Announcements

- No lectures for next week. Enjoy the break!
- The following Tuesday on 11/28, we dedicate the lecture time for your group project work. Submit the update of your group status on the Canvas project plan page for each team prior to this meeting. Be ready to give a short presentation for update.
- Is your team project wrapping up good? Check any team members struggling? Make sure you adjust your team’s goals/milestones.
- Project Team Presentation on 12/5 in three weeks
  - Submit your team’s power-point slides to me by 12/4 5pm
  - Q&A after the talks on Canvas, please take notes!
  - After the presentation, peer evaluation as part of grading!

Overview

- Last lecture
  - Medical Image Registration II
  - Intensity-based registration
  - Non-rigid registration
  - Thin Plate Splines/Cubic B-Splines & Demons

- Today’s lecture
  - After Segmentation
    - Feature Extraction
    - Classification/Recognition
  - After Registration
    - Change Analysis
  - Course Overview
Image Quantification

CSC621-821
Biomedical Imaging and Analysis
Dr. Kazunori Okada

Goal of What We Study Today

• Segmentation and Registration are pre-process to do …
• What?
• Segmentation
  – Object classification / Recognition
  – Diagnosis/Screening
• Registration
  – Comparison / Change Analysis
  – Follow-up study
• We learn today basics of what comes next…
After Segmentation

- Segmentation is the first step for many post processes: pre-processing
- Yields different image representation
  - Boundary
  - Medial Axis
  - Region

1. Feature extraction
2. Classification/Recognition

Feature Extraction

- Features express intrinsic characteristics of an object & extracted from a result of segmentation
  - Often designed to be independent to scaling, rotation, translation
  - Used for later detection, classification, recognition tasks
- Shape (Geometry)
  - Boundary
  - Medial Axis
  - Binarized Image
- Topology (Non-geometric shape info)
- Texture (Intensity distribution)
  - Intensity Statistics
Invariant Descriptors/Features

- We want features that do not depend on **location**, **orientation** and **scale**

![Feature space](image)

Boundary Detection

- Object boundary can be detected by
  - Any contour-based segmentation (Snake, Level set)
  - Use morphological operations from the region-based segmentation results (midterm 2)
  - Use watersheds on gradient image
  - Edge linking / Border tracking (Canny)

- Boundary Features (see Ch.11 of DIP book)
  - Signature
  - Chain code
  - Polygonal Approximation
  - Boundary segments
  - Geometric Measures
Signature

- 2D boundary to 1D function
- Invariant to location, but will depend on rotation and scaling.

**Figure 11.5**
Distance versus angle signatures.
In (a) \( r(\theta) \) is constant. In (b), the signature consists of repetitions of the pattern
\[ r(\theta) = A \sec \theta \text{ for } 0 \leq \theta \leq \pi/4 \text{ and } r(\theta) = A \csc \theta \text{ for } \pi/4 < \theta \leq \pi/2. \]

Boundary Segments

- Boundary segments: decompose a boundary into segments.
- Use of the convex hull of the region enclosed by the boundary is a powerful tool for robust decomposition of the boundary.
Geometric Measures of Boundary

• Boundary Length
  – the number of pixels/voxels along a boundary gives a rough approximation of its length.

• Curvature
  – the rate of change of slope
  – to measure a curvature accurately at a point in a digital boundary is difficult
  – the difference between the slopes of adjacent boundary segments is used as a descriptor of curvature at the point of intersection of segments

Medial Axis / Skelton

• Produce a one pixel wide graph that has the same basic shape of the region, like a stick figure of a human. It can be used to analyze the geometric structure of a region which has bumps and “arms”.

