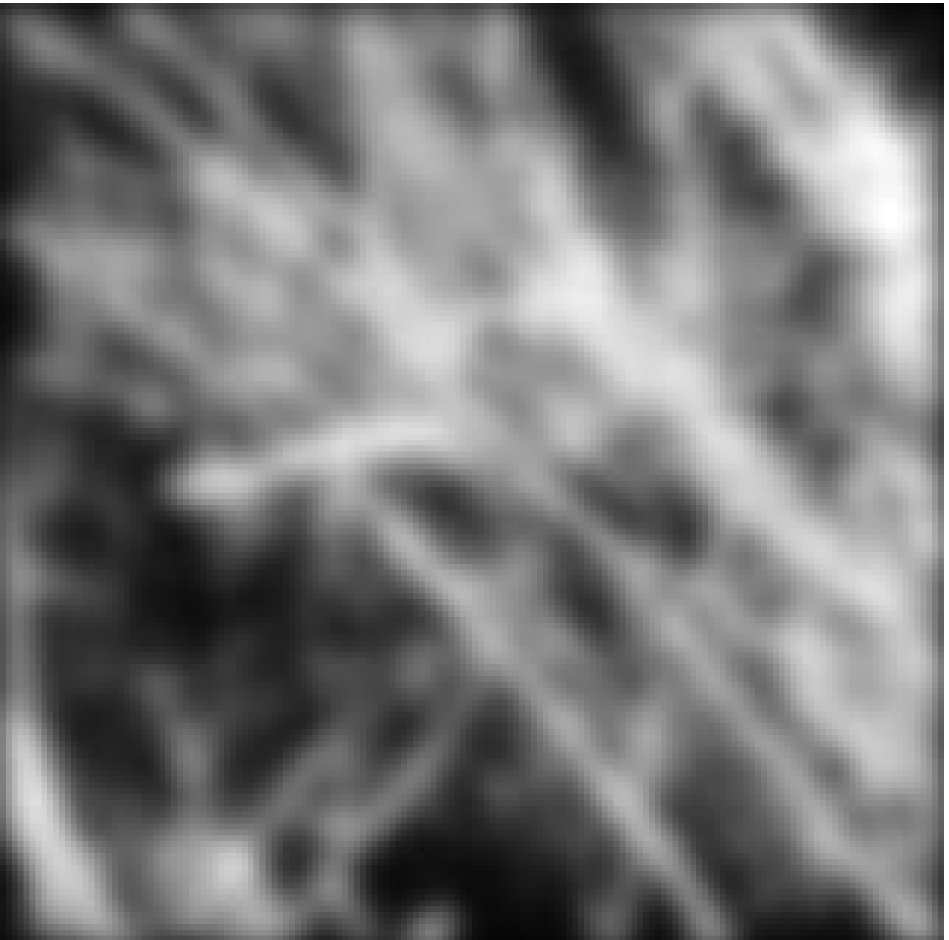
Gaussian

Filtering

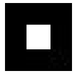
Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

