Announcement

Quiz #8 is available in Blackboard.

Due date: 11:59pm EST, Thursday, April 10th

Open book and open notes

Notes for Project 2

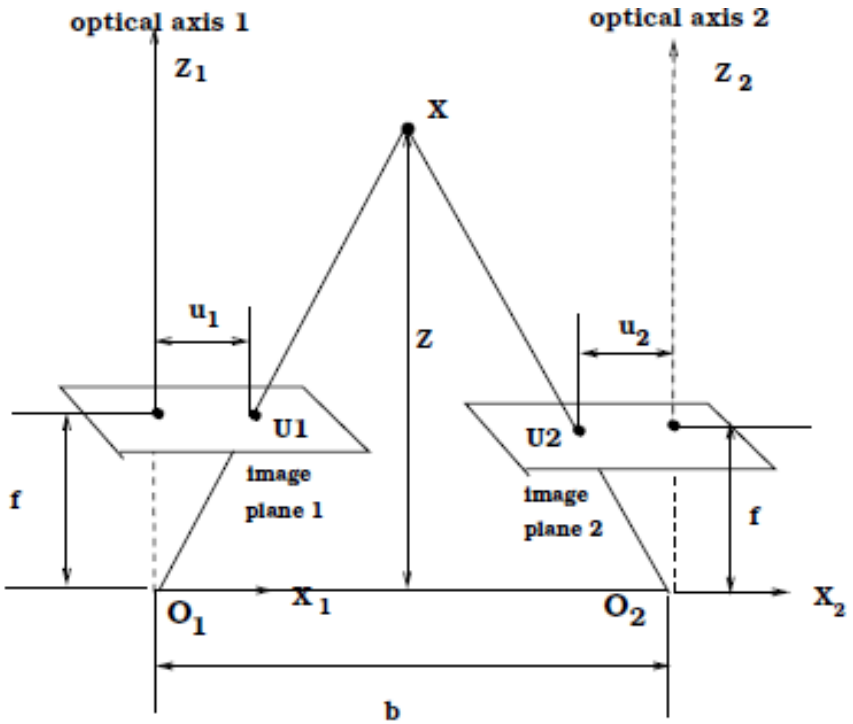
Draw epipolar lines for rectified images

$$\mathbf{F} = \mathbf{W_l}^{-\mathbf{t}}\mathbf{E}\mathbf{W_r}^{-\mathbf{1}}$$

F matrix should be updated for rectified images, where no rotation between two cameras.

$$\mathbf{E} = \mathbf{S}$$

$$\mathbf{F} = \mathbf{W_{rect}}^{-\mathbf{t}}\mathbf{S}\mathbf{W_{rect}}^{-\mathbf{1}}$$