![Figure 11.7](image)
Thinning Algorithm

• Iteratively thin a binary image into a Skelton
  – A contour point is any pixel with value 1 and having at least one 8-neighbor valued 0
  – Let

\[ 0 \leq N(p_1) \leq 5 \]

\[ T(p_1) : \text{the number of 0-1 transitions} \]

in the ordered sequence

\[ p_2, p_3, ..., p_8, p_9, p_2 \]

<table>
<thead>
<tr>
<th></th>
<th>( p_0 )</th>
<th>( p_2 )</th>
<th>( p_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_8 )</td>
<td>( p_1 )</td>
<td>( p_4 )</td>
<td></td>
</tr>
<tr>
<td>( p_7 )</td>
<td>( p_6 )</td>
<td>( p_5 )</td>
<td></td>
</tr>
</tbody>
</table>

Thinning Algorithm: Step 1

• Flag a contour point \( p_1 \) for deletion if the following conditions are satisfied

(a) \( 2 \leq N(p_1) \leq 6 \)
(b) \( T(p_1) = 1 \)
(c) \( p_2 \cdot p_4 \cdot p_6 = 0 \)
(d) \( p_4 \cdot p_6 \cdot p_8 = 0 \)

<table>
<thead>
<tr>
<th></th>
<th>( p_0 )</th>
<th>( p_2 )</th>
<th>( p_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_8 )</td>
<td>( p_1 )</td>
<td>( p_4 )</td>
<td></td>
</tr>
<tr>
<td>( p_7 )</td>
<td>( p_6 )</td>
<td>( p_5 )</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 11.9**
Illustration of conditions (a) and (b) in Eq. (11.1-1). In this case \( N(p_1) = 4 \) and \( T(p_1) = 3 \).
Thinning Algorithm: Step2

• Flag a contour point \( p_1 \) for deletion again. However, conditions (a) and (b) remain the same, but conditions (c) and (d) are changed to

\[
\begin{align*}
\text{(d')} & \quad p_2 \cdot p_6 \cdot p_8 = 0 \\
\text{(c')} & \quad p_2 \cdot p_4 \cdot p_8 = 0
\end{align*}
\]

\[
\begin{array}{ccc}
 p_0 & p_2 & p_3 \\
 p_8 & p_1 & p_4 \\
 p_7 & p_6 & p_5 \\
\end{array}
\]

Overall Procedure for TA

• A thinning algorithm:
  – (1) applying step 1 to flag border points for deletion
  – (2) deleting the flagged points
  – (3) applying step 2 to flag the remaining border points for deletion
  – (4) deleting the flagged points
  – This procedure is applied iteratively until no further points are deleted.
Skelton Example

- Structure
- Handwriting

Region-based Features

- Features extracted from a region
  - Binary region
    - First compute binary connected-component by thresholding followed by connected components labeling
    - Then extract feature from the binary region.
  - Greyscale regions
    - Assume a segmentation process provides a form of binary mask or a condition to tell which pixel/voxel belongs to a region
    - Extract features from a set of intensity values in the region
Binary Image Characteristics

- Extract some useful features from a binary connected component
  - Area/Volume
  - Compactness
  - Centroid
  - Circularity
  - Bounding box
  - Extremal points
  - Spatial moments
  - Moment Invariants

Area/Volume

- We denote the set of pixels in a binary connected component region by $R$ (intensity value is 1)
- Assuming square pixels/voxels, area $A$ is given by counting the number of pixels in $R$
  \[ A = \sum_{(x,y) \in R} 1 \]
- Volume is given by counting in 3D
  \[ V = \sum_{(x,y,z) \in R} 1 \]
- When voxel is not squared,
  \[ V = \sum_{(x,y,z) \in R} (x_{\text{size}} \times y_{\text{size}} \times z_{\text{size}}) \]
Compactness

- The perimeter $P$ of a region $R$
  - the length of its boundary $B$
  - Cardinality of a set of all boundary pixels/voxels
- The compactness of a region:
  \[ c = \frac{(\sum_{(x,y) \in B} 1)^2}{\sum_{(x,y) \in R} 1} \left( \frac{\text{# boundary pixels}}{\text{area}} \right)^2 \]
  \[ c = \frac{4 \pi^2 r^2}{1^2} = 4 \pi \]
  \[ (r, r, \ldots, c, \ldots) \]