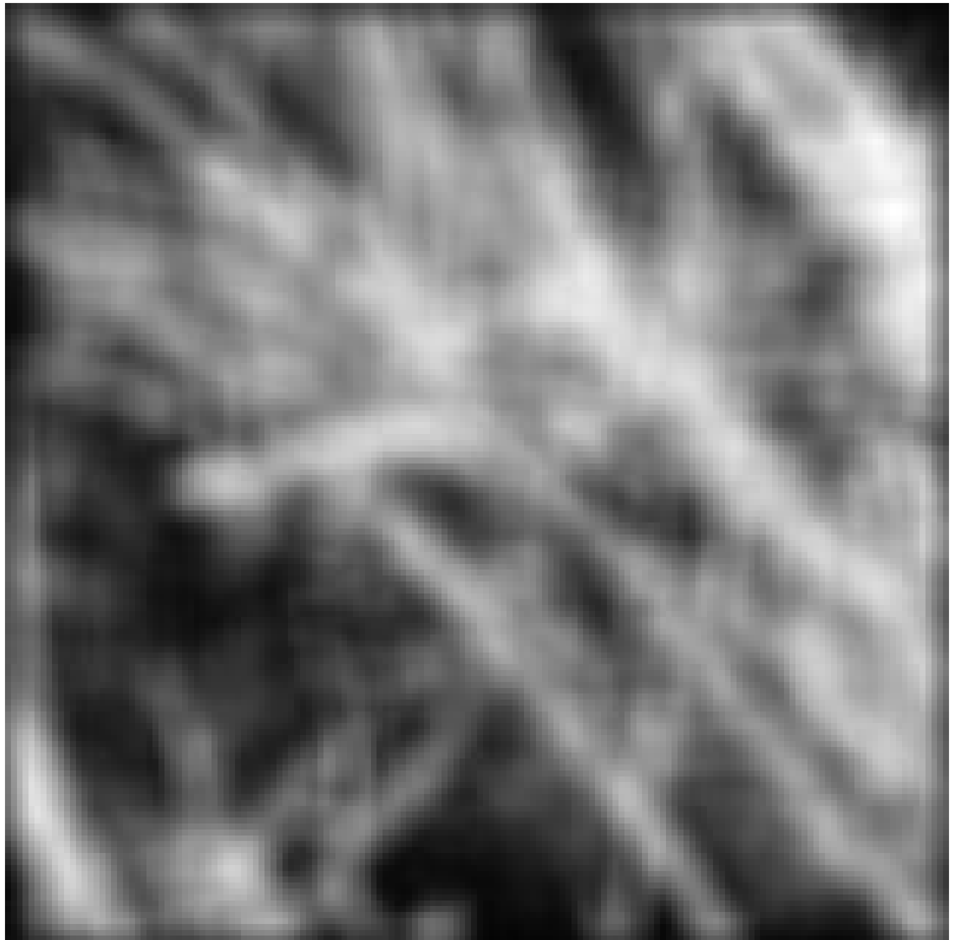
Gaussian



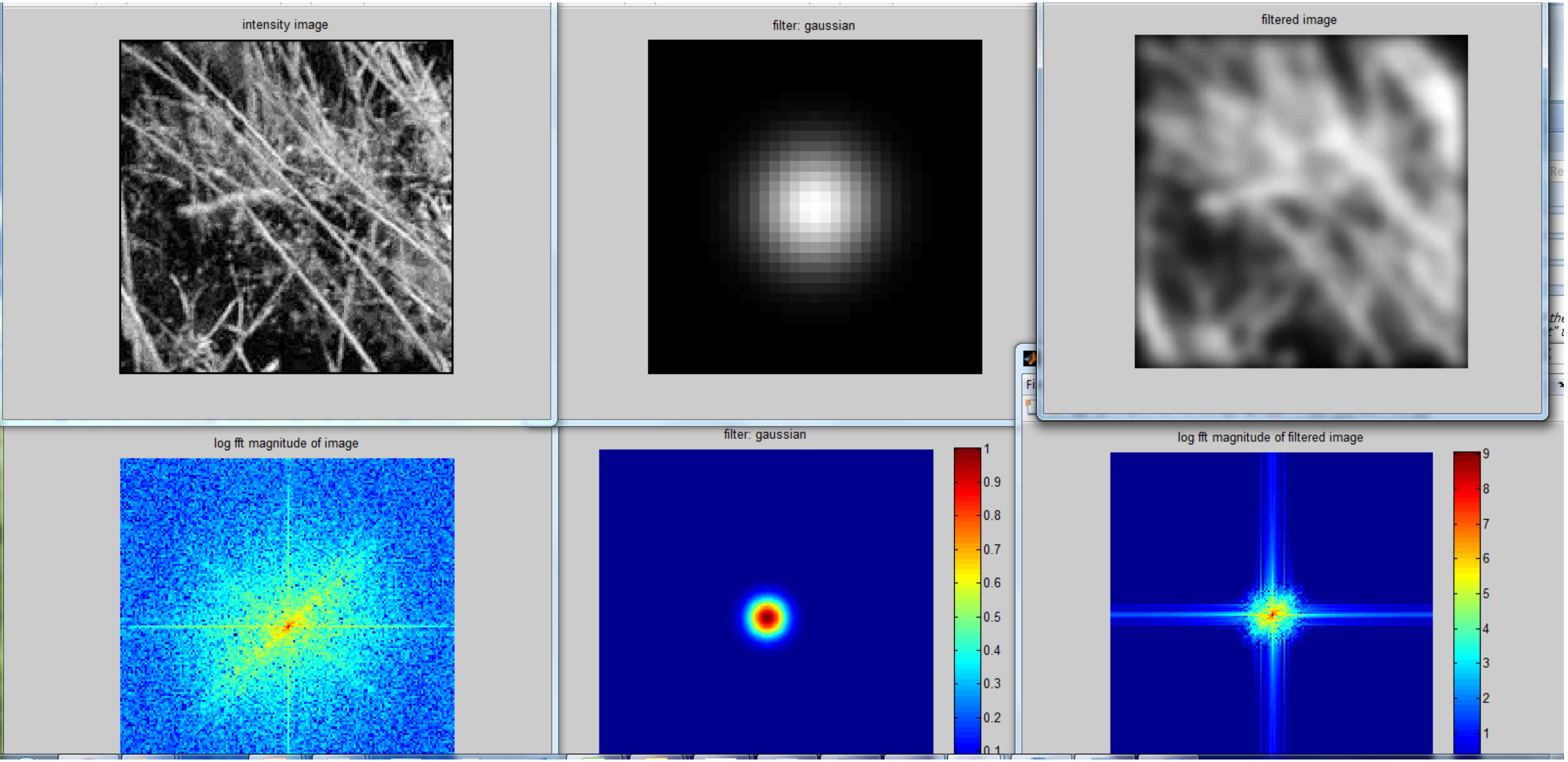


Box filter



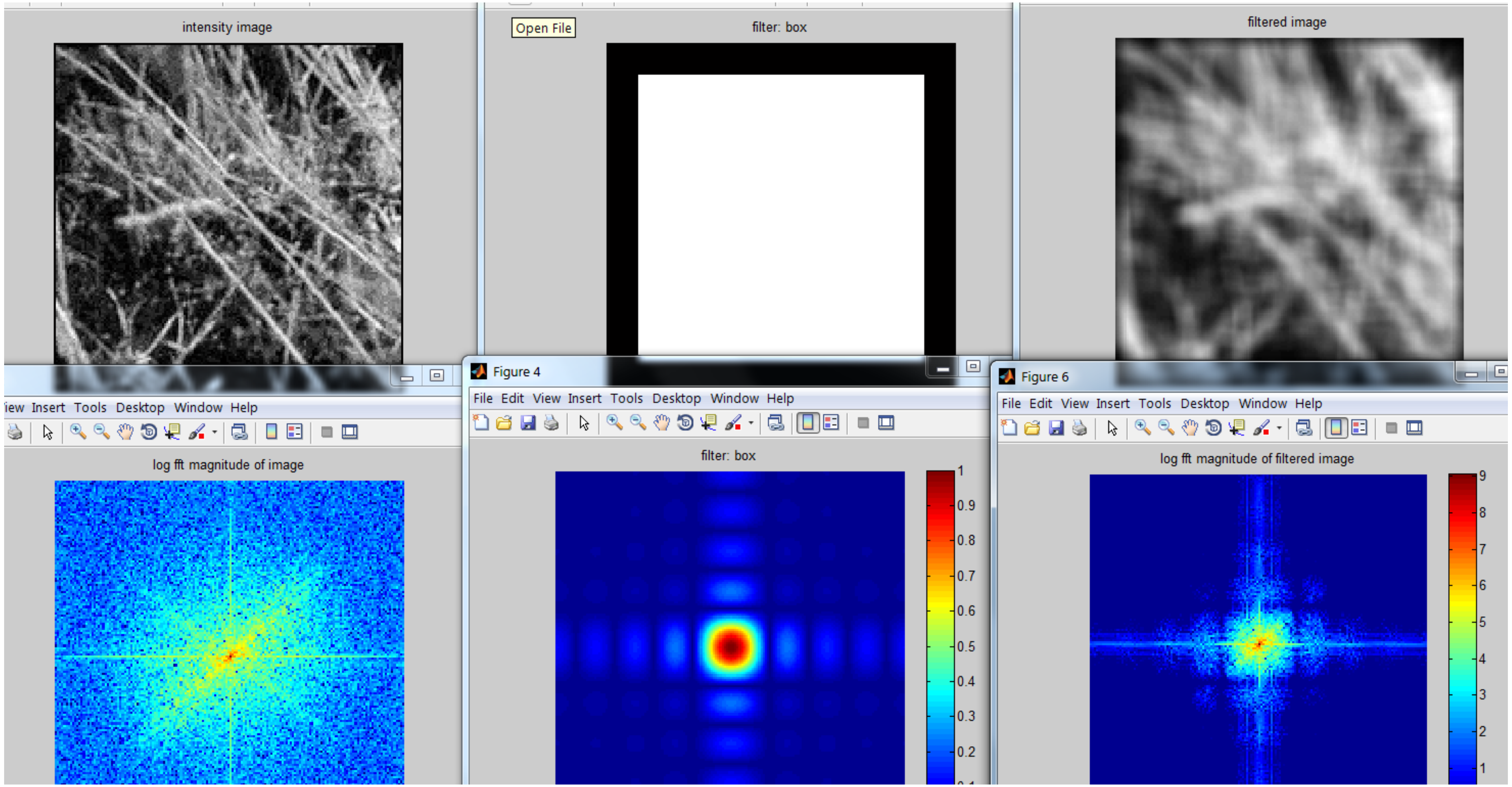


Gaussian



intensity image
filter: gaussian
filtered image
log fft magnitude of image
filter: gaussian
log fft magnitude of filtered image