courtesy of Dr. Qiang Ji

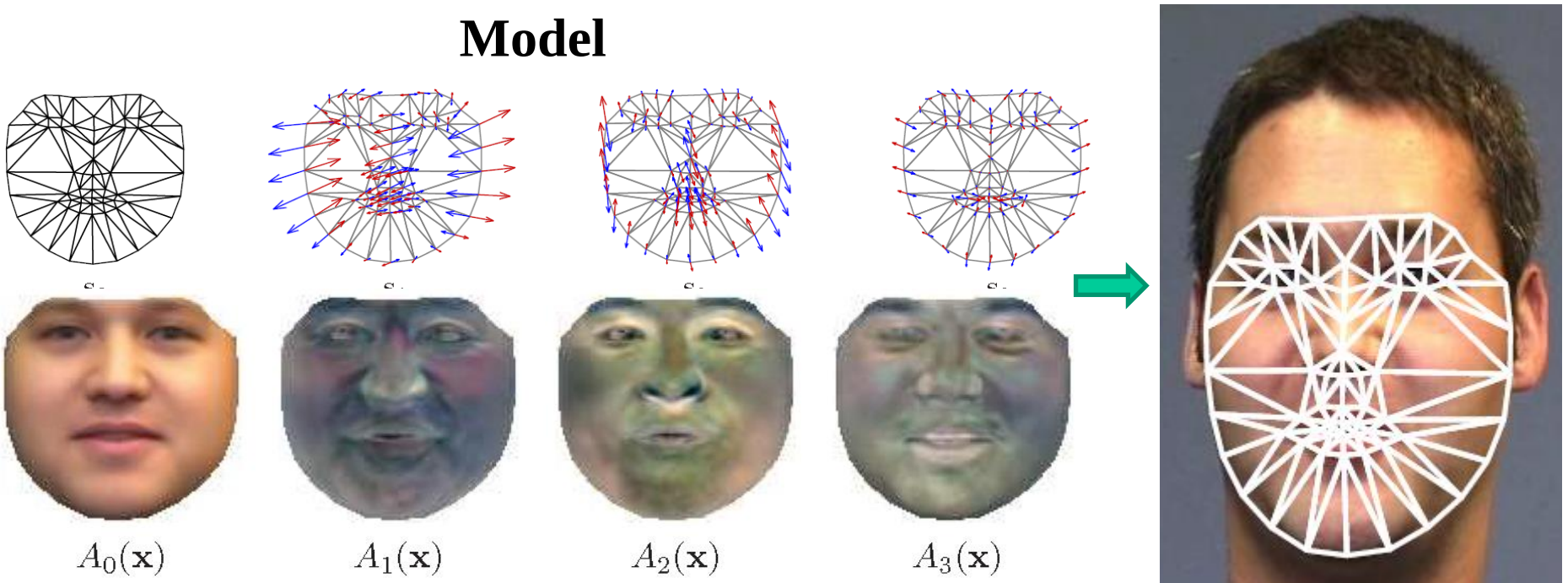
Today

Model-based image registration

- **Active Appearance Model**

Image segmentation

Active Appearance Model



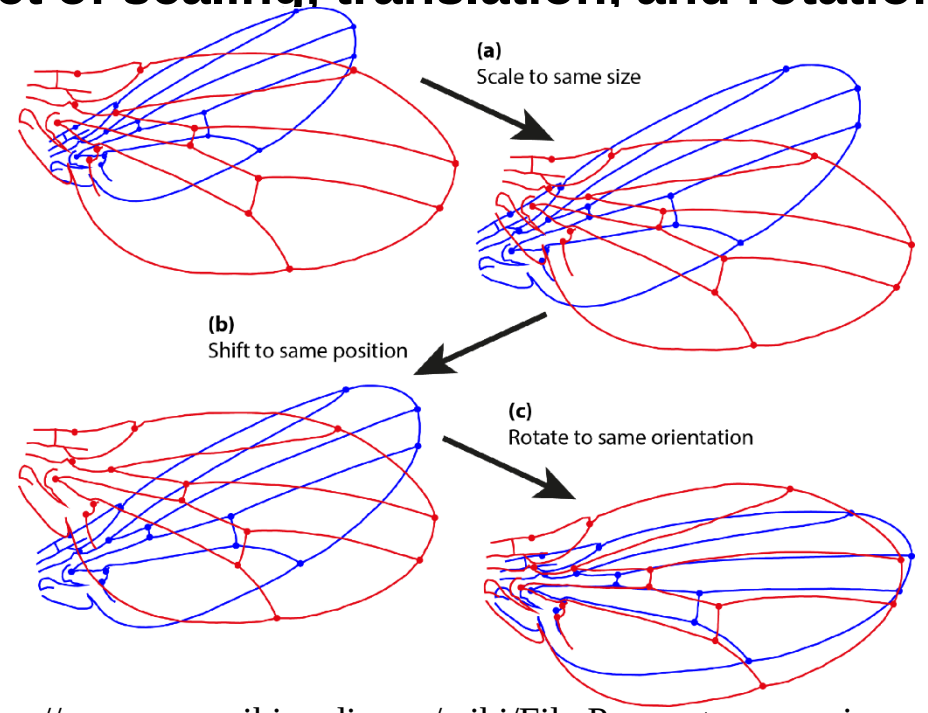
Adapted from Matthews and Baker, "Active Appearance Models Revisited", IJCV 2004

- For every triangle, an affine transformation is employed to established the correspondence between the model and the image
- A set of local affine transformation approximates nonrigid transformation ➡ **Piecewise affine transformation**

Training a Shape Model

Major steps:

- **Manually label landmarks for each training image**
- **Align training shapes by Procrustes Analysis to minimize the effect of scaling, translation, and rotation**



https://commons.wikimedia.org/wiki/File:Procrustes_superimposition.png

Training a Shape Model

Major steps:

- **Manually label landmarks for each training image**
- **Align training shapes by Procrustes Analysis to minimize the effect of scaling, translation, and rotation**

1. Translate each example so that its centre of gravity is at the origin.
2. Choose one example as an initial estimate of the mean shape and scale so that $|\bar{\mathbf{x}}| = 1$.
3. Record the first estimate as $\bar{\mathbf{x}}_0$ to define the default reference frame.
4. Align all the shapes with the current estimate of the mean shape.
5. Re-estimate mean from aligned shapes.
6. Apply constraints on the current estimate of the mean by aligning it with $\bar{\mathbf{x}}_0$ and scaling so that $|\bar{\mathbf{x}}| = 1$.
7. If not converged, return to **4**.

T. F. Cootes. Statistical models of appearance for computer vision

Training a Shape Model

Major steps:

- **Manually label landmarks for each training image**
- **Align training shapes by Procrustes Analysis to minimize the effect of scaling, translation, and rotation**
- **Perform PCA on the aligned shapes**
- **Retain the major variations**

Appearance Model

The appearance of an AAM

- characterizes the texture information of the object
- defined inside the mean shape

An appearance can be represented as a linear combination of a mean appearance and a set of appearance variations

$$\mathbf{A}(\mathbf{x}) = \mathbf{A_0}(\mathbf{x}) + \sum_{i=1}^{K} \lambda_i \, \mathbf{A_i}(\mathbf{x}) \longrightarrow \text{All pixels in } \mathbf{s_0}$$

$A_0(\mathbf{x})$ $A_1(\mathbf{x})$ $A_2(\mathbf{x})$ $A_3(\mathbf{x})$

Adapted from Matthews and Baker, "Active Appearance Models Revisited", IJCV 2004

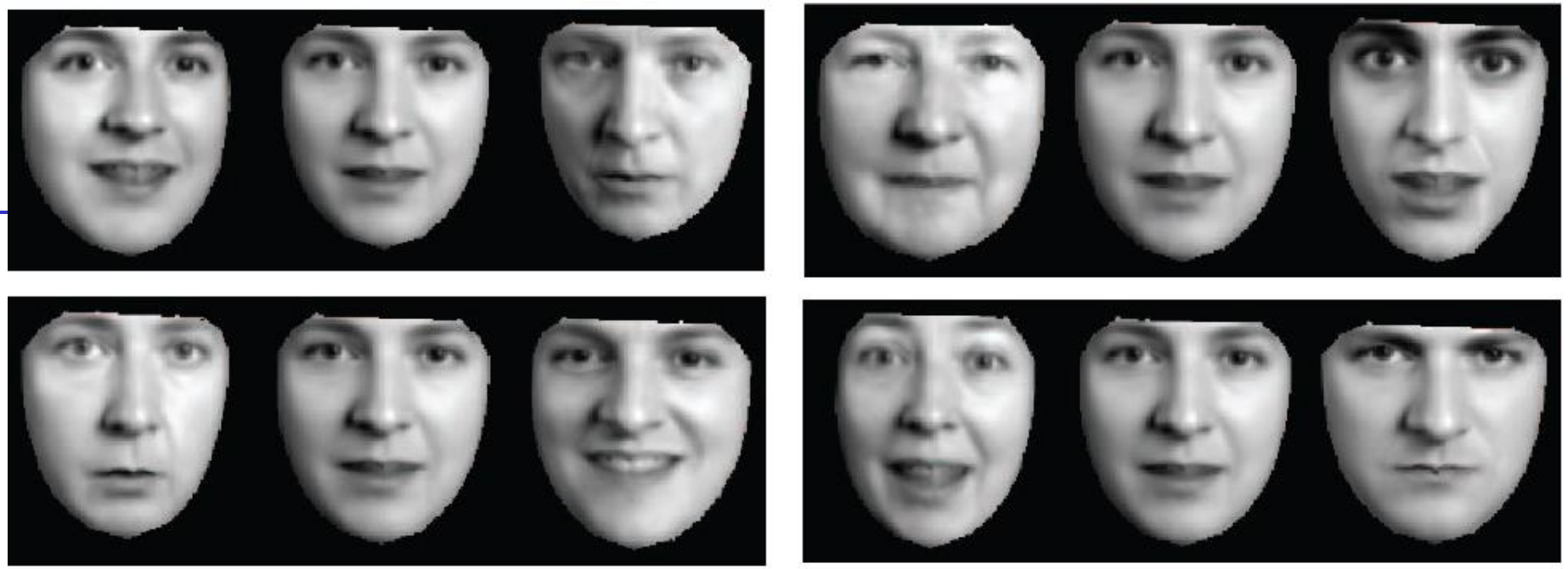
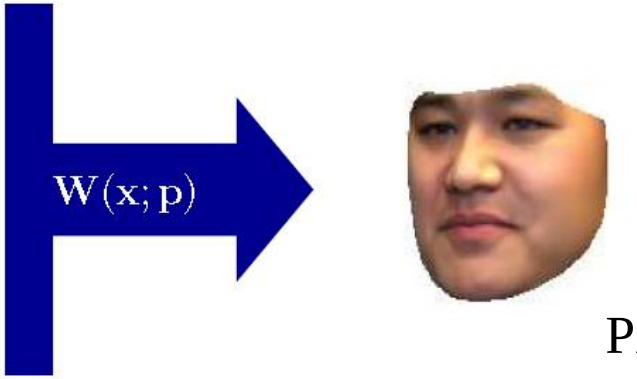


