

Filtering

Introduction to Computer Vision
CSE 152
Lecture 8

CSE152, Winter 2013

Intro Computer Vision

Announcements

- HW 1 is due today
- HW2 will be available in the next couple days.
- See links on web page for reading on binary image processing (e-reserves)
- Midterm February 14, in class

CSE152, Winter 2013

Intro Computer Vision

Binary System Summary

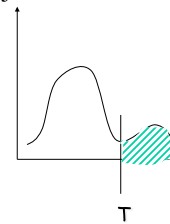
1. Acquire images and binarize (thresholding, color labels, etc.).
2. Possibly clean up image using morphological operators.
3. Determine regions (blobs) using connected component exploration
4. Compute position, area, and orientation of each blob using moments
5. Compute features that are rotation, scale, and translation invariant using Moments (e.g., Eigenvalues of normalized moments).

CSE152, Spr 2010

Intro Computer Vision

P-Tile Method

- If the *size* of the object is approx. known, pick T s.t. the area under the histogram corresponds to the size of the object:



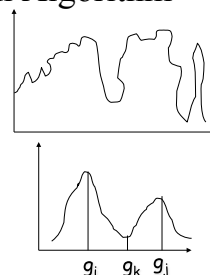
CSE152, Spr 2010

[From Octavia Camps]

Intro Computer Vision

“Peakiness” Detection Algorithm

- Find the two **HIGHEST LOCAL MAXIMA** that **MINIMUM DISTANCE APART**: g_i and g_j
- Find **lowest point** between them: g_k
- Measure “peakiness”:
 - $\min(H(g_i), H(g_j)) / H(g_k)$
- Find (g_i, g_j, g_k) with highest peakiness

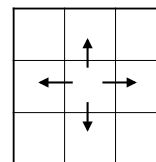


CSE152, Spr 2010

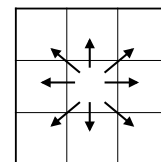
[From Octavia Camps]

Intro Computer Vision

Four & Eight Connectedness



Four Connected

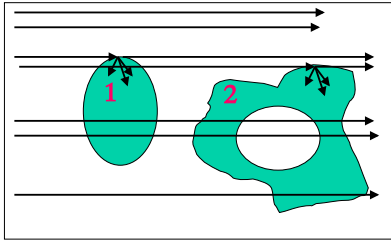


Eight Connected

CSE152, Spr 2010

Intro Computer Vision

Recursive Labeling Connected Component Exploration



CSE152, Spr 2010

Intro Computer Vision

Properties extracted from binary image

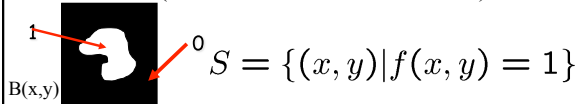
- A tree showing containment of regions
- Properties of a region
 1. Genus – number of holes
 2. Centroid
 3. Area
 4. Perimeter
 5. Moments (e.g., measure of elongation)
 6. Number of “extrema” (indentations, bulges)
 7. Skeleton

CSE152, Spr 2010

Intro Computer Vision

Moments

(related to moments of inertia)



Given a pair of non-negative integers (j,k) the discrete (j,k)th moment of S is :

$$M_{jk}(S) = \sum_{(x,y) \in S} x^j y^k$$

The order of the M_{jk} moment is j + k.

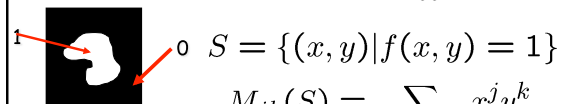
$$M_{j,k} = \sum_{x=1}^n \sum_{y=1}^m B(x,y) x^j y^k$$

- Fast way to implement computation over n by m image or window
- One object

CSE152, Spr 2010

Intro Computer Vision

Area: Moment M_{00}



$$M_{jk}(S) = \sum_{(x,y) \in S} x^j y^k$$

Example:

$$M_{00}(S) = \sum_{(x,y) \in S} x^0 y^0 = \sum_{(x,y) \in S} 1 = \#(S)$$

Area of S !!