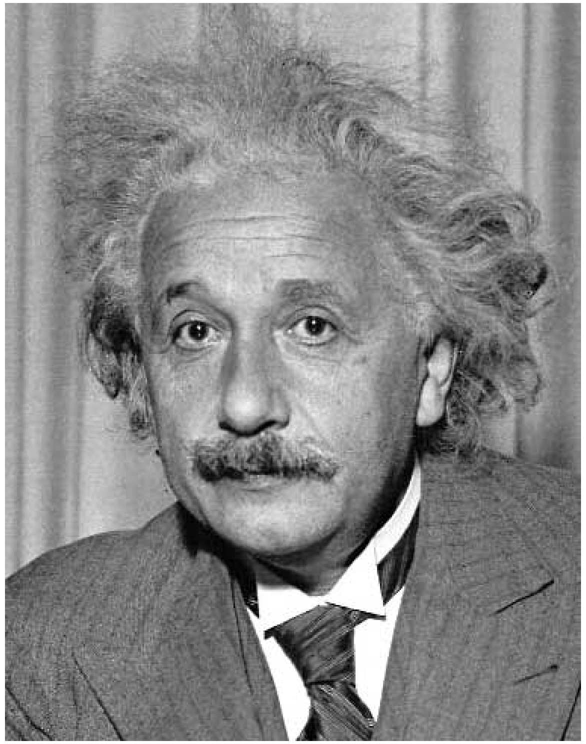
Box Filter


intensity image
Open File
filter: box
filtered image
Figure 4
Figure 6
iew Insert Tools Desktop Window Help
File Edit View Insert Tools Desktop Window Help
File Edit View Insert Tools Desktop Window Help
log fft magnitude of image
filter: box
log fft magnitude of filtered image
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
9
8
7
6
5
4
3
2
1

Matching with filters

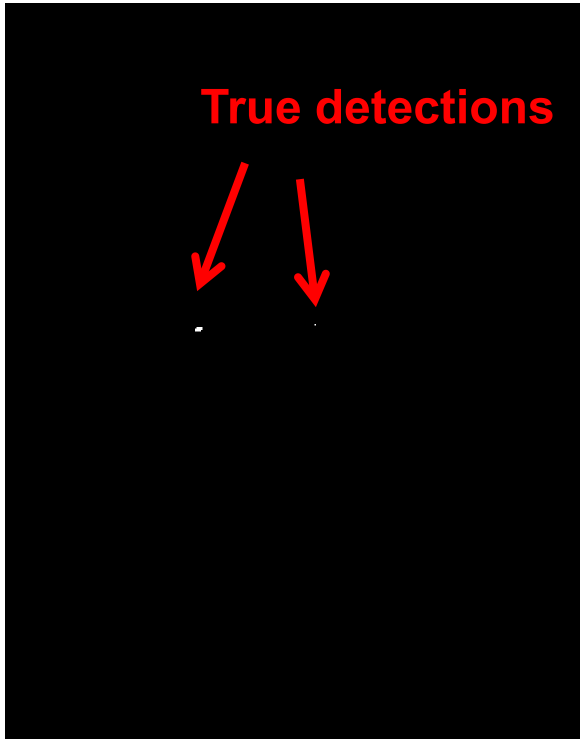
- Goal: find  in image
- Method 2: SSD

$$h[m,n]=\sum_{k,l}(g[k,l]-f[m+k,n+l])^2$$



Input

1- sqrt(SSD)



Thresholded Image

Matching with filters

- Goal: find  in image
- Method 3: Normalized cross-correlation

mean template — $\bar{g}$

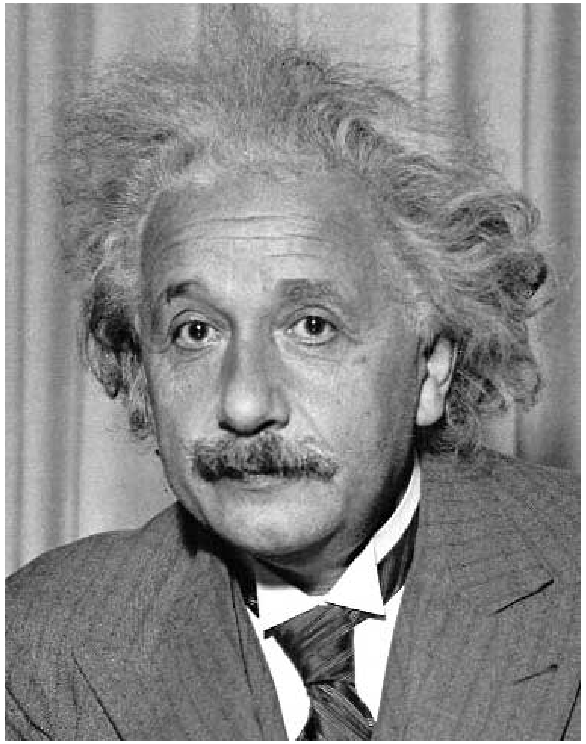
mean image patch — $\bar{f}_{m,n}$

$$h[m,n] = \frac{\sum_{k,l} (g[k,l] - \bar{g})(f[m-k,n-l] - \bar{f}_{m,n})}{\left(\sum_{k,l} (g[k,l] - \bar{g})^2 \sum_{k,l} (f[m-k,n-l] - \bar{f}_{m,n})^2 \right)^{0.5}}$$