FIGURE 12.9: Different face intensity masks generated by moving deformation parameters to different values. Each block shows the effect of a different parameter; the center of that block shows the parameter at the mean value (where the mean is taken over numerous example faces), and the left (resp. right) of the block shows the parameter at mean plus (resp. minus) three standard deviations. Note how a range of expressions is encoded by these parameter variations. *This figure was originally published as Figure 2 of "Active Appearance Models," by T. Cootes, G. Edwards, and C. Taylor, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001,* © *IEEE, 2001.*

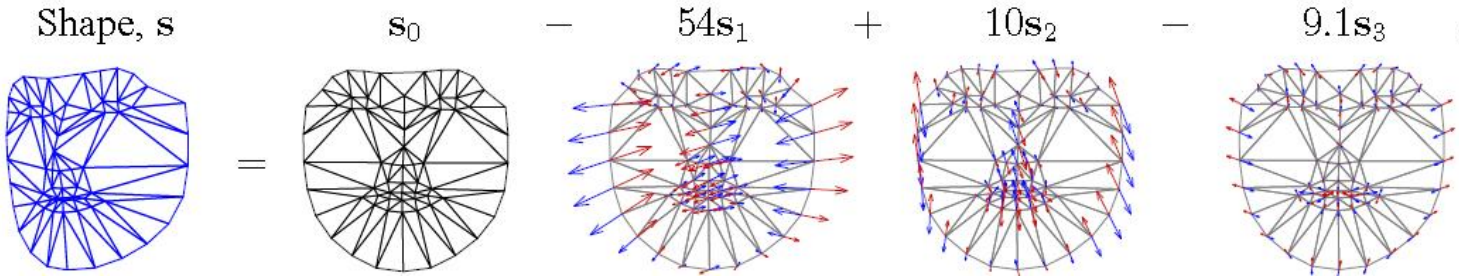
AAM Model Instantiation



AAM Model Instance

$M(\mathbf{W}(\mathbf{x};\mathbf{p}))$

Piecewise affine warping



Matthews and Baker, "Active Appearance Models Revisited", IJCV 2004

Registration – AAM Fitting

Goal: obtain optimal shape and appearance parameters to minimize the appearance difference between the model and the target image

$$\min_{\lambda,\mathbf{p}} \left\| \mathbf{A_0}(\mathbf{x}) + \sum_{i=1}^{K} \lambda_i \, \mathbf{A_i}(\mathbf{x}) - \mathrm{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) \right\|^2$$

Reconstructed appearance Warped image

- Difference is measured in the same coordinate system – the mean shape
- Assuming an initial parameter set is known, iteratively update the parameters

Registration – AAM Fitting

$$\left\| \mathbf{A_0}(\mathbf{x}) + \sum_{i=1}^{K} \lambda_i \, \mathbf{A_i}(\mathbf{x}) - \mathrm{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) \right\|^2$$

$$= \left\| \mathbf{A_0}(\mathbf{x}) + \sum_{i=1}^{K} \lambda_i \, \mathbf{A_i}(\mathbf{x}) - \mathrm{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) \right\|^2_{span(\mathbf{A_i})^{\perp}}$$

$$+ \left\| \mathbf{A_0}(\mathbf{x}) + \sum_{i=1}^{K} \lambda_i \, \mathbf{A_i}(\mathbf{x}) - \mathrm{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) \right\|^2_{span(\mathbf{A_i})}$$

Registration – AAM Fitting

$$\left\| \mathbf{A_0}(\mathbf{x}) + \sum_{i=1}^{K} \lambda_i \, \mathbf{A_i}(\mathbf{x}) - \mathrm{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) \right\|^2$$

$$= \| \mathbf{A_0}(\mathbf{x}) - \mathrm{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) \|^2_{span(\mathbf{A_i})^\perp} + \left\| \mathbf{A_0}(\mathbf{x}) + \sum_{i=1}^{K} \lambda_i \, \mathbf{A_i}(\mathbf{x}) - \mathrm{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) \right\|^2_{span(\mathbf{A_i})}$$

Only depends on $\mathbf{p}$

depends on $\boldsymbol{\lambda}$

- Find optimal $\mathbf{p}$ by minimizing the first term
- Fixed $\mathbf{p}$, solve $\boldsymbol{\lambda}$ by minimizing the second term