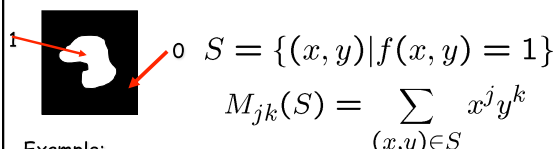
$$M_{0,0} = \sum_{x=1}^n \sum_{y=1}^m B(x,y)$$

- Fast way to implement computation over n by m image or window
- One object

CSE152, Spr 2010

Intro Computer Vision

Moments: Centroid



$$M_{jk}(S) = \sum_{(x,y) \in S} x^j y^k$$

Example:

$$M_{10}(S) = \sum_{(x,y) \in S} x^1 y^0 = \sum_{(x,y) \in S} x \quad M_{01}(S) = \sum_{(x,y) \in S} x^0 y^1 = \sum_{(x,y) \in S} y$$

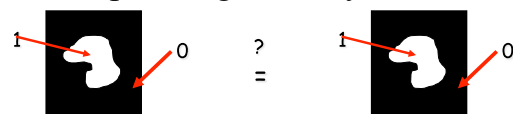
$$\frac{M_{10}(S)}{M_{00}(S)} = \frac{\sum_{(x,y) \in S} x}{\#(S)} = \bar{x} \quad \frac{M_{01}(S)}{M_{00}(S)} = \frac{\sum_{(x,y) \in S} y}{\#(S)} = \bar{y}$$

Center of gravity (centroid) of S !!

CSE152, Spr 2010

Intro Computer Vision

Shape recognition by Moments



Recognition could be done by comparing moments

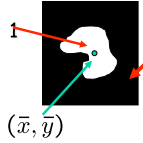
However, moments M_{jk} are not invariant under:

- Translation
- Scaling
- Rotation
- Skewing

CSE152, Spr 2010

Intro Computer Vision

Central Moments



$S = \{(x, y) | f(x, y) = 1\}$

$$\bar{x} = \frac{M_{10}(S)}{M_{00}(S)} \quad \bar{y} = \frac{M_{01}(S)}{M_{00}(S)}$$

Given a pair of non-negative integers (j,k) the central (j,k)th moment of S is given by:

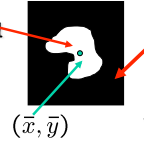
$$\mu_{jk}(S) = \sum_{(x,y) \in S} (x - \bar{x})^j (y - \bar{y})^k$$

Or the central moments can be computed from precomputed regular moments

$$\mu_{jk} = \sum_{m=1}^j \sum_{n=1}^k \binom{j}{m} \binom{k}{n} (-\bar{x})^{(j-m)} (-\bar{y})^{(k-n)} M_{mn}$$

CSE152, Spr 2010 Intro Computer Vision

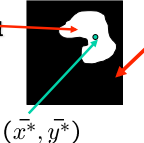
Central Moments



$S = \{(x, y) | f(x, y) = 1\}$

$$\mu_{jk}(S) = \sum_{(x,y) \in S} (x - \bar{x})^j (y - \bar{y})^k$$

Translation by $T = (a, b)$:

$$S_T = \{(x^*, y^*) | x^* = x + a, y^* = y + b, (x, y) \in S\}$$


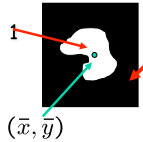
$$\bar{x}^* = \frac{M_{10}(S_T)}{M_{00}(S_T)} = \bar{x} + a \quad \bar{y}^* = \frac{M_{01}(S_T)}{M_{00}(S_T)} = \bar{y} + b$$

$$\mu_{jk}(S_T) = \mu_{jk}(S)$$

Translation INVARIANT!

