Overview

• Brief summary of optics and aperture
• Camera body: shutter
• The sensor
  – Based on slides by G. Doretto

• Light and Shading
• Linear filters
  – Based on slides by D. Hoiem
Camera Body: Optics
Pinhole camera

- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the **aperture**
  - How does this transform the image?
Thin lens formula

\[ \frac{1}{s_o} + \frac{1}{s_i} = \frac{1}{f} \]
Changing the focal length vs. changing the viewpoint

- Telephoto makes it easier to select background (a small change in viewpoint is a big change in background).
  - changing the focal length lets us move back from a subject, while maintaining its size on the image
  - but moving back changes perspective relationships
Camera Body: Aperture
Aperture

- Diameter of the lens opening (controlled by diaphragm)
- Expressed as a fraction of focal length, in \textit{f-number} \( N \)
  - \( f/2.0 \) on a 50mm lens means that the aperture is 25mm
  - \( f/2.0 \) on a 100mm lens means that the aperture is 50mm
- Disconcerting: small f-number = big aperture
- What happens to the area of the aperture when going from \( f/2.0 \) to \( f/4.0 \)?
- Typical f-numbers are (each of them counts as one f/stop)
  - \( f/2.0, f/2.8, f/4, f/5.6, f/8, f/11, f/16, f/22, f/32 \)
    - See the pattern?

![Diagram of aperture settings: Full aperture, Medium aperture, Stopped down]
Depth of field

- We allow for some tolerance
Camera Body: Shutter
Shutter speed

- Controls how long the film/sensor is exposed
- Pretty much linear effect on exposure
- Usually in fraction of a second:
  - 1/30, 1/60, 1/125, 1/250, 1/500
  - Get the pattern?
- On a normal lens, normal humans can hand-hold down to 1/60
Main effect of shutter speed

- Motion blur
- Halving shutter speed doubles motion blur

From Photography, London et al.
Effect of shutter speed

• Freezing motion

Walking people
Running people
Car
Fast train

1/125 1/250 1/500 1/1000
Exposure

- Two main parameters:
  - Aperture (in f number)
  - Shutter speed (in fraction of a second)

- Exposure = irradiance x time
  \[ H = E \times T \]

- Irradiance (E)
  - controlled by aperture

- Exposure time (T)
  - controlled by shutter
Reciprocity

• Reciprocity

The same exposure is obtained with an exposure twice as long and an aperture area half as big

– Hence square root of two progression of f stops vs. power of two progression of shutter speed

From Photography, London et al.
Reciprocity

• Assume we know how much light we need
• We have the choice of an infinity of shutter speed/aperture pairs

![Shutter speed/aperture pairs]

• What will guide our choice of a shutter speed?
  – Freeze motion vs. motion blur, camera shake
• What will guide our choice of an aperture?
  – Depth of field, distortion reduction, diffraction limit
• Often we must compromise
  – Open more to enable faster speed (but shallow DoF)
Small aperture (deep depth of field), slow shutter speed (motion blurred). In this scene, a small aperture (f/16) produced great depth of field; the nearest paving stones as well as the farthest trees are sharp. But to admit enough light, a slow shutter speed (1/8 sec) was needed; it was too slow to show moving pigeons successfully. It also meant that a tripod had to be used to hold the camera steady.

From Photography, London et al.
Medium aperture (moderate depth of field), medium shutter speed (some motion sharp). A medium aperture (f/4) and shutter speed (1/125 sec) sacrifice some background detail to produce recognizable images of the birds. But the exposure is still too long to show the motion of the birds’ wings sharply.
Large aperture (shallow depth of field), fast shutter speed (motion sharp). A fast shutter speed (1/500 sec) stops the motion of the pigeons so completely that the flapping wings are frozen. But the wide aperture (f/2) needed gives so little depth of field that the background is now out of focus.