Registration – AAM Fitting

Lucas-Kanade (Forward –additive) algorithm:

$$\epsilon = \|\mathbf{A_0(x)} - \mathbf{I(W(x; p + \Delta p))}\|^2$$

$$\mathbf{p} \Leftarrow \mathbf{p} + \mathbf{\Delta p}$$

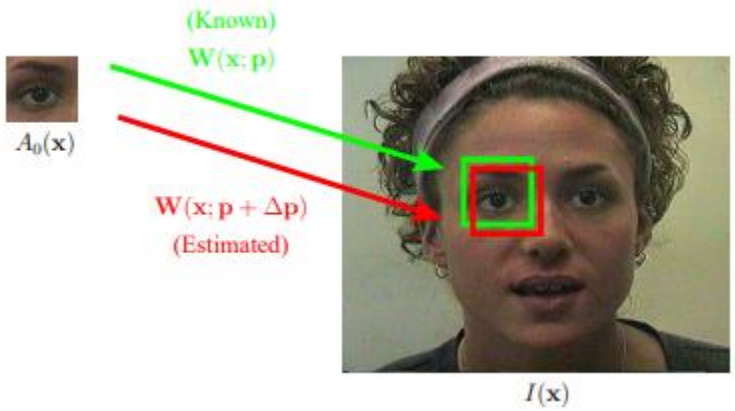


Figure 5: A schematic overview of the Lucas-Kanade (forwards-additive) image alignment algorithm. Given current estimates of the parameters $\mathbf{p}$, Lucas-Kanade linearizes the problem and solves for incremental updates to the parameters $\Delta\mathbf{p}$ that are then added to the current estimates $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$.

Registration – AAM Fitting

Forwards compositional image alignment

- Warp the image to the model

$$\epsilon = \|\mathbf{A_0}(\mathbf{x}) - \mathbf{I}(\mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p}))\|^2$$

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \Leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})$$

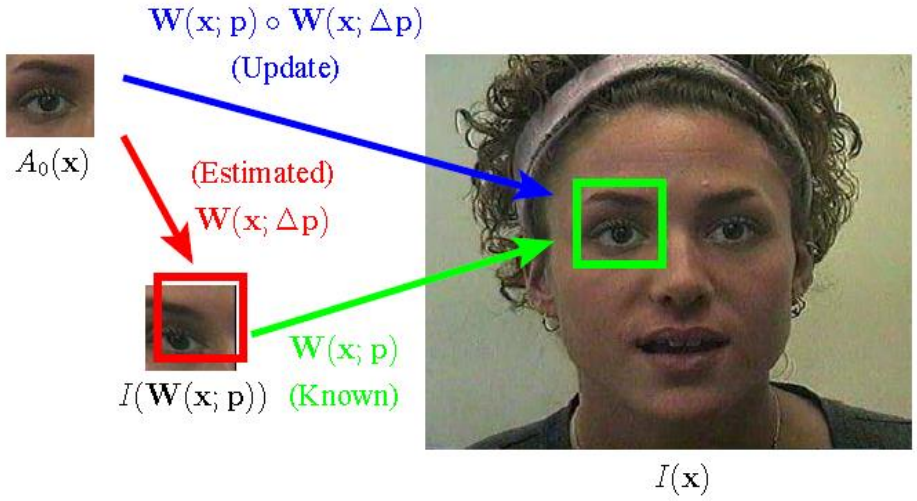
Inverse compositional image alignment

- Warp the model to the image

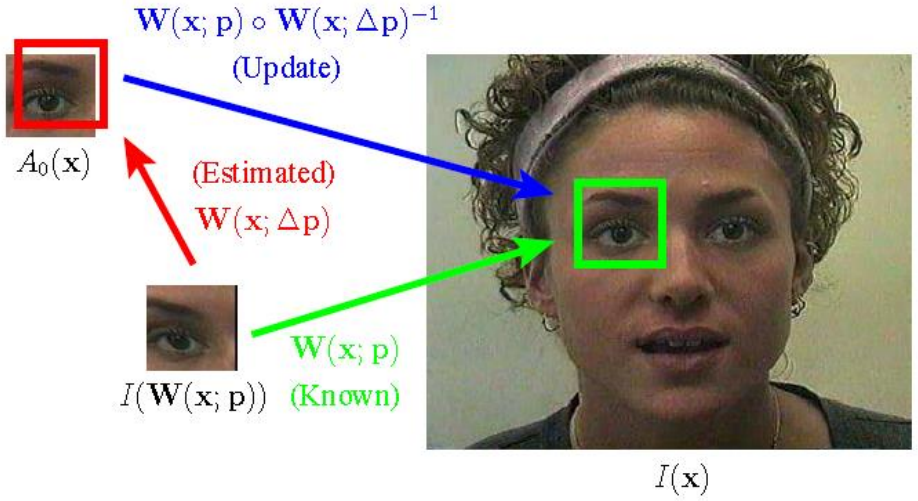
$$\epsilon = \left\|\mathbf{A_0}\left(\mathbf{W}(\mathbf{x};\Delta\mathbf{p})\right) - \mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}))\right\|^2$$

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \Leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$$

Registration – AAM Fitting



(a) Forwards Compositional



(b) Inverse Compositional

Figure 6: (a) A schematic overview of the forwards-compositional image alignment algorithm. Given current estimates of the parameters, the forwards compositional algorithm solves for an incremental warp $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ rather than a simple update to the parameters $\Delta\mathbf{p}$. The incremental warp is then composed with the current estimate of the warp. (b) A schematic overview of the inverse-compositional image alignment algorithm. The roles of $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ and $A_0(\mathbf{x})$ are reversed and the incremental warp $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ is estimated in the other (inverse) direction. The incremental warp therefore has to be inverted before it is composed with the current estimate of the warp.

Matthews and Baker, "Active Appearance Models Revisited", IJCV 2004

AAM Fitting

The Inverse Compositional Algorithm with Appearance Variation

Pre-compute:

(3) Evaluate the gradient ∇A_0 of the template $A_0(\mathbf{x})$
(4) Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; 0)$
(5) Compute the modified steepest descent images $\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
(6) Compute the Hessian matrix using modified steepest descent images

Iterate:

(1) Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$
(2) Compute the error image $I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - A_0(\mathbf{x})$
(7) Compute dot product of modified steepest descent images with error image
(8) Compute $\Delta \mathbf{p}$ by multiplying by inverse Hessian
(9) Update the warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta \mathbf{p})^{-1}$

Post-computation:

(10) Compute λ_i

Matthews and Baker, "Active Appearance Models Revisited", IJCV 2004

FIGURE 12.10: Active appearance models registered to face images. On the **left**, the initial configuration of the model (blurry blob over the face; original face is second from right). As the minimization process proceeds, the search improves the registration to produce, in the final converged state, the registration on the **right**. Once we have this registration, the location of the vertices of the mesh and the deformation parameters encode the shape of the face. *This figure was originally published as Figure 5 of "Active Appearance Models," by T. Cootes, G. Edwards, and C. Taylor, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001,* © *IEEE, 2001.*

AAM Challenges

- Initialization
- Insufficient training samples
- Large local deformation

Reading Assignments

[1] T. F. Cootes. Statistical models of appearance for computer vision. Online technical report available from http://www.face-rec.org/algorithms/AAM/app_models.pdf , Sept. 2001.