CSE152, Spr 2010 Intro Computer Vision

Normalized Moments



$S = \{(x, y) | f(x, y) = 1\}$

$$\mu_{jk}(S) = \sum_{(x,y) \in S} (x - \bar{x})^j (y - \bar{y})^k$$

$$\sigma_x = \sqrt{\frac{\mu_{20}(S)}{M_{00}(S)}} \quad \sigma_y = \sqrt{\frac{\mu_{02}(S)}{M_{00}(S)}}$$

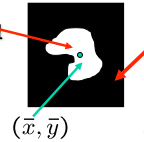
Given a pair of non-negative integers (j,k) the normalized (j,k)th moment of S is given by:

$$m_{jk}(S) = \sum_{(x,y) \in S} \left(\frac{x - \bar{x}}{\sigma_x} \right)^j \left(\frac{y - \bar{y}}{\sigma_y} \right)^k$$

Normalized moments are scale and translation invariant.

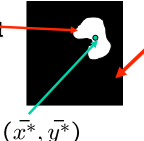
CSE152, Spr 2010 Intro Computer Vision

Normalized Moments



$S = \{(x, y) | f(x, y) = 1\}$

Scaling by (a,c) and translating by $T = (b, d)$:

$$S_{ST} = \{(x^*, y^*) | x^* = ax + b, y^* = cy + d, (x, y) \in S\}$$


$$m_{jk}(S_{ST}) = m_{jk}(S)$$

Scaling and translation INVARIANT!

CSE152, Spr 2010 Intro Computer Vision

Region orientation from Second Moment Matrix



(x-bar, y-bar)

1. Compute second centralized moment matrix

$$\begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \quad \begin{bmatrix} m_{20} & m_{11} \\ m_{11} & m_{02} \end{bmatrix}$$
 - Symmetric, positive definite matrix
 - Positive Eigenvalues
 - Orthogonal Eigenvectors
2. Compute Eigenvectors of Moment Matrix to obtain orientation
3. Eigenvalues are independent of orientation, translation!

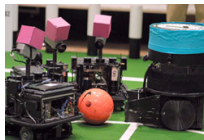
CSE152, Spr 2010 Intro Computer Vision

Moments

- Regular Moments M_{jk}
- Central Moments μ_{jk} : Translation invariant
- Normalized Moments m_{jk} : Translation and scale Invariant
- Eigenvalues of Second Moment Matrix: translation, scale, and rotation invariant.
- Hu Moments: Higher than second order, translation, rotation and scale invariant

CSE152, Spr 2010 Intro Computer Vision

Binarization using Color



- Object's in robocup are distinguished by color.
- How do you binarize the image so that pixels where ball is located are labeled with 1, and other locations are 0?
- Let $C_b = (r \ g \ b)^T$ be the color of the ball.

CSE152, Spr 2010

Intro Computer Vision

Binarization using Color

- Let $c(u,v)$ be the color of pixel (u,v)
- Simple method

$$b(u,v) = \begin{cases} 1 & \text{if } (\|c(u,v) - c_b\|)^2 \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$
- Better alternative (why?)
 - Convert $c(u,v)$ to HSV space $H(u,v)$, $S(u,v)$, $V(u,v)$
 - Convert c_b to HSV
 - Check that HS distance is less than threshold ϵ and brightness (V) is greater than a threshold $V > \tau$

CSE152, Spr 2010

Intro Computer Vision

Color Blob tracking

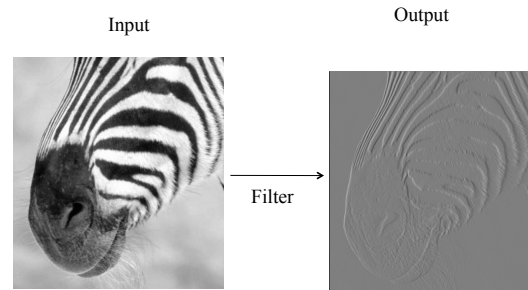


- Color-based tracker gets lost on white knight: Same Color

CSE152, Spr 2010

Intro Computer Vision

An example of a filtered image

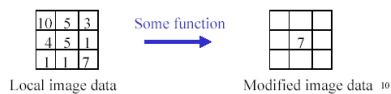


CSE152, Spr 2010

Intro Computer Vision

What is image filtering?