From Photography, London et al.
Metering

- Photosensitive sensors measure scene luminance
- Usually TTL (through the lens)
- Simple version: center-weighted average

- Assumption? Failure cases?
  - Usually assumes that a scene is 18% gray
  - Problem with dark and bright scenes
White polar bear given exposure suggested by meter

White polar bear given 2 stops more exposure

Gray elephant given exposure suggested by meter

Black gorilla given exposure suggested by meter

Black gorilla given 2 stops less exposure
Exposure & Metering

• The camera metering system measures how bright the scene is

• In **Aperture priority mode**, the photographer sets the aperture, the camera sets the shutter speed

• In **Shutter-speed priority mode**, the photographers sets the shutter speed and the camera deduces the aperture
  – In both cases, reciprocity is exploited

• In **Program mode**, the camera decides both exposure and shutter speed (middle value more or less)

• In **Manual**, the user decides everything (but can get feedback)
Pros and cons of various modes

• Aperture priority
  – Direct depth of field control
  – Cons: can require impossible shutter speed (e.g. with f/1.4 for a bright scene)

• Shutter speed priority
  – Direct motion blur control
  – Cons: can require impossible aperture (e.g. when requesting a 1/1000 speed for a dark scene)
    • Note that aperture is somewhat more restricted

• Program
  – Almost no control, but no need for neurons

• Manual
  – Full control, but takes more time and thinking
Recap

- **focal length**
- **focus distance**
- **sensor size**
- **depth of field**
- **field of view**
- **aperture**
- **lens**
Focal length

<30mm: wide angle
50mm: standard
>100mm telephoto

Affected by sensor size (crop factor)
Exposure

- Aperture (f number)
  - Expressed as ratio between focal length and aperture diameter: \( \text{diameter} = \frac{f}{\text{<f number>}} \)
  - f/2.0, f/2.8, f/4.0, f/5.6, f/8.0, f/11, f/16 (factor of \( \sqrt{2} \))
  - Small f number means large aperture
  - Main effect: depth of field
  - A good standard lens has max aperture f/1.8.
  - A cheap zoom has max aperture f/3.5

- Shutter speed
  - In fraction of a second
  - 1/30, 1/60, 1/125, 1/250, 1/500 (factor of 2)
  - Main effect: motion blur

- Sensitivity
  - Gain applied to sensor
  - In ISO, bigger number, more sensitive (100, 200, 400, 800, 1600)
  - Main effect: sensor noise

Reciprocity between these three numbers:
for a given exposure, one has two degrees of freedom.
Sensor Chip
Image Formation

Illumination (energy) source

Scene element

Imaging system

(Internal) image plane
Digital camera

A digital camera replaces film with a sensor array

- Each cell in the array is light-sensitive diode that converts photons to electrons
- Two common types: Charge Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS)
Sensor Array

FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.
The raster image (pixel matrix)
Interlace vs. progressive scan

Progressive scan

Interlace

Color Image
Images in Matlab

- Images represented as a matrix
- Suppose we have a NxM RGB image called “im”
  - $\text{im}(1,1,1)$ = top-left pixel value in R-channel
  - $\text{im}(y, x, b)$ = $y$ pixels down, $x$ pixels to right in the $b^{th}$ channel
  - $\text{im}(N, M, 3)$ = bottom-right pixel in B-channel
- `imread(filename)` returns a uint8 image (values 0 to 255)
  - Convert to double format (values 0 to 1) with `im2double`

```
0.92 0.93 0.94 0.97 0.62 0.37 0.85 0.97 0.93 0.92 0.99
0.95 0.89 0.82 0.89 0.56 0.31 0.75 0.92 0.81 0.95 0.91
0.89 0.72 0.51 0.55 0.51 0.42 0.57 0.41 0.49 0.91 0.92
0.96 0.95 0.88 0.94 0.56 0.46 0.91 0.87 0.90 0.97 0.95
0.71 0.81 0.81 0.87 0.57 0.37 0.80 0.88 0.89 0.79 0.85
0.49 0.62 0.60 0.58 0.50 0.60 0.58 0.50 0.61 0.45 0.33
0.86 0.84 0.74 0.58 0.51 0.39 0.73 0.92 0.91 0.49 0.74
0.96 0.67 0.54 0.85 0.48 0.37 0.88 0.90 0.94 0.82 0.93
0.69 0.49 0.56 0.66 0.43 0.42 0.77 0.73 0.71 0.90 0.99
0.79 0.73 0.90 0.67 0.33 0.61 0.69 0.79 0.73 0.93 0.97
0.91 0.94 0.89 0.49 0.41 0.78 0.78 0.77 0.89 0.99 0.93
0.79 0.73 0.90 0.67 0.33 0.61 0.69 0.79 0.73 0.93 0.97
0.91 0.94 0.89 0.49 0.41 0.78 0.78 0.77 0.89 0.99 0.93
0.79 0.73 0.90 0.67 0.33 0.61 0.69 0.79 0.73 0.93 0.97
0.91 0.94 0.89 0.49 0.41 0.78 0.78 0.77 0.89 0.99 0.93
0.79 0.73 0.90 0.67 0.33 0.61 0.69 0.79 0.73 0.93 0.97
0.91 0.94 0.89 0.49 0.41 0.78 0.78 0.77 0.89 0.99 0.93
```
CCD color sampling