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, June 2001.

[3] I. Matthews and S. Baker, Active Appearance Models Revisited, IJCV 2004.

Image Resampling & Interpolation

Need to resample the image when

- Rescaling
- Geometrical transformation
- The output image coordinates are not discrete

Interpolation methods:

- Nearest neighbor
  - Fast and simple
  - Loss of sharpness
  - Artifacts (checkerboard)
- Bilinear
- Bicubic
  - Images are sharpest
  - Fine details are preserved
  - Slow

Image Resampling & Interpolation

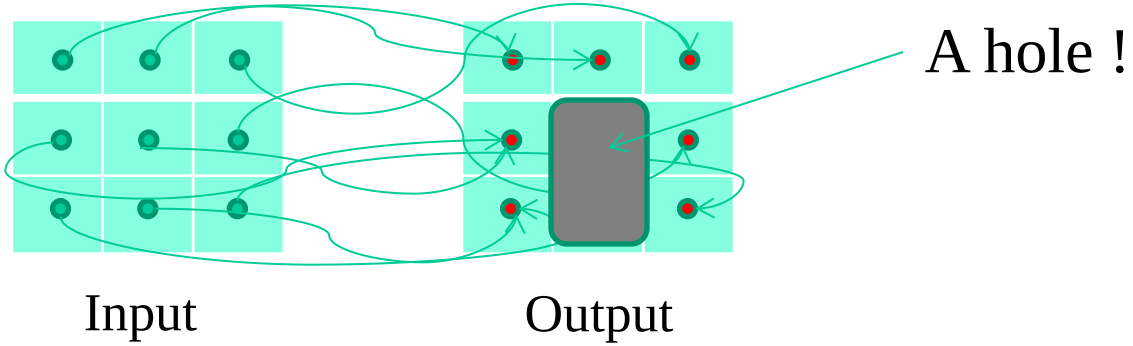
Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods

a b c
d e f

FIGURE 2.24 (a) Image reduced to 72 dpi and zoomed back to its original size (3692 × 2812 pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

Issues on Image Resampling & Interpolation

Missing points in forward mapping



Solution: perform a backward mapping

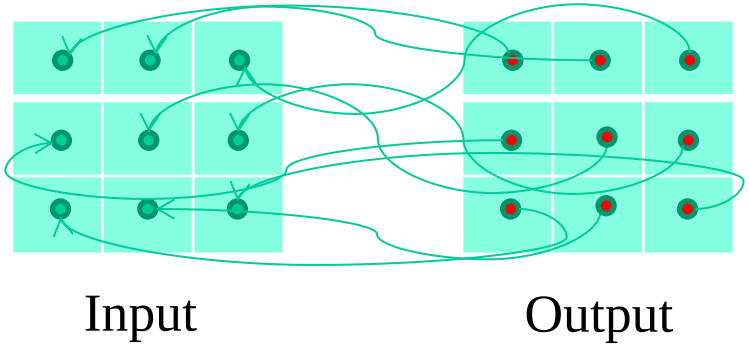
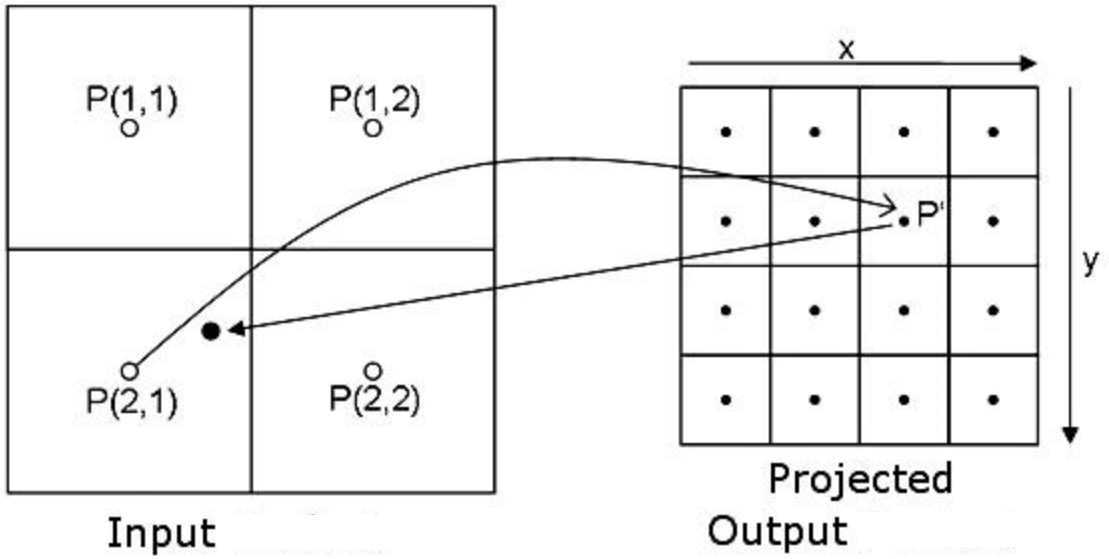