- Modify the pixels in an image based on some function of a local neighborhood of the pixels.



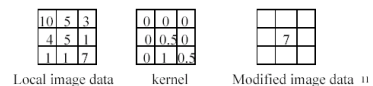
(From Bill Freeman)

CSE152, Spr 2010

Intro Computer Vision

Linear functions

- Simplest: linear filtering.
 - Replace each pixel by a linear combination of its neighbors.
- The prescription for the linear combination is called the "convolution kernel".



(Freeman)

CSE152, Spr 2010

Intro Computer Vision

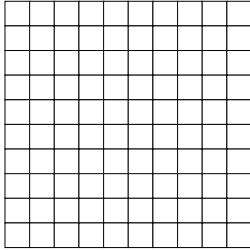
Smoothing by Averaging

Kernel: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$



CSE152, Spr 2010
Intro Computer Vision

Convolution



$*$

1	2	1
1	2	1
-1	-2	-1

Kernel (K)

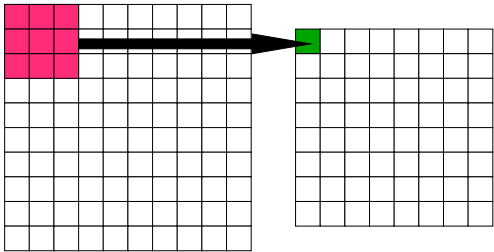
Note: Typically Kernel is relatively small in vision applications.

Image (I)

CSE152, Spr 2010
Intro Computer Vision

Convolution: $R = K * I$

$m=2$



I
R

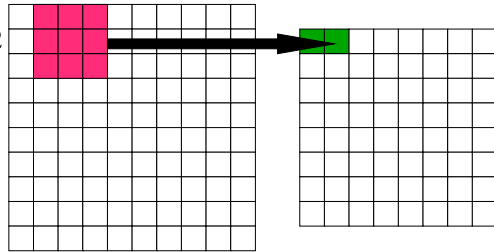
$$R(i, j) = \sum_{h=-m/2}^{m/2} \sum_{k=-m/2}^{m/2} K(h, k) I(i-h, j-k)$$

Kernel size is $m+1$ by $m+1$

CSE152, Spr 2010
Intro Computer Vision

Convolution: $R = K * I$

$m=2$



I
R

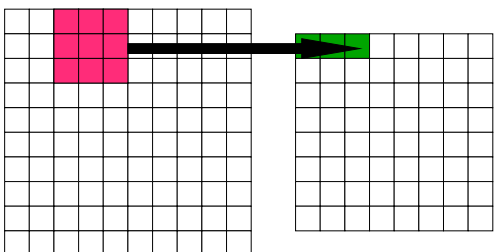
$$R(i, j) = \sum_{h=-m/2}^{m/2} \sum_{k=-m/2}^{m/2} K(h, k) I(i-h, j-k)$$

Kernel size is $m+1$ by $m+1$

CSE152, Spr 2010
Intro Computer Vision

Convolution: $R = K * I$

$m=2$



I
R

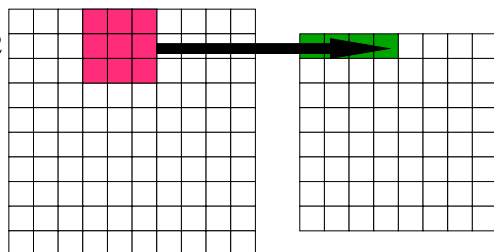
$$R(i, j) = \sum_{h=-m/2}^{m/2} \sum_{k=-m/2}^{m/2} K(h, k) I(i-h, j-k)$$