• Problem: a photosite can record only one number
• We need 3 numbers for color
Some approaches to color sensing

• Scan 3 times (temporal multiplexing)
  – Drum scanners
  – Flat-bed scanners
  – Russian photographs from 1800’s

• Use 3 detectors
  – High-end 3-tube or 3-ccd video cameras

• Use spatially offset color samples (spatial multiplexing)
  – Single-chip CCD color cameras
  – Human eye
3 CCD Sensor

- 3-chip vs. 1-chip: quality vs. cost
Spatial Multiplexing: Bayer Grid

• Why more green?
  – We have 3 channels and square lattice doesn’t like odd numbers
  – It’s the spectrum “in the middle”
  – More important to human perception of brightness
Practical Color Sensing: Bayer Grid
Recap: Camera sensor

RGB Inside the Camera

- Incoming Visible light
- Visible Light passes through IR-Blocking Filter
- Color Filters control the color light reaching a sensor
- Color blind sensors convert light reaching each sensor into electricity

Sensor Chip: Gain

Diagram showing the process from camera irradiance to JPEG output, highlighting the sensor chip and gain stage.
Sensitivity (ISO)

- Third variable for exposure: gain applied to sensor
- Linear effect (200 ISO needs half the light as 100 ISO)
- Film photography: trade sensitivity for grain

Digital photography: trade sensitivity for noise
Light and Shading

Slides by D. Hoiem
• What determines a pixel’s intensity?
• What can we infer about the scene from pixel intensities?
How does a pixel get its value?

Light emitted

Fraction of light reflects into camera

Sensor

Lens
How does a pixel get its value?

- Major factors
  - Illumination strength and direction
  - Surface geometry
  - Surface material
  - Nearby surfaces
  - Camera gain/exposure
Basic models of reflection

• Specular: light bounces off at the incident angle
  – E.g., mirror

• Diffuse: light scatters in all directions
  – E.g., brick, cloth, rough wood
Lambertian reflectance model

- Some light is absorbed (function of albedo $\rho$)
- Remaining light is scattered (diffuse reflection)
- Examples: soft cloth, concrete, matte paints
Diffuse reflection: Lambert’s cosine law

Intensity does *not* depend on viewer angle.
- Amount of reflected light proportional to $\cos(\theta)$
- Visible solid angle also proportional to $\cos(\theta)$
Most surfaces have both specular and diffuse components

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

Photo: northcountryhardwoodfloors.com

Typically, specular component is small
Intensity and Surface Orientation

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

\[ I(x) = \rho(x)(S \cdot N(x)) \]

\( \rho \) = albedo
\( S \) = directional source
\( N \) = surface normal
\( I \) = reflected intensity
Recap

• When light hits a typical surface
  – Some light is absorbed \((1-\rho)\)
    • More absorbed for low albedos
  – Some light is reflected diffusely
    • Independent of viewing direction
  – Some light is reflected specularly
    • Light bounces off (like a mirror), depends on viewing direction
Other possible effects

transparency

light source

refraction

light source
fluorescence

light source

\( \lambda_1 \)

\( \lambda_2 \)

phosphorescence

light source

t=1

t>1
subsurface scattering
BRDF: Bidirectional Reflectance Distribution Function

- Model of local reflection that tells how bright a surface appears when viewed from one direction when light falls on it from another
  - Ratio of measured outgoing radiance in direction $(\theta_e, \phi_e)$ to irradiance from direction $(\theta_i, \phi_i)$
  - Reciprocal

$$\rho(\theta_i, \phi_i, \theta_e, \phi_e; \lambda) = \frac{L_e(\theta_e, \phi_e)}{E_i(\theta_i, \phi_i)} = \frac{L_e(\theta_e, \phi_e)}{L_i(\theta_i, \phi_i) \cos \theta_i d\omega}$$
Color