Matlab: `normxcorr2(template, im)`

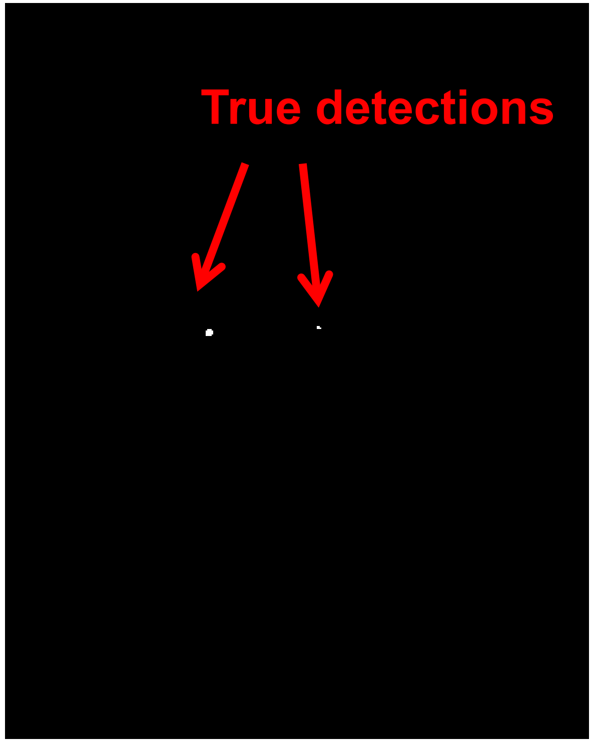
Matching with filters

- Goal: find  in image
- Method 3: Normalized cross-correlation



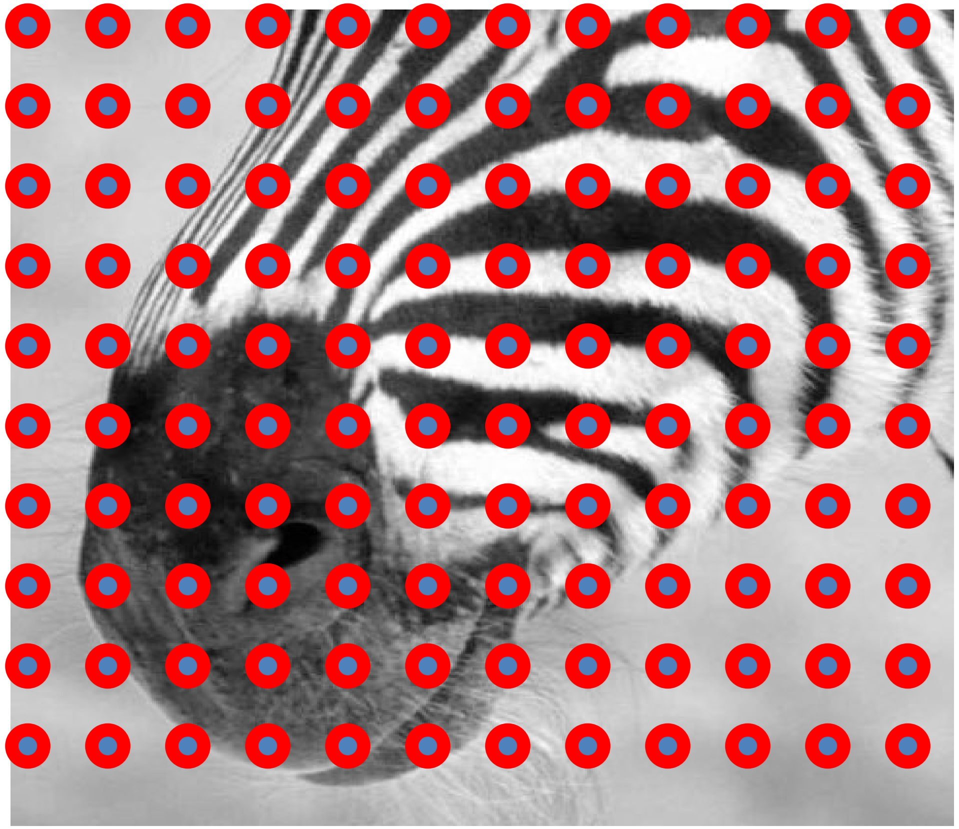
Input

Normalized X-Correlation



Thresholded Image

Subsampling by a factor of 2

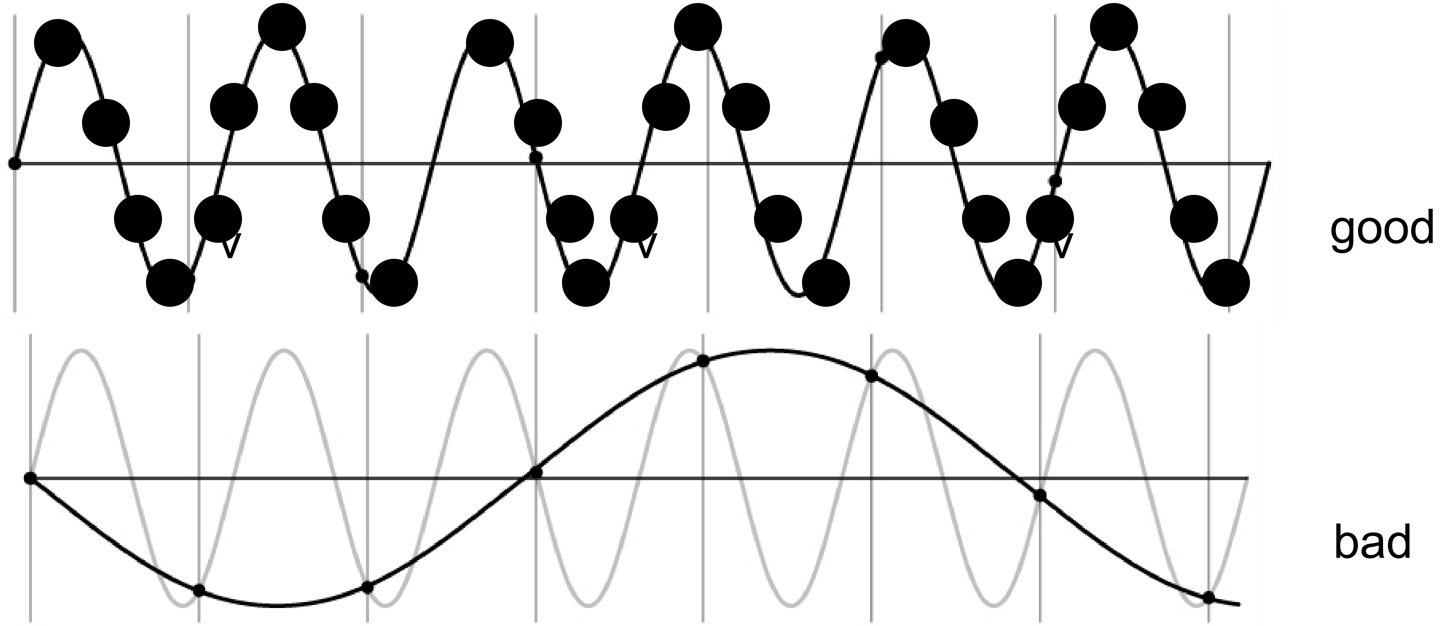


Throw away every other row and column to create a 1/2 size image

Problem: This approach causes "aliasing"

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



Algorithm for downsampling by factor of 2

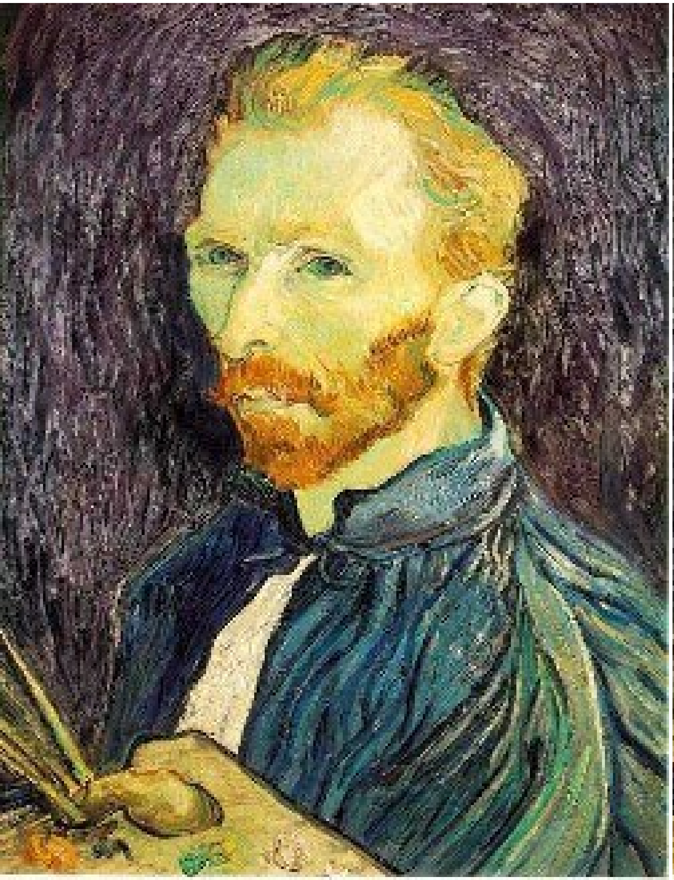
1. Start with image(h, w)
2. Apply low-pass filter

   im_blur = imfilter(image, fspecial('gaussian', 13, 2))