Kernel size is $m+1$ by $m+1$

CSE152, Spr 2010
Intro Computer Vision

Convolution: $R = K * I$

$m=2$

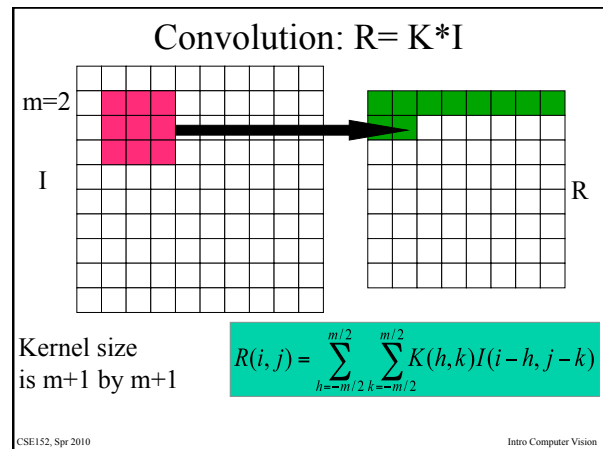
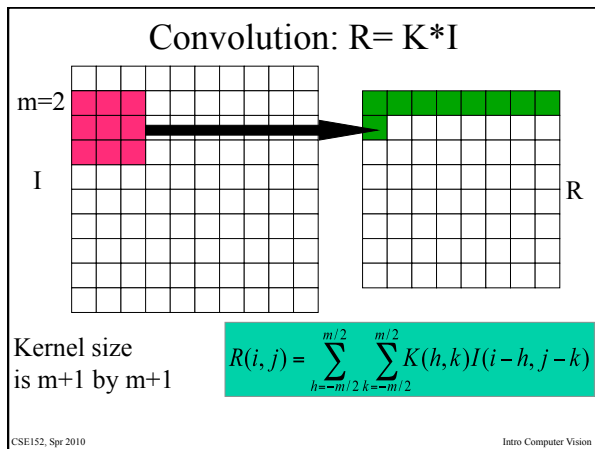
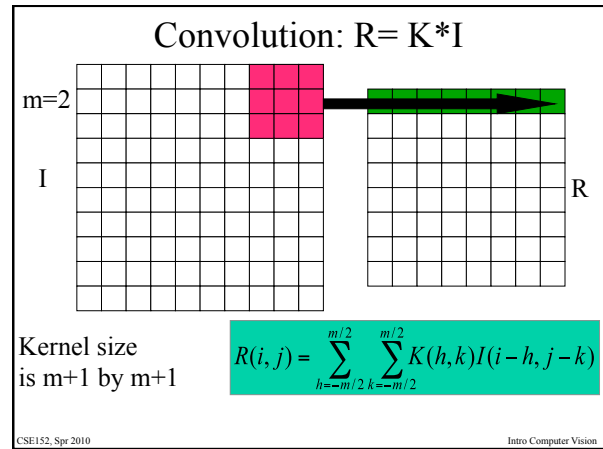
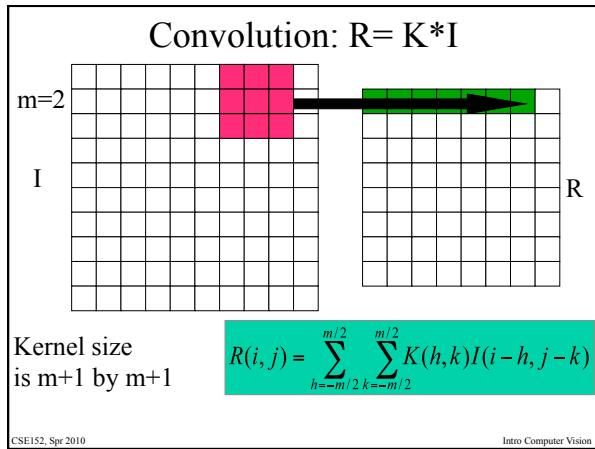
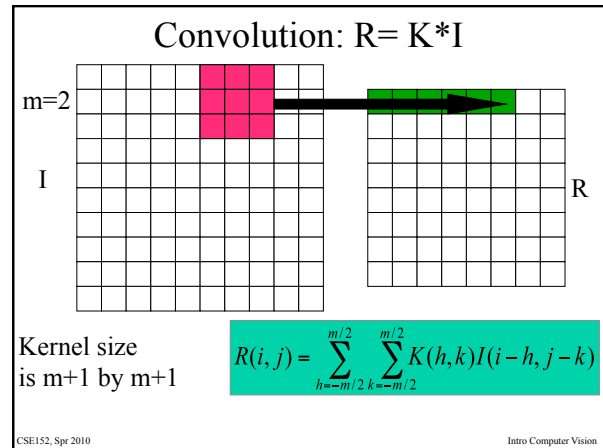
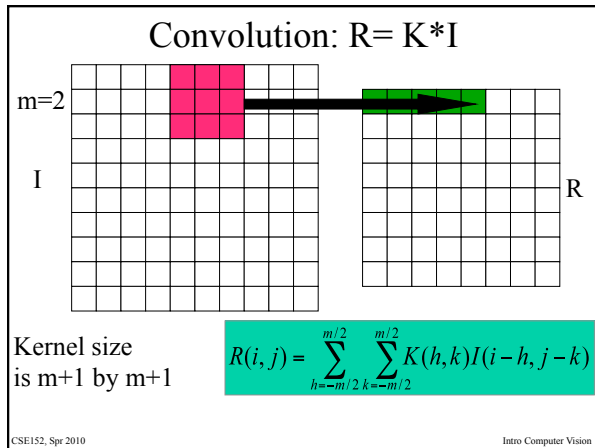