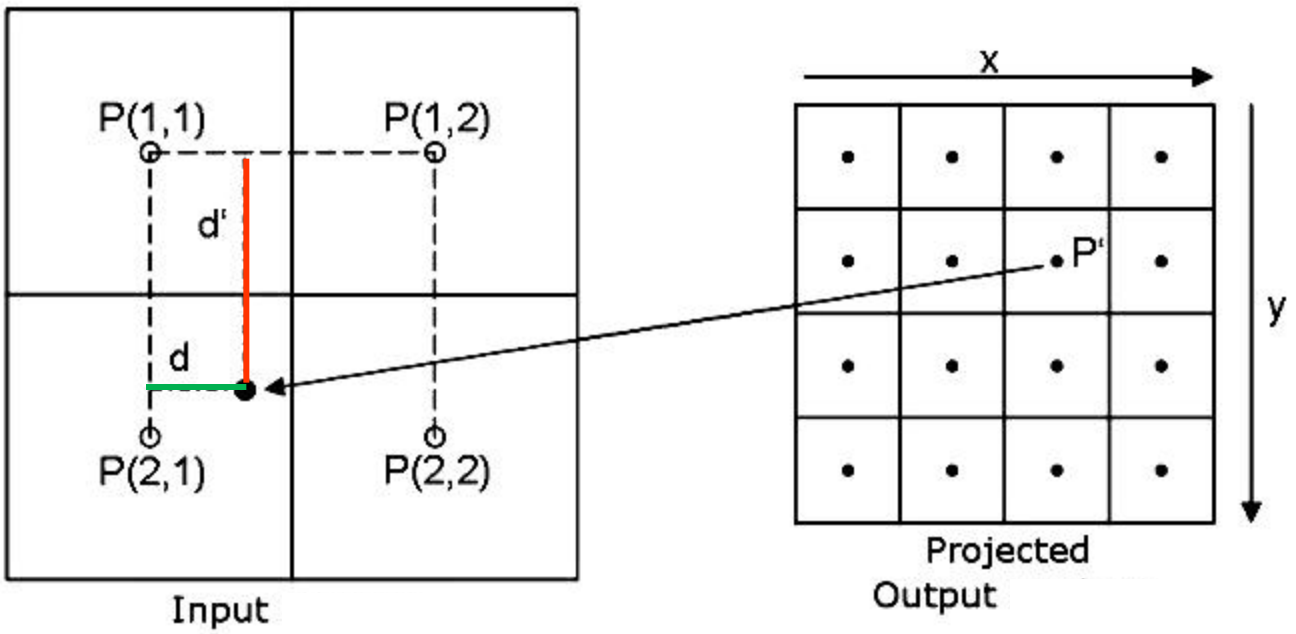
Image Interpolation – Nearest Neighbor



http://www.brockmann-consult.de/beam/doc/help/general/ResamplingMethods.html

Assign each pixel in the output image with the nearest neighbor in the input image.

Image Interpolation – Bilinear

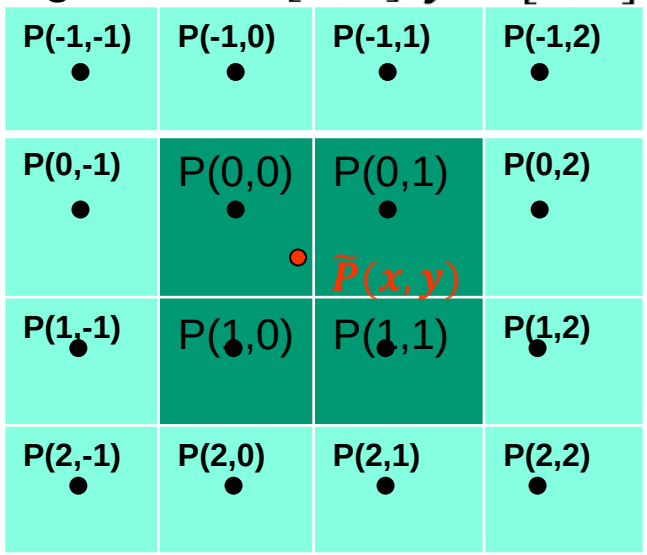


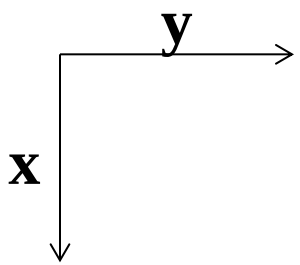
http://www.brockmann-consult.de/beam/doc/help/general/ResamplingMethods.html

$$P' = P(1,1)(1-d)(1-d') + P(1,2)d(1-d') + P(2,1) * d' * (1-d) + P(2,2)dd'$$

Image Interpolation – Bicubic

If we know the intensity values, derivatives, and cross derivatives for the four corners (0,0), (0,1), (1,0), and (1,1), we can interpolate any point (x,y) in the region $x \in [0,1], y \in [0,1]$