Light is composed of a spectrum of wavelengths

Human Luminance Sensitivity Function

Slide Credit: Efros

http://www.yorku.ca/eye/photopik.htm
Some examples of the reflectance spectra of surfaces
The color of objects

- Colored light arriving at the camera involves two effects
  - The color of the light source (illumination + inter-reflections)
  - The color of the surface

\[
\int_{\lambda} \sigma(\lambda) \rho(\lambda) E(\lambda) d\lambda
\]

Receptor response of k'th receptor class

Incoming spectral radiance \( E(\lambda) \)

Outgoing spectral radiance \( E(\lambda) \rho(\lambda) \)

Spectral albedo \( \rho(\lambda) \)
Why RGB?

If light is a spectrum, why are images RGB?
Human color receptors

- Long (red), Medium (green), and Short (blue) cones, plus intensity rods
- Fun facts
  - “M” and “L” on the X-chromosome
    - That’s why men are more likely to be color blind
  - “L” has high variation, so some women are tetrachromatic
  - Some animals have 1 (night animals), 2 (e.g., dogs), 4 (fish, birds), 5 (pigeons, some reptiles/amphibians), or even 12 (mantis shrimp) types of cones

http://en.wikipedia.org/wiki/Color_vision
Color spaces: RGB

Default color space

Some drawbacks
• Strongly correlated channels
• Non-perceptual

Color spaces: HSV

Intuitive color space

H (S=1, V=1)

S (H=1, V=1)

V (H=1, S=0)
Color spaces: YCbCr

Fast to compute, good for compression, used by TV

\[ Y' = 16 + \frac{65.738 \cdot R'_D}{256} + \frac{129.057 \cdot G'_D}{256} + \frac{25.064 \cdot B'_D}{256} \]
\[ C_B = 128 + \frac{-37.945 \cdot R'_D}{256} - \frac{74.494 \cdot G'_D}{256} + \frac{112.439 \cdot B'_D}{256} \]
\[ C_R = 128 + \frac{112.439 \cdot R'_D}{256} - \frac{94.154 \cdot G'_D}{256} - \frac{18.285 \cdot B'_D}{256} \]
Color spaces: CIE L*a*b*

“Perceptually uniform” color space

Luminance = brightness
Chrominance = color
Which contains more information?
(a) intensity (1 channel)
(b) chrominance (2 channels)
Most information in intensity

Only color shown – constant intensity
Most information in intensity

Only intensity shown – constant color
Most information in intensity

Original image
So far: light $\rightarrow$ surface $\rightarrow$ camera

- Called a local illumination model
- But much light comes from surrounding surfaces

From Koenderink slides on image texture and the flow of light
Inter-reflection is a major source of light
Inter-reflection affects the apparent color of objects.
Scene surfaces also cause shadows

- Shadow: reduction in intensity due to a blocked source
Shadows

Point Source

Cast Shadow Boundary

Self Shadow Boundary

Area Source

Occluder

1 2 3

1 2 3
Models of light sources

• Distant point source
  – One illumination direction
  – E.g., sun

• Area source
  – E.g., white walls, diffuser lamps, sky

• Ambient light
  – Substitute for dealing with interreflections

• Global illumination model
  – Account for interreflections in modeled scene
The plight of the poor pixel

- A pixel’s brightness is determined by
  - Light source (strength, direction, color)
  - Surface orientation
  - Surface material and albedo
  - Reflected light and shadows from surrounding surfaces
  - Gain on the sensor

- A pixel’s brightness tells us nothing by itself
And yet we can interpret images...