I
R

$$R(i, j) = \sum_{h=-m/2}^{m/2} \sum_{k=-m/2}^{m/2} K(h, k) I(i-h, j-k)$$

Kernel size is $m+1$ by $m+1$

CSE152, Spr 2010
Intro Computer Vision



Convolution: $R = K * I$

$m=2$

I

R

Kernel size is $m+1$ by $m+1$

$$R(i, j) = \sum_{h=-m/2}^{m/2} \sum_{k=-m/2}^{m/2} K(h, k) I(i-h, j-k)$$

CSE152, Spr 2010 Intro Computer Vision

Linear filtering (warm-up slide)



0	0	0
0	1	0
0	0	0

Pixel offset

?

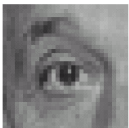
CSE152, Spr 2010 Intro Computer Vision

Linear filtering (warm-up slide)



0	0	0
0	1	0
0	0	0

Pixel offset



Filtered (no change)

CSE152, Spr 2010 Intro Computer Vision

Linear filtering



0	0	0
0	0	1
0	0	0

Pixel offset

?

CSE152, Spr 2010 Intro Computer Vision

shift



0	0	0
0	0	1
0	0	0

Pixel offset



Shifted one Pixel to the left

CSE152, Spr 2010 Intro Computer Vision

Linear filtering



1	1	1
1	1	1
1	1	1

$\frac{1}{9}$

?

CSE152, Spr 2010 Intro Computer Vision

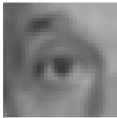
Blurring



original

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1



Blurred (filter applied in both dimensions).

CSE152, Spr 2010
(Swiped from Bill Freeman)
Intro Computer Vision

Blur examples

impulse



original

coefficient



Pixel offset

2.4



filtered

CSE152, Spr 2010
Intro Computer Vision

Blur examples

impulse



original

coefficient



Pixel offset

2.4



filtered

edge



original

coefficient



Pixel offset

8



filtered

CSE152, Spr 2010
Intro Computer Vision

Practice with linear filters



0	0	0
0	2	0
0	0	0

 $- \frac{1}{9}$

1	1	1
1	1	1
1	1	1

?