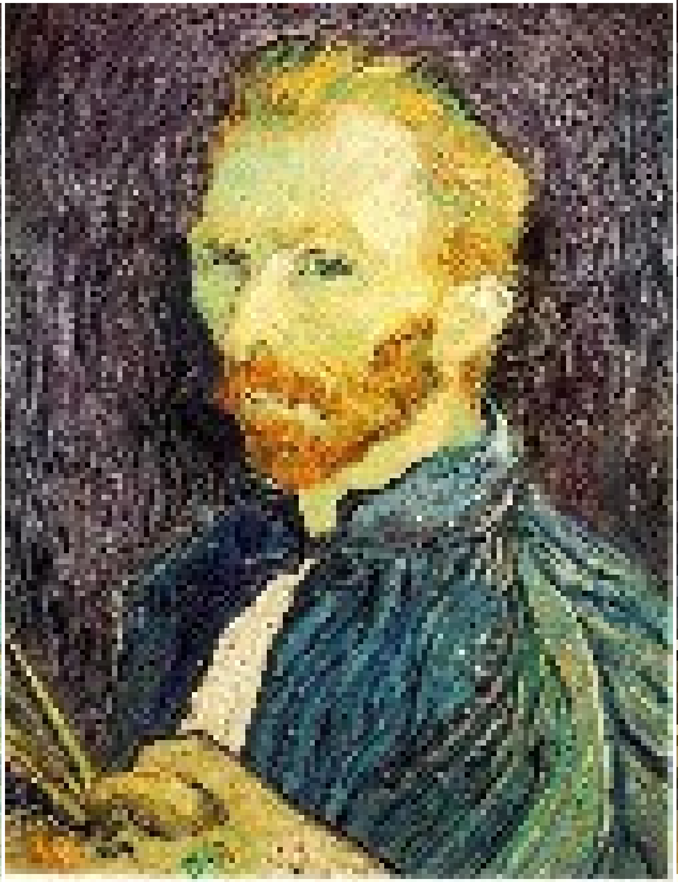
3. Sample every other pixel

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

Subsampling without pre-filtering



1/2

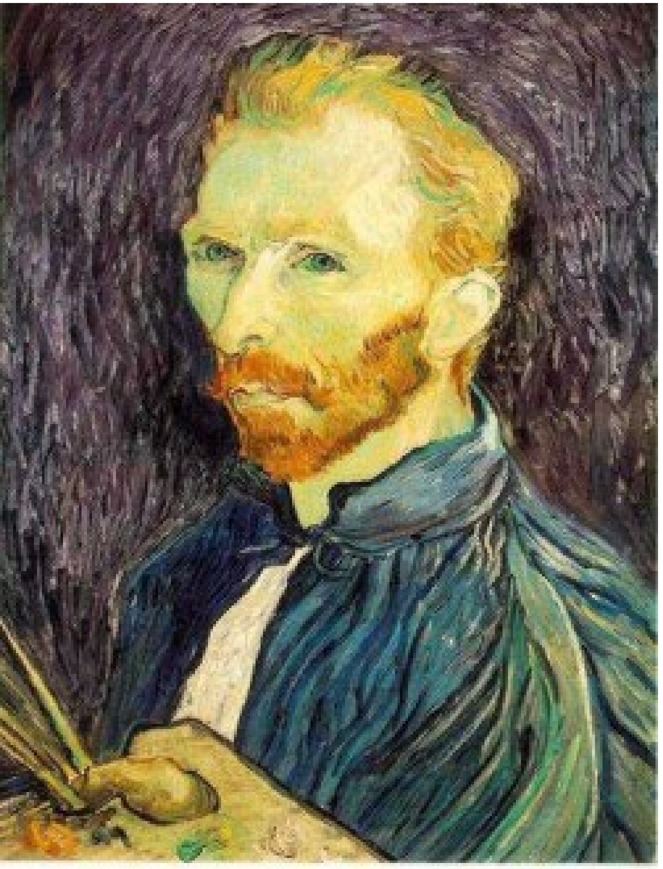


1/4 (2x zoom)

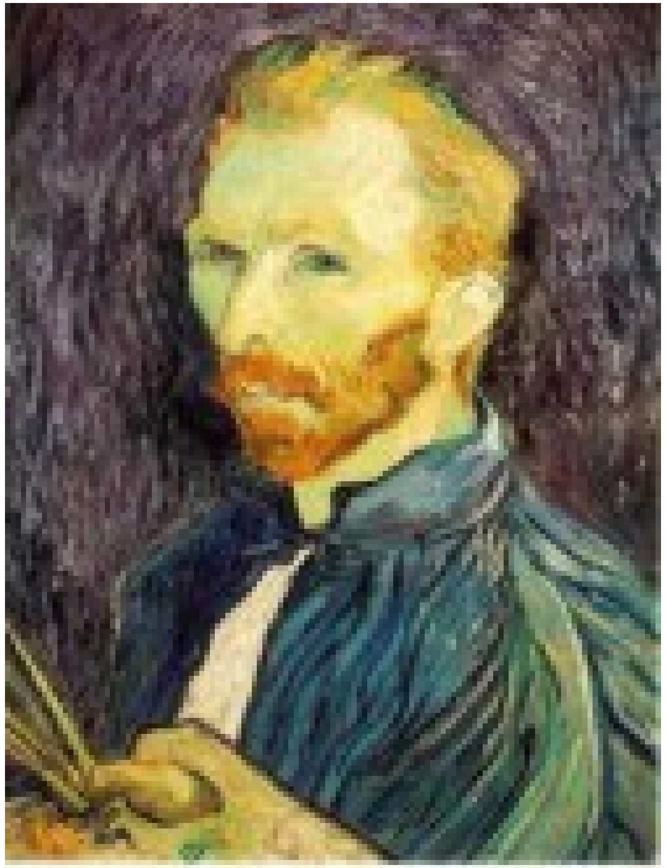
1/8 (4x zoom)

