Lecture 13
Shape

ch. 9, sec. 1-8, 12-14 of *Machine Vision* by Wesley E. Snyder & Hairong Qi

Spring 2017

16-725 (CMU RI) : BioE 2630 (Pitt)

Dr. John Galeotti

The content of these slides by John Galeotti, © 2012 - 2017 Carnegie Mellon University (CMU), was made possible in part by NIH NLM contract# HHSN276201000580P, and is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0/ or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA. Permissions beyond the scope of this license may be available either from CMU or by emailing itk@galeotti.net.

The most recent version of these slides may be accessed online via http://itk.galeotti.net/

Shape Analysis

- Image analysis requires quantification of image contents
  - We desire a relatively small number of highly meaningful image descriptors.
- But, segmentation gives us *lots* of data.
- We need a way to derive meaningful measures from a segmentation.

Shape Analysis

Segmentation
(Pixel labeling from object differentiation)

Image Understanding
(By means of shape analysis)

- How can I quantify the shape of this object?
- What, physically, is this segmented object?
- Does it look normal?

3

Shape Analysis & Linear Transformations

- We want to identify objects…
  - Based on numerical shape descriptors.
- But:
  - Changing the the zoom (size), position, or orientation of an object (or the “camera”) changes the contents of the resulting image.
- We often need…
  - Shape descriptors that evaluate to the same (vector or scalar) value for all sizes, positions, and/or orientations of any given shape

4

Shape Analysis & Linear Transformations

- Most shape descriptors are not invariant to all linear transforms.
- Many are not even invariant to similarity transformations
- Similarity transforms (i.e. *pose* transforms):
  - Translation and/or rotation only
  - Do not change the “shape” of an object

5

A digression into transformations

- Linear transforms can be implemented as a matrix that multiplies the vector coordinates of each pixel in an object
  - Example of rotating shape S about the z-axis (2D in-plane rotation):

$$S' = R_Z S = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 4 & 1 & 3 & 2 \\ 3 & 7 & 9 & 8 \end{bmatrix}$$

Coordinates of point 1 in shape S

Coordinates of point 3 in shape S

- Several types:
  - Rotation
  - Translation
  - Zoom
  - Affine
    - skew
    - different scaling in different directions
  - Perspective
    - lines stay straight, but not parallel

6

Homogeneous coordinates

- What:
  - A slick way to implement translation via matrix multiplication
- How:
  - Add the “dummy” coordinate of 1 to the end of every coordinate vector:

$$X' = \begin{bmatrix} \cos\theta & -\sin\theta & dx \\ \sin\theta & \cos\theta & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

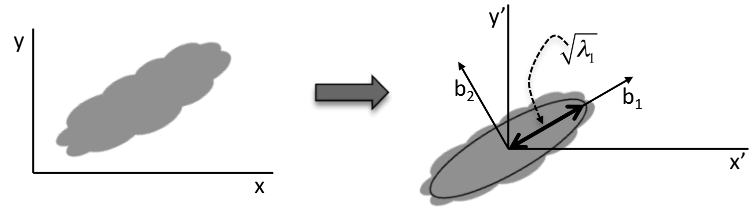
7

Transformations for Medical Imaging

- In medical imaging, we usually don’t have optical perspective.
  - So, we usually don’t want or need invariance to perspective transformations.
  - We often don’t even need affine transforms.
- In medical imaging, we know the size of each voxel.
  - So, in some cases, we don’t want or need invariance to scale/zoom either.

8

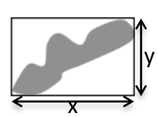
PCA (K-L Expansion)



- **Big Picture:** Fitting a hyper-ellipsoid & then (typically) reducing dimensionality by flattening the shortest axes
- Same as fitting an (N+1)-dimensional multivariate Gaussian, and then taking the level set corresponding to one standard deviation
- Mathematically, PCA reduces the dimensionality of data by mapping it to the first n eigenvectors (*principal components*) of the data's covariance matrix
- The first principal component is the eigenvector with the largest eigenvalue and corresponds to the longest axis of the ellipsoid
- The variance along an eigenvector is exactly the eigenvector's eigenvalue
- This is VERY important and VERY useful. Any questions?