Original

CSE152, Spr 2010
Source: OpenStax

Practice with linear filters



0	0	0
0	2	0
0	0	0

 $- \frac{1}{9}$

1	1	1
1	1	1
1	1	1





Sharpening filter
- Accentuates differences with local average

CSE152, Spr 2010
Source: OpenStax

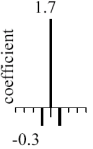
Sharpening example

8



original

coefficient



-0.3

11.2

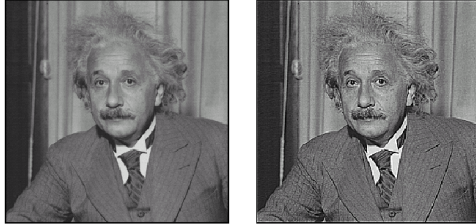


-0.25

Sharpened
(differences are accentuated; constant areas are left untouched).

CSE152, Spr 2010
Intro Computer Vision

Sharpening



before

after

CSE152, Spr 2010

Intro Computer Vision

Smoothing by Averaging

Kernel: 



CSE152, Spr 2010

Intro Computer Vision

Properties of Continuous Convolution (Same for discrete)

Let f, g, h be images and $*$ denote convolution

$$f * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x-u, y-v) g(u, v) du dv$$

- Commutative: $f * g = g * f$
- Associative: $f * (g * h) = (f * g) * h$
- Linear: for scalars a & b and images f, g, h
 $(af + bg) * h = a(f * h) + b(g * h)$
- Differentiation rule

$$\frac{\partial}{\partial x} (f * g) = \frac{\partial f}{\partial x} * g = f * \frac{\partial g}{\partial x}$$

CSE152, Spr 2010

Intro Computer Vision

Filtering to reduce noise

- Noise is what we're not interested in.
 - We'll discuss simple, low-level noise today: Light fluctuations; Sensor noise; Quantization effects; Finite precision
 - Not complex: shadows; extraneous objects.
- A pixel's neighborhood contains information about its intensity.
- Averaging noise reduces its effect.

CSE152, Spr 2010

Intro Computer Vision

Additive noise

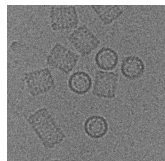
- $I = S + N$. Noise doesn't depend on signal.
- We'll consider:

$$I_i = s_i + n_i \text{ with } E(n_i) = 0$$

s_i deterministic. n_i a random var.

n_i, n_j independent for $i \neq j$

n_i, n_j identically distributed



CSE152, Spr 2010

Intro Computer Vision

Gaussian Noise:
sigma=1



Gaussian Noise:
sigma=16



CSE152, Spr 2010

Intro Computer Vision

Average Filter

- Mask with positive entries, that sum 1.
- Replaces each pixel with an average of its neighborhood.
- If all weights are equal, it is called a BOX filter.

$$\frac{1}{9} \begin{matrix} & \text{F} & \\ \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \end{matrix}$$

(Camps)

CSE152, Spr 2010

Intro Computer Vision

Smoothing by Averaging

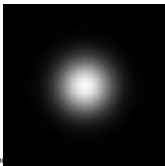
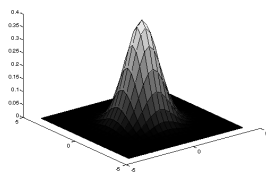
Kernel: 