Slide by Steve Seitz

Subsampling with Gaussian pre-filtering



Gaussian 1/2



G 1/4

G 1/8

Slide by Steve Seitz

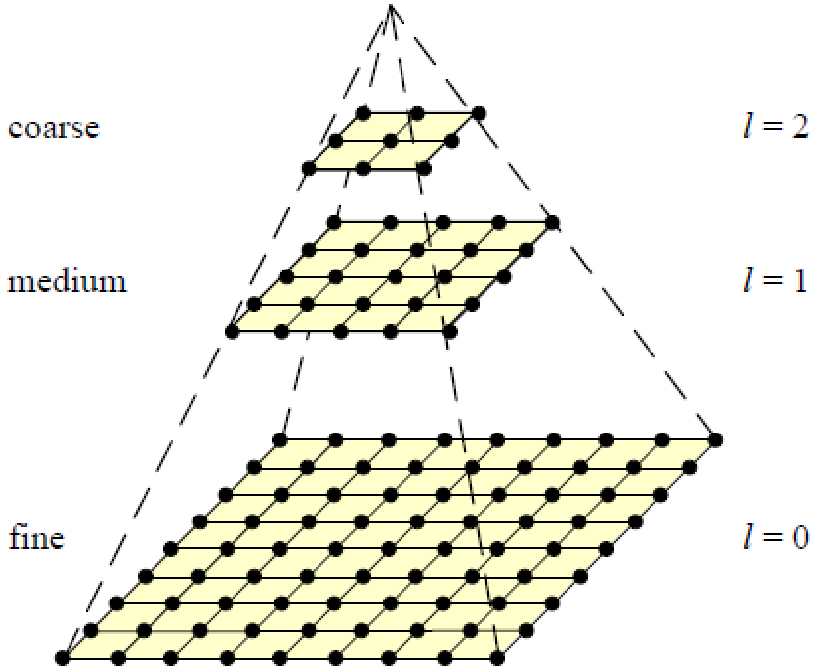
Sampling

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

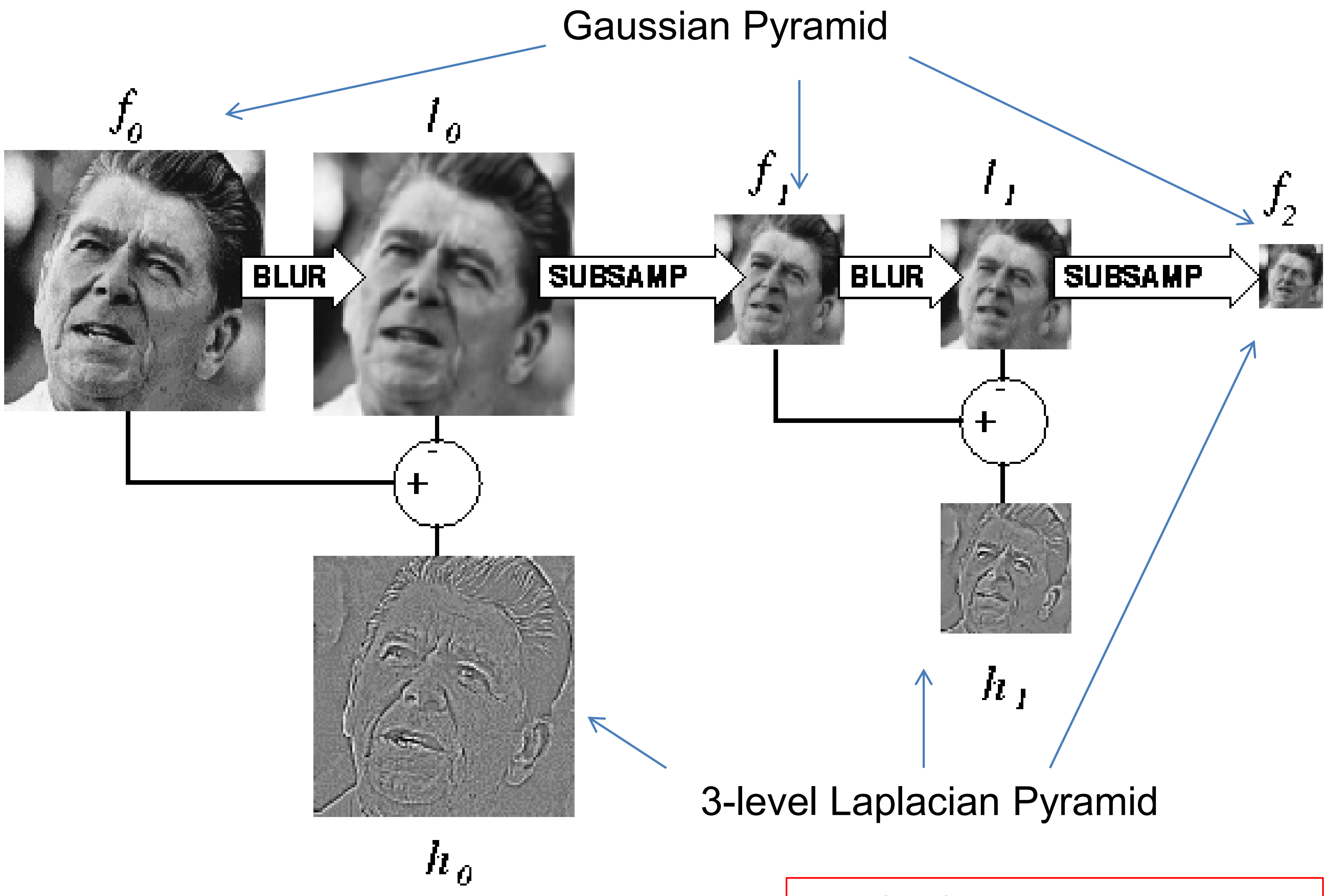
Image: http://www.flickr.com/photos/igorms/136916757/

Gaussian pyramid

- Useful for coarse-to-fine matching
- Applications include multi-scale object detection, image alignment, optical flow, point tracking