Basic Shape Descriptors

- Trivial to compute—O(n) with a *small* coefficient:
  - Average, max, and min *intensity*
  - *Area* (A) and *perimeter** (P)
  - *Thinness / compactness / isoperimetric measure* (T), if based on P^2/A
  - *Center of mass (i.e. center of gravity)* → $\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} x_i \\ y_i \end{bmatrix}$
  - *X-Y Aspect Ratio* →

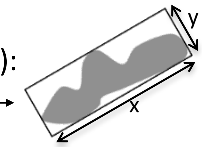


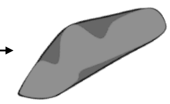
- Easy to compute:
  - *Number of holes*
  - *Triangle similarity* (ratio of side lengths to P)

* Perimeter has several definitions; some are trivial to compute

Basic Shape Descriptors

- Requires PCA first, which itself is O(D^3+$D^2 n$):
  - Approximate *minimum aspect ratio*
  - Approximate *diameter* (D)
  - *Thinness* / *compactness* / *isoperimetric measure* (T), if based on D/A
- O(n log n):
  - Convex discrepancy
- Difficult to compute:
  - Exact *diameter* = absolute max chord
  - Exact *minimum aspect ratio*
  - *Symmetry*, mirror or rotational





* Perimeter has several definitions; some are difficult to compute

11

Shape Analysis in (Simple)ITK

- SimpleITK's LabelShapeStatisticsImageFilter:
  - http://www.itk.org/SimpleITKDoxygen/html/classitk_1_1simple_1_1LabelShapeStatisticsImageFilter.html
- Underlying ITK Filter & Data Classes:
  - http://www.itk.org/Doxygen/html/classitk_1_1LabelImageToShapeLabelMapFilter.html
  - http://www.itk.org/Doxygen/html/classitk_1_1ShapeLabelObject.html
- C++ ITK Example:
  - http://www.itk.org/Doxygen/html/WikiExamples_2ImageProcessing_2ShapeAttributes_8cxx-example.html

12

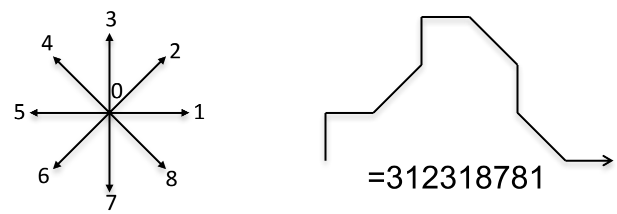
Method of Normalization

- Idea: Transform each shape's image region into a canonical frame before attempting to identify shapes
- Simple, but common, example:
  - Move origin to the center of gravity (CG) of the current shape
  - Used by central moments (next slide)
- Complex example:
  - Attempt to compute and apply an affine transform to each object such that all right-angle-triangle objects appear *identical*

Moments

- Easy to calculate
- Sequence of derivation:
  - Moments: $m_{pq} = \sum x^p y^q f(x,y)$
  - Central moments: μ_{pq} (origin @ CG)
  - Normalized central moments: η_{pq}
    - Invariant to translation & scale
  - **Invariant moments: φ_n**
    - **Invariant to translation, rotation, & scale**
    - Only 7 of them in 2D
    - Equations are in the text
- Problem: Sensitive to quantization & sampling

Chain codes

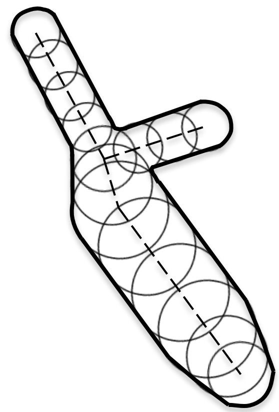


- Describe the boundary as a sequence of steps
  - Typically in 2D each step direction is coded with a number
- Conventionally, traverse the boundary in the counter-clockwise direction
- Useful for many things, including syntactic pattern recognition

Fourier Descriptors

- Traverse the boundary
  - Like for chain codes
- But, take the FT of the sequence of boundary-point coordinates
  - In 2D, use regular FT with i = *y-axis*
- Equivalences make invariance “easy”:
  - Translation = DC term
  - Scale = multiplication by a constant
  - Rotation about origin = phase shift
- Problem: Quantization error

Medial Axis



- I may revisit this in another lecture (if time allows)
- For now:
  - Locus of the centers of the maximal bi-tangent circles/spheres/…

17

Deformable Templates

- Represent a shape by the active contour that segments it
  - Deforming the contour deforms the shape
- Two shapes are considered similar if the boundary of one can be "easily" deformed into the boundary of the other.
  - E.g., "easy" = small strain on the deformed curve and low energy required to deform the curve

18

Generalized Cylinders (GCs)

- Fit a GC to a shape
  - This can be challenging
- Get two descriptive functions:
  - Axis of the GC
    - A vector-valued function
  - Radius along the axis
    - Typically a scalar-valued function