CSE152, Spr 2010

Intro Computer Vision

An Isotropic Gaussian



- The picture shows a smoothing kernel proportional to

$$e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

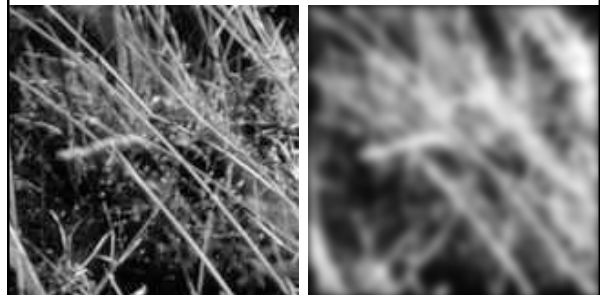
(which is a reasonable model of a circularly symmetric fuzzy blob)

CSE152, Spr 2010

Intro Computer Vision

Smoothing with a Gaussian

Kernel: 

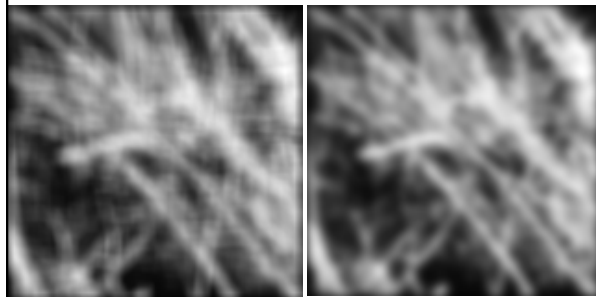


CSE152, Spr 2010

Intro Computer Vision

Smoothing with a Gaussian

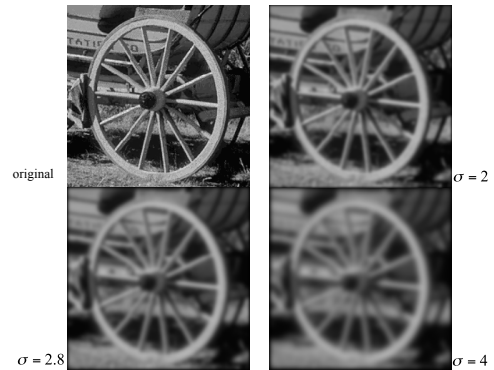
Kernel: 



CSE152, Spr 2010

Intro Computer Vision

Gaussian Smoothing



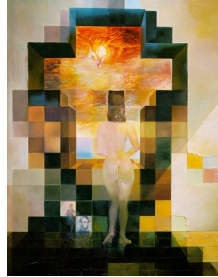
CSE152, Spr 2010

Intro Computer Vision

Gaussian Smoothing



by Charles Allen Gillbert



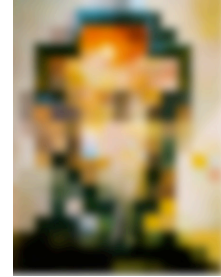
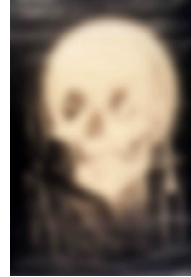
by Harmon & Julesz

CSE152, Spr 2010

http://www.michaelbach.de/ot/cog_blureffects/index.html

Intro Computer Vision

Gaussian Smoothing



http://www.michaelbach.de/ot/cog_blureffects/index.html

CSE152, Spr 2010

Intro Computer Vision

$\sigma=0.05$	$\sigma=0.1$	$\sigma=0.2$	
			no smoothing
			$\sigma=1$ pixel
			$\sigma=2$ pixels

The effects of smoothing
Each row shows smoothing with gaussians of different width; each column shows different realizations of an image of gaussian noise.

CSE152, Spr 2010

Intro Computer Vision

Efficient Implementations

Both, the BOX filter and the Gaussian filter are separable:

- First convolve each row with a 1-D filter
- Then convolve each column with a 1-D filter.

$$K_r = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$K_c = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

For Gaussian kernels $g_1(x)$ and $g_2(x)$,

- If g_1 & g_2 respectively have variance σ_1^2 & σ_2^2
- Then $g_1 * g_2$ has variance $\sigma_1^2 + \sigma_2^2$

CSE152, Spr 2010

Intro Computer Vision