$$\tilde{P}(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} x^i y^j$$

Need to solve the 16 coefficients

Hierarchy of Computer Vision Problems

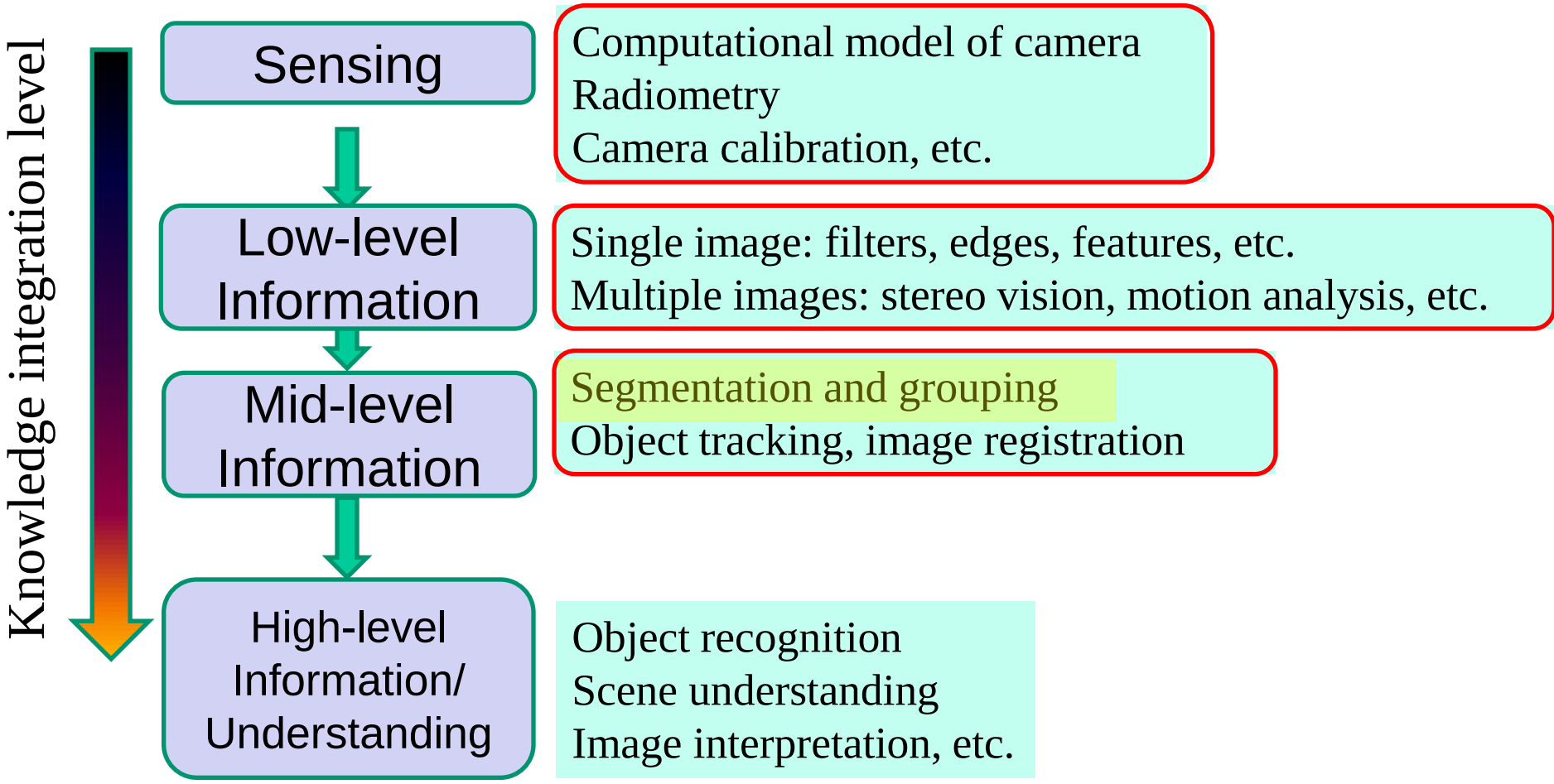


Image Segmentation

A process that partitions R into subregions $R_1, R_2, \ldots, R_n$

Microsoft multiclass segmentation data set

Image Segmentation – Applications

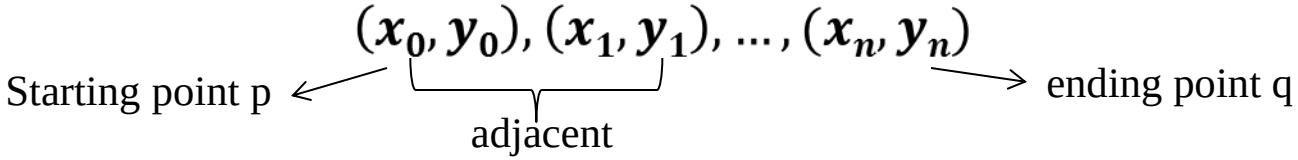
Object localization

Object recognition

Specifically important for medical imaging

Brief Review of Connectivity

- Path from p to q: a sequence of distinct pixels with coordinates



- p and q are *connected*: if there is a path from p to q in S
- *Connected component*: all the pixels in S connected to p
- *Connected set*: S has only one connected component
- R is a region if R is a connected set
- R_i and R_j are adjacent if $R_i \cup R_j$ is a connected set

Image Segmentation

(a) $\bigcup_{i=1}^{n} R_i = R$

(b) R_i is a connected set, $i = 1,\ldots,n$

(c) $R_i \cap R_j = \phi, \forall i \neq j$

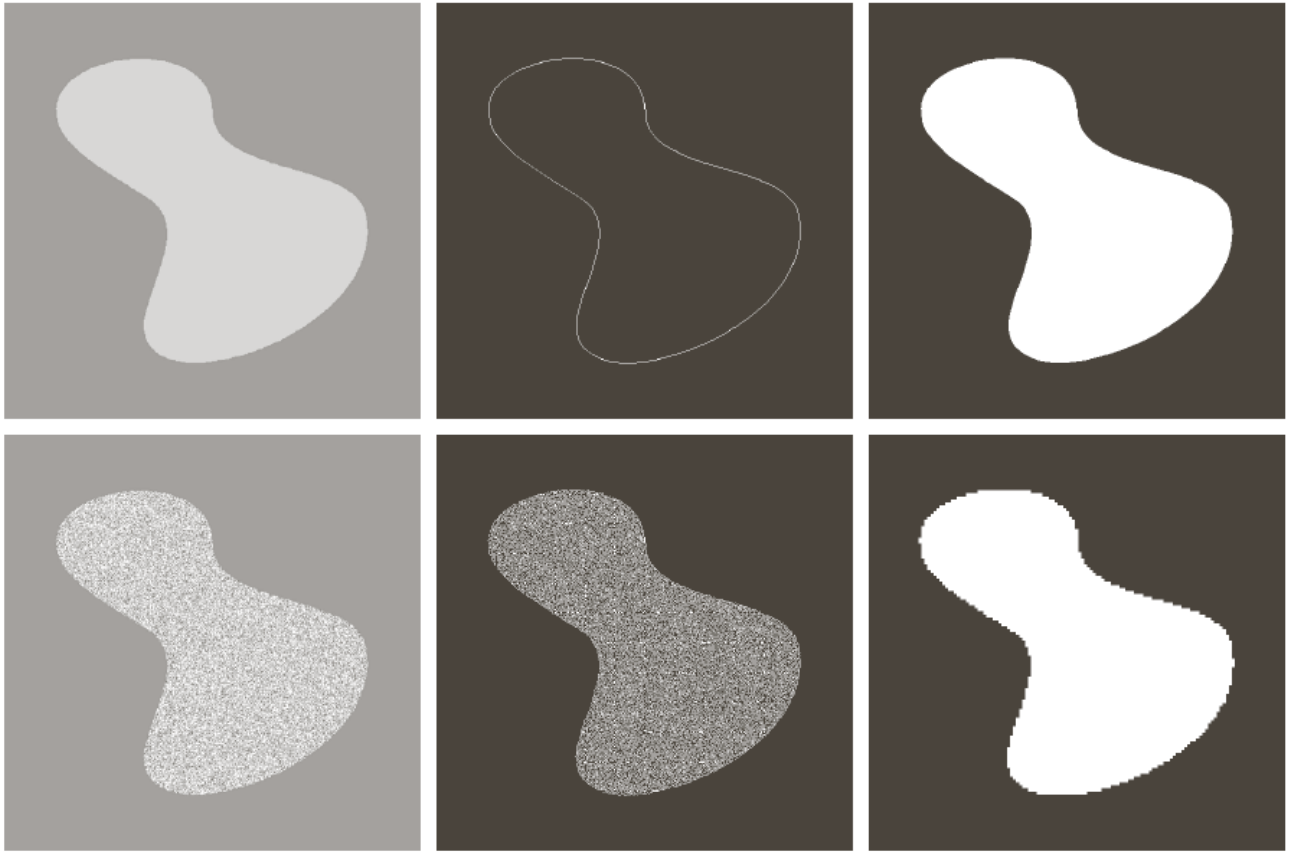
(d) $Q(R_i) = TRUE$

(e) $Q(R_i \cup R_j) = FALSE$ for adjacent regions R_i and R_j

Two categories based on intensity properties:

- Discontinuity – edge-based algorithms
- Similarity – region-based algorithms

Edge-based and Region-based Segmentation



a b c
d e f

FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Figure from "Digital Image Processing", 3rd edition, Gonzalez and Woods

Brief Review on Simple Edge Detectors

- **First-order derivative**
  - E.g., Roberts (2x2), Prewitt (3x3), Sobel (3x3, smooth + difference)
  - Thicker edge
  - One operator for one edge direction
- **Second-order derivative**
  - Laplacian (3x3)
  - Double edge
  - Zero-crossing
- **Common issues:**
  - Sensitive to image noise
  - Cannot deal with the scale change of the image

Advanced Edge Detection Techniques

- Deal with image noise
- Exploit the properties of image

→ Work much better for real images

Advanced edge detectors:

- Laplacian of Gaussian (LoG) $\nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$
- Difference of Gaussian (DoG)

$$DOG(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}, \quad \sigma_1 > \sigma_2$$

- Canny

Marr-Hildreth Detector (LoG)

Observations:

- Intensity changes are dependent on the image scale
- A sudden intensity change (step) causes a peak/trough in the1st order derivative and a zero-crossing in the 2nd order derivative
- The 2nd order derivative is especially sensitive to noise

➡ Smooth the image using a Gaussian filter first before applying the Laplacian

Gaussian $\longrightarrow G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$ **Laplacian of a Gaussian (LoG)**

$$g(x,y) = \nabla^2[G(x,y) \otimes f(x,y)] = \nabla^2 G(x,y) \otimes f(x,y)$$

- Varying σ values for scale changes
- Rotation invariant in edge detection

LoG Filtering

$$g(x,y) = \nabla^2 G(x,y) \otimes f(x,y)$$
$$= \nabla^2 [G(x,y) \otimes f(x,y)]$$

1. Filter the input image with an *nxn* Gaussian filter.
2. Compute the Laplacian of the intermediate image resulting from Step 1.
3. Find the zero-crossings of the image from Step 2.
   - opposite signs of the neighbors
   - the difference should be significant

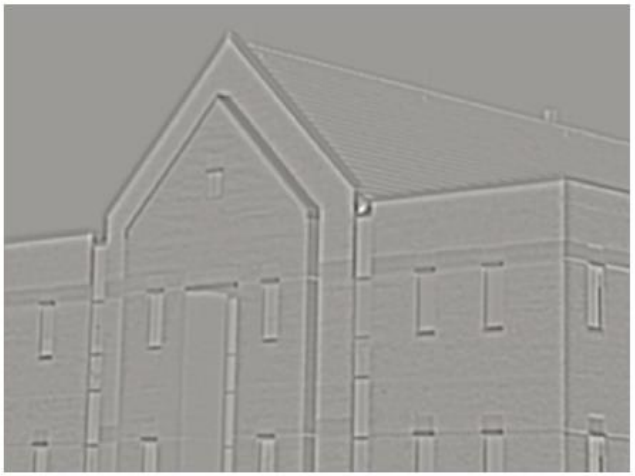
Note:

- Window size $n >= 6\sigma$ and n is an odd number

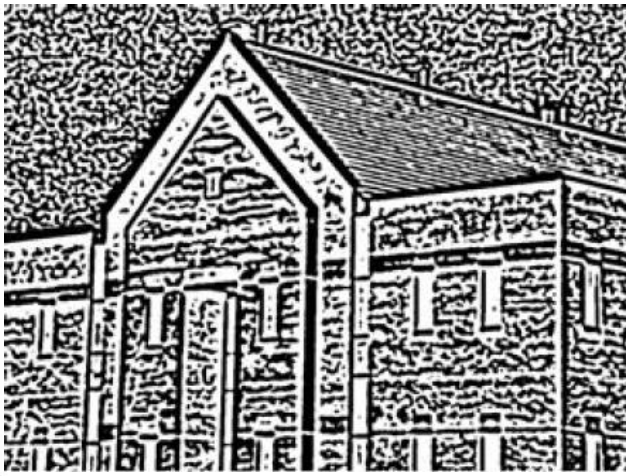
An Example – Edges are 1 Pixel Thick

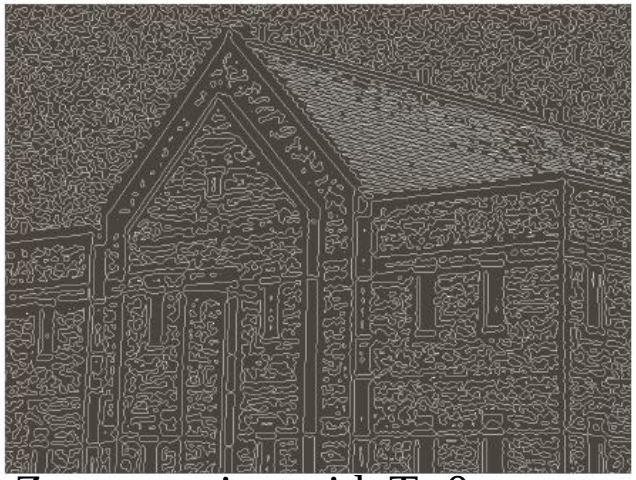
Original

LoG filtering



LoG filtering with T=0





Zero-crossing with T=0

Zero-crossing with T=4%max

Figure from "Digital Image Processing", 3rd edition, Gonzalez and Woods