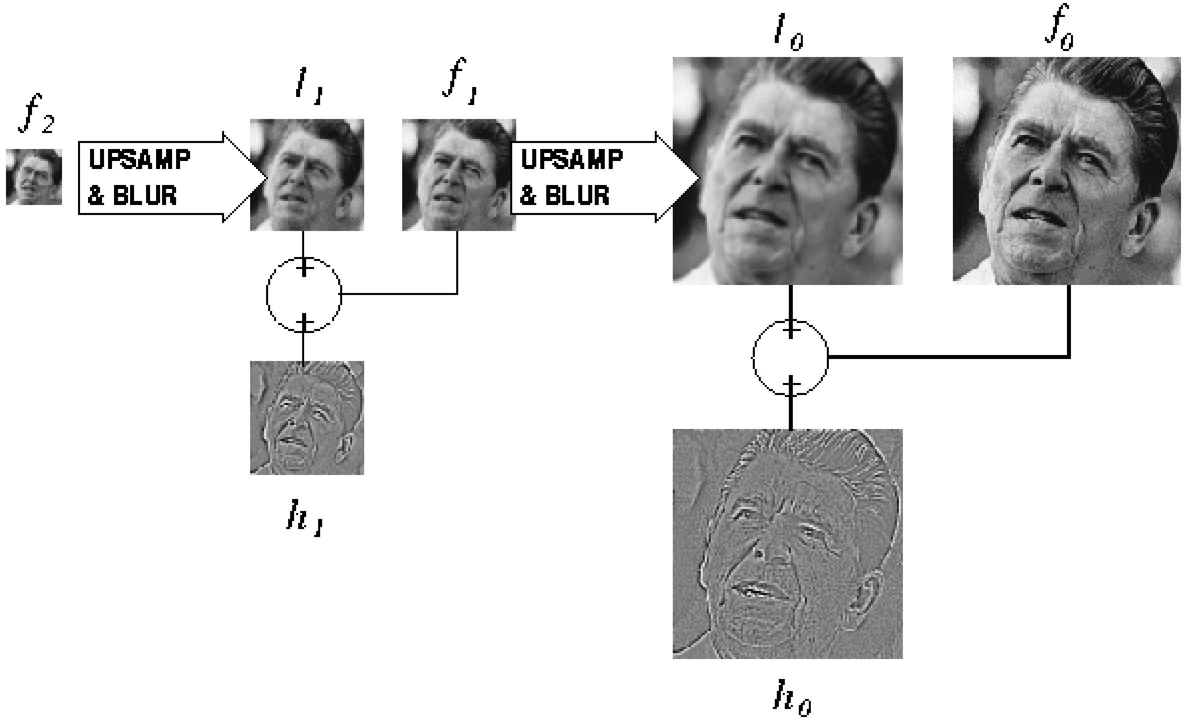
Computing Gaussian/Laplacian Pyramid



Useful figure to remember

http://sepwww.stanford.edu/~morgan/texturematch/paper_html/node3.html

Image reconstruction from Laplacian pyramid



Practice questions

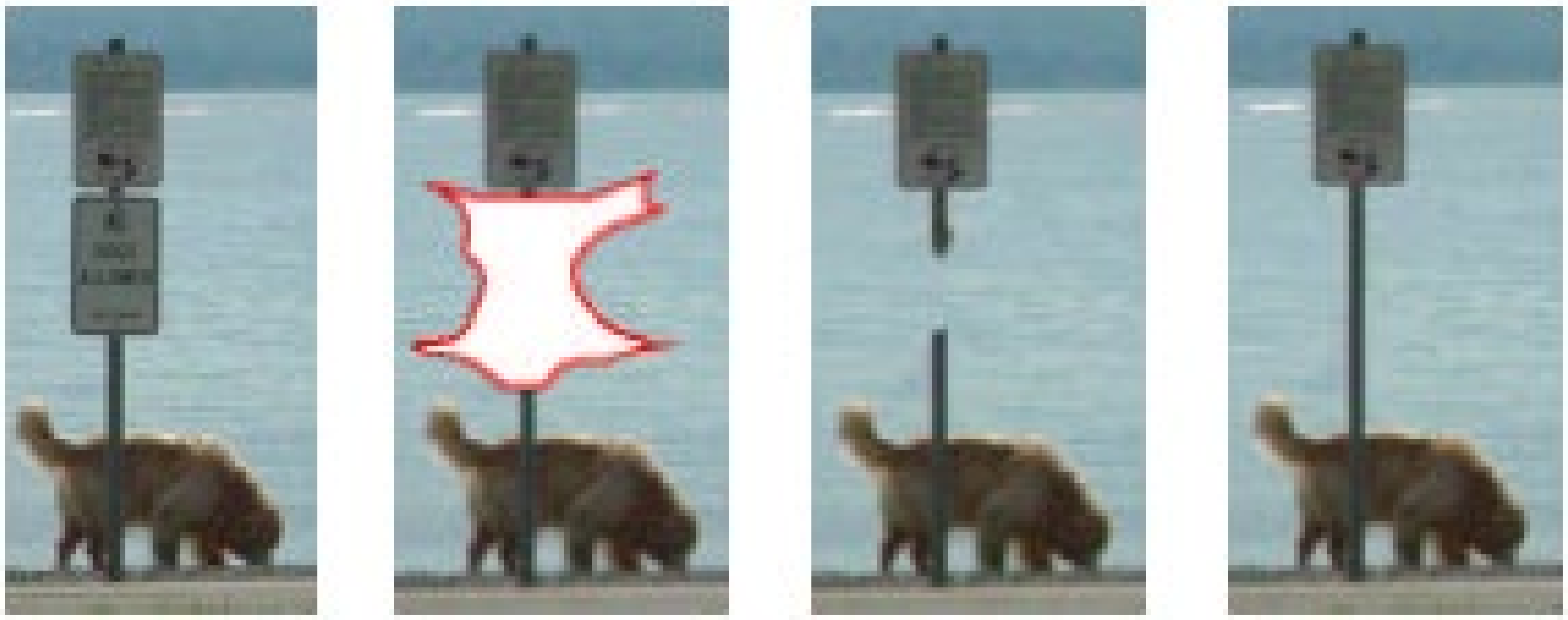
- 1,2,5,7,9,12

4. Cutting and pasting

- Texture synthesis
- Selecting image regions
  - Intelligent scissors
  - Graph cuts
- Compositing
  - Alpha masks
  - Feathering
  - Laplacian pyramid blending
  - Poisson blending

Texture synthesis and hole-filling

- Efros & Leung: To fill in pixels, take the center pixel from a patch that matches known surrounding values
- Criminisi et al: Work from the outside in, prioritizing pixels with strong gradients

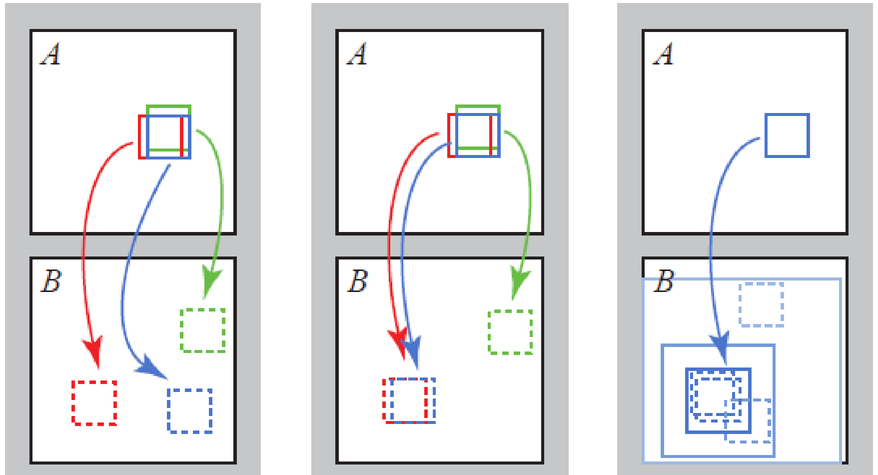


PatchMatch: Optimization

- Goal: Solve for labels that minimize some cost function
- Key assumption: a pixel and its neighbor very likely have the same or similar labels

PatchMatch Algorithm basics

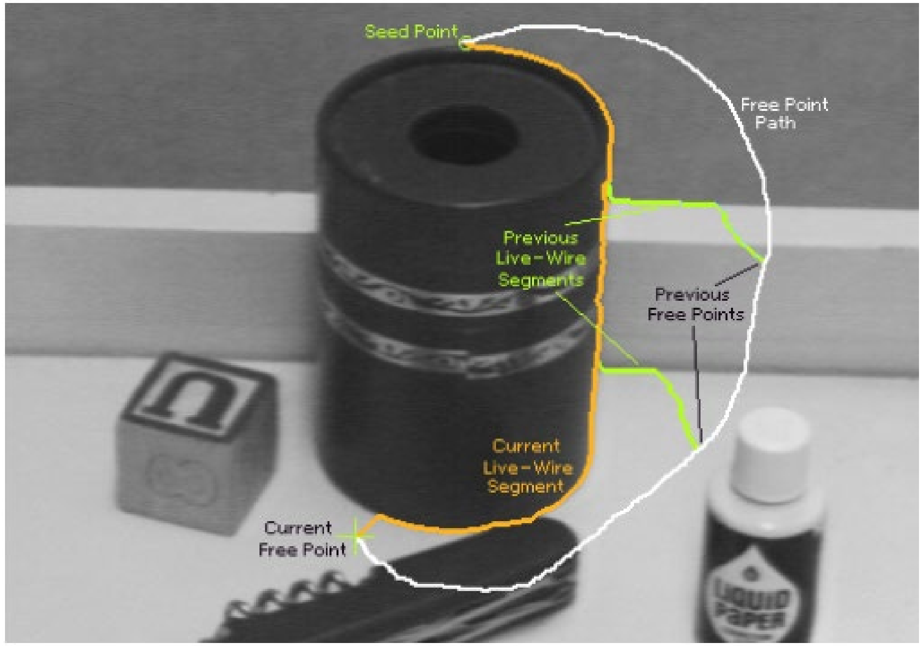
1. Randomly initialize matches
2. Scan across image (forward and backward)
    a. Check if neighbor's offsets or random perturbations around current offset produce better scores
    b. Keep best found so far
3. Repeat (2) several times