- Key idea: for nearby scene points, most factors do not change much
- The information is mainly contained in *local differences* of brightness
Darkness = Large Difference in Neighboring Pixels
What is this?
What differences in intensity tell us about shape

- Changes in surface normal
- Texture
- Proximity
- Indents and bumps
- Grooves and creases

Photos Koenderink slides on image texture and the flow of light
Color constancy

• Interpret surface in terms of albedo or “true color”, rather than observed intensity
  – Humans are good at it
  – Computers are not nearly as good
One source of constancy: local comparisons
http://www.echalk.co.uk/amusements/OpticalIllusions/colourPerception/colourPerception.html
Things to remember

• Important terms: diffuse/specular reflectance, albedo, umbra/penumbra

• Observed intensity depends on light sources, geometry/material of reflecting surface, surrounding objects, camera settings

• Objects cast light and shadows on each other

• Differences in intensity are primary cues for shape
Pixels and Linear Filters

Slides by D. Hoiem
The raster image (pixel matrix)
Image filtering

• Image filtering: for each pixel, compute function of local neighborhood and output a new value
  – Same function applied at each position
  – Output and input image are typically the same size
Image filtering

- Linear filtering: function is a weighted sum/difference of pixel values

- Really important
  - Enhance images
    - Denoise, smooth, increase contrast, etc.
  - Extract information from images
    - Texture, edges, distinctive points, etc.
  - Detect patterns
    - Template matching
Example: box filter

\[
g[\cdot, \cdot] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\]
Image filtering

\[ f[\cdot,\cdot] \]

\[ h[\cdot,\cdot] \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]
Image filtering

\[ f[\cdot, \cdot] \]

\[ h[\cdot, \cdot] \]

\[ h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l] \]

Credit: S. Seitz
Image filtering

\[ f[\cdot,\cdot] \]

\[ h[\cdot,\cdot] \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]
Image filtering

\[
f[\cdot, \cdot] = h[\cdot, \cdot]
\]

\[
h[m, n] = \sum_{k, l} g[k, l] f[m + k, n + l]
\]

Credit: S. Seitz
Image filtering

\[ f[\cdot, \cdot] \]

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

\[ h[\cdot, \cdot] = \sum_{k,l} g[k,l] f[m+k, n+l] \]

Credit: S. Seitz
Image filtering

$$f[\cdot,\cdot]$$

$$h[\cdot,\cdot]$$

$$h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]$$

Credit: S. Seitz
Image filtering

\[ f[\cdot,\cdot] \quad g[\cdot,\cdot] = \frac{1}{9} \]

\[ h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l] \]

Credit: S. Seitz
Image filtering

\[ f[\cdot, \cdot] \]

\[ g[\cdot, \cdot] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \]

\[ h[m, n] = \sum_{k, l} g[k, l] f[m + k, n + l] \]

Credit: S. Seitz
Box Filter

What does it do?

- Replaces each pixel with an average of its neighborhood
- Achieve smoothing effect (remove sharp features)

\[ g[\cdot, \cdot] \]

\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]
Smoothing with box filter
Practice with linear filters

Original

Source: D. Lowe
Practice with linear filters

Original

Filtered (no change)

Source: D. Lowe
Practice with linear filters

Original

Source: D. Lowe
Practice with linear filters

Original

Shifted left
By 1 pixel

Source: D. Lowe
Practice with linear filters

Original

(Note that filter sums to 1)

Source: D. Lowe
Practice with linear filters

Original

Sharpening filter
- Accentuates differences with local average

Source: D. Lowe
Sharpening

before

after

Source: D. Lowe
Other filters

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

Sobel

Vertical Edge
(absolute value)
Other filters

Sobel

Horizontal Edge
(absolute value)
Basic gradient filters

Horizontal Gradient

\[
\begin{bmatrix}
0 & 0 & 0 \\
-1 & 0 & 1 \\
0 & 0 & 0
\end{bmatrix}
\]

or

\[
\begin{bmatrix}
-1 & 0 & 1 \\
-1 & 0 & 1
\end{bmatrix}
\]

Vertical Gradient

\[
\begin{bmatrix}
0 & -1 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\]

or

\[
\begin{bmatrix}
-1 \\
0 \\
1
\end{bmatrix}
\]
Examples

Write as filtering operations, plus some pointwise operations: +, -, .*,
Filtering vs. Convolution

• 2d filtering
  \[ h = \text{filter2}(g, f); \quad \text{or} \]
  \[ h = \text{imfilter}(f, g); \]
  \[ h[m, n] = \sum_{k, l} g[k, l] f[m + k, n + l] \]

• 2d convolution
  \[ h = \text{conv2}(g, f); \]
  \[ h[m, n] = \sum_{k, l} g[k, l] f[m - k, n - l] \]
Key properties of linear filters