(a) Initialization (b) Propagation (c) Search

Intelligent Scissors

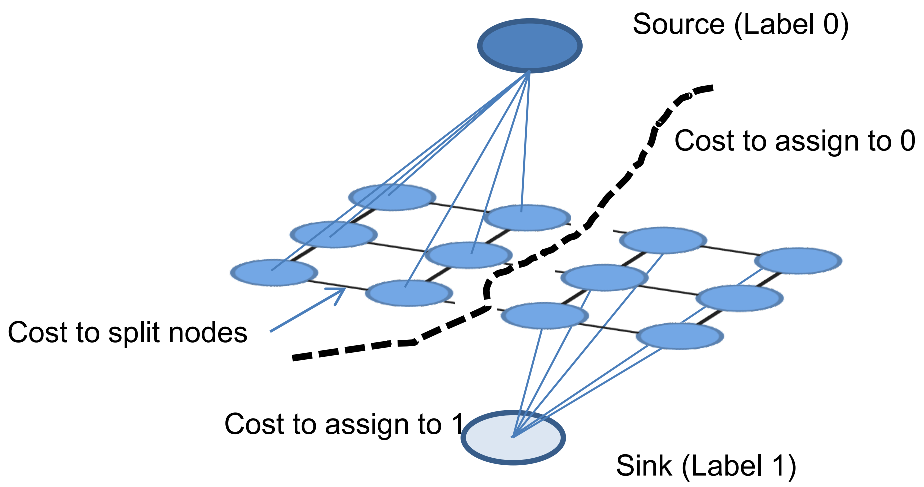
- You can treat the image as a graph
  - Nodes = pixels, edges connect neighboring pixels



Intelligent Scissors: Good boundaries are a short (high gradient) path through the graph

Graph Cuts

- You can treat the image as a graph
  - Nodes = pixels, edges connect neighboring pixels



Graph cut: Good boundaries are a cheap cut, where some pixels want to be foreground, and some to be background

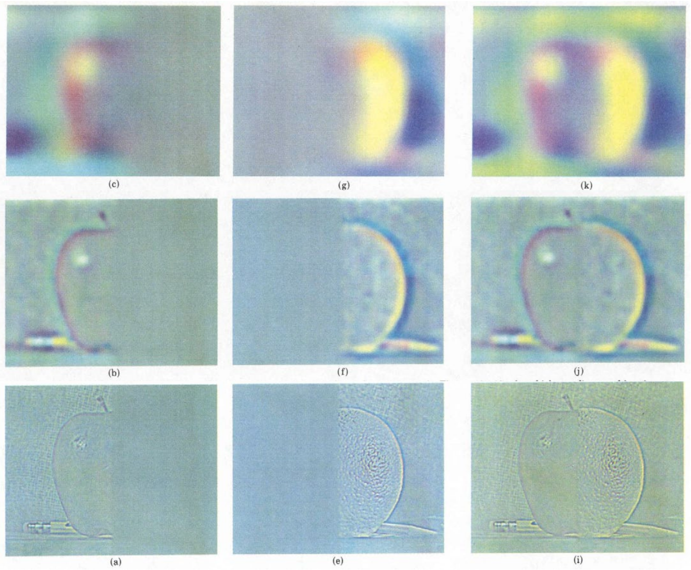
Compositing and Blending

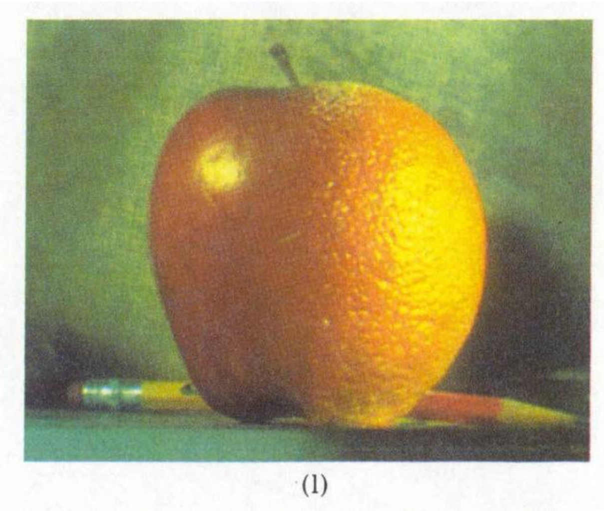
- Feathering: blur mask around its edges
- Laplacian blending: blend low-frequency slowly, high frequency quickly
  - Blend with alpha mask values ranging from 0 to 1

laplacian level 4

laplacian level 2

laplacian level 0





Question

1) I am trying to blend this bear into this pool. What problems will I have if I use:
   a) Alpha compositing with feathering
   b) Laplacian pyramid blending
   c) Poisson editing?

Transformations

- Practice questions

Basic 2D transformations as 3x3 matrices

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s_x} & 0 & 0 \\ 0 & \boldsymbol{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Scale

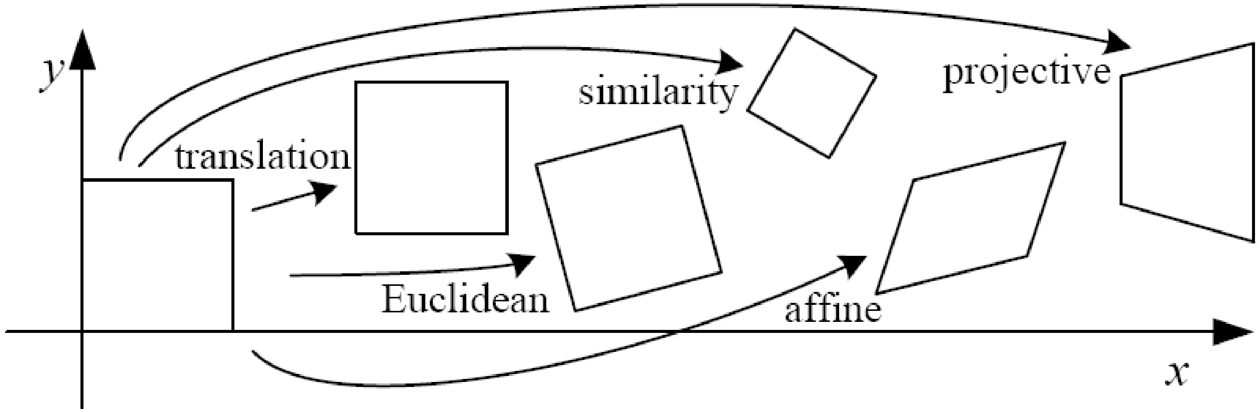
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \beta_x & 0 \\ \beta_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear

2D image transformations



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$			
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$			
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$			
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$			
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$			

These transformations are a nested set of groups

- Closed under composition and inverse is a member

Practice questions

- 32, 34, 36

Good luck!

- Exam is Thursday
  - Take on your own, in your own private space
  - Exam window is 9:30am to 10pm (can start exam by 10pm)
  - Let me know by email if you have trouble with access
  - Most students have 90 minutes
  - DRES students should automatically have additional time. Let me know if not

- Do not come to class on Thursday

- Questions?