**Linearity:**
\[ \text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2) \]

**Shift invariance:** same behavior regardless of pixel location
\[ \text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f)) \]

Any linear, shift-invariant operator can be represented as a convolution

Source: S. Lazebnik
More properties

• Commutative: \( a * b = b * a \)
  – Conceptually no difference between filter and signal

• Associative: \( a * (b * c) = (a * b) * c \)
  – Often apply several filters one after another: \(((a * b_1) * b_2) * b_3)\)
  – This is equivalent to applying one filter: \(a * (b_1 * b_2 * b_3)\)

• Distributes over addition: \( a * (b + c) = (a * b) + (a * c) \)

• Scalars factor out: \(ka * b = a * kb = k(a * b)\)

• Identity: unit impulse \(e = [0, 0, 1, 0, 0]\), \(a * e = a\)

Source: S. Lazebnik
Important filter: Gaussian

- Spatially-weighted average

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

5 x 5, $\sigma = 1$

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.003</td>
<td>0.013</td>
<td>0.022</td>
<td>0.013</td>
<td>0.003</td>
</tr>
<tr>
<td>0.013</td>
<td>0.059</td>
<td>0.097</td>
<td>0.059</td>
<td>0.013</td>
</tr>
<tr>
<td>0.022</td>
<td>0.097</td>
<td>0.159</td>
<td>0.097</td>
<td>0.022</td>
</tr>
<tr>
<td>0.013</td>
<td>0.059</td>
<td>0.097</td>
<td>0.059</td>
<td>0.013</td>
</tr>
<tr>
<td>0.003</td>
<td>0.013</td>
<td>0.022</td>
<td>0.013</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Slide credit: Christopher Rasmussen
Smoothing with Gaussian filter
Smoothing with box filter
Gaussian filters

- Remove “high-frequency” components from the image (low-pass filter)
  - Images become more smooth
- Convolution with self is another Gaussian
  - So can smooth with small-width kernel, repeat, and get same result as larger-width kernel would have
  - Convolving two times with Gaussian kernel of width $\sigma$ is same as convolving once with kernel of width $\sigma\sqrt{2}$
- *Separable* kernel
  - Factors into product of two 1D Gaussians

Source: K. Grauman
Separability of the Gaussian filter

\[ G_\sigma(x, y) = \frac{1}{2\pi \sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \]

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

The 2D Gaussian can be expressed as the product of two functions, one a function of \( x \) and the other a function of \( y \).

In this case, the two functions are the (identical) 1D Gaussian.

Source: D. Lowe
Separability example

2D filtering (center location only)

The filter factors into a product of 1D filters:

Perform filtering along rows:

Followed by filtering along the remaining column:

Source: K. Grauman
Separability

• Why is separability useful in practice?
Practical matters

How big should the filter be?

- Values at edges should be near zero ← important!
- Rule of thumb for Gaussian: set filter half-width to about $3\sigma$
Practical matters

• What about near the edge?
  – the filter window falls off the edge of the image
  – need to extrapolate
  – methods:
    • clip filter (black)
    • wrap around
    • copy edge
    • reflect across edge

Source: S. Marschner
Practical matters

– methods (MATLAB):
  • clip filter (black): \texttt{imfilter(f, g, 0)}
  • wrap around: \texttt{imfilter(f, g, ‘circular’)}
  • copy edge: \texttt{imfilter(f, g, ‘replicate’)}
  • reflect across edge: \texttt{imfilter(f, g, ‘symmetric’)}

Source: S. Marschner
Practical matters

• What is the size of the output?
• MATLAB: filter2(g, f, \textit{shape})
  – \textit{shape} = ‘full’: output size is sum of sizes of f and g
  – \textit{shape} = ‘same’: output size is same as f
  – \textit{shape} = ‘valid’: output size is difference of sizes of f and g

Source: S. Lazebnik
Slide Credits

• This set of sides contains contributions kindly made available by the following authors
  – Gianfranco Doretto
  – Derek Hoiem
  – Alexei Efros
  – Svetlana Lazebnik
  – Kristen Grauman
  – Frédo Durand
  – David Lowe
  – Steve Marschner
  – Steve Seitz
  – Richard Szeliski