Centroid

- Center of mass: given by spatial average
  \[ \bar{x} = \frac{1}{A} \sum_{(x,y) \in R} x \]
  \[ \bar{y} = \frac{1}{A} \sum_{(x,y) \in R} y \]
- In 3D, replace $A$ by $V$ and also compute an average of $z$ coordinates
  \[ \bar{z} = \frac{1}{V} \sum_{(x,y,z) \in R} z \]
Circularity

- Measure the deviation from a perfect circle
  
  Circularity: \( C = \frac{\mu_R}{\sigma_R} \)

  where \( \mu_R \) and \( \sigma_R^2 \) are the mean and variance of the distance from the centroid of the shape to the boundary pixels \((x_k, y_k)\).

  - Mean radial distance:
    \[
    \mu_R = \frac{1}{K} \sum_{k=0}^{K-1} \| (x_k, y_k) - (\bar{x}, \bar{y}) \|
    \]

  - Variance of radial distance:
    \[
    \sigma_R^2 = \frac{1}{K} \sum_{k=0}^{K-1} \left( \| (x_k, y_k) - (\bar{x}, \bar{y}) \| - \mu_R \right)^2
    \]

Bounding Box

- Given a binary connected component \( R \),

- A bounding box is defined by a rectangle (2D) or polygon (3D) with its corners defined by:
  
  \[
  \begin{align*}
  x_{\text{min}} & \geq \min_x \text{ and } x_{\text{max}} \leq \max_x \\
  y_{\text{min}} & \geq \min_y \text{ and } y_{\text{max}} \leq \max_y
  \end{align*}
  \]
Extremal Points: Break

- Thirion introduced extremal points, which are points of locally maximal curvature in both principal directions of the surface. These points are specific points on the crest lines and can be seen as the generalization of corner points for smooth surfaces in 3D.

Moments (Shape Statistics)

- Binary connected component is denoted by $b(x,y)$, then $k$-th moment for $x$ is defined by
  $$M(k) = \iint x^k b(x,y) dx dy$$
- $k=0^{th}$ Moment = Area $A$  
  $$A = \iint b(x,y) dx dy = \sum_{x} \sum_{y} b(x,y)$$
- $k=1^{st}$ Moment = Center of Mass 
  $$\bar{x} = \frac{1}{A} \iint x b(x,y) dx dy$$  \quad $$\bar{y} = \frac{1}{A} \iint y b(x,y) dx dy$$
- $k=2^{nd}$ Moment = Variance 
  $$\sigma_x^2 = \iint x^2 b(x,y) dx dy$$  \quad $$\sigma_y^2 = \iint y^2 b(x,y) dx dy$$
Central Moments

- \( R \) is a subset of image pixels (region).
- Central \((j,k)\)th moment defined as:
  \[
  \mu_{jk} = \sum_x \sum_y (x - \bar{x})^j (y - \bar{y})^k b(x,y) = \sum_{(x,y) \in R} (x - \bar{x})^j (y - \bar{y})^k
  \]
- Invariant to translation of \( R \).
- Interpretation:
  - 0th central moment: area
  - 1st central moment: mean
  - 2nd central moment: variance
  - 3rd central moment: skewness
  - 4th central moment: kurtosis

Moment Invariants

- Normalized central moments
  \[
  \eta_{pq} = \frac{\mu_{pq}}{\mu_{00}}, \quad \gamma = \frac{p + q}{2} + 1 \quad \text{if } q \geq 0
  \]
- From those, a set of invariant moments can be defined for object description.
  \[
  \phi_1 = \eta_{20} + \eta_{02}, \quad \phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \quad \phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \quad \phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2
  \]
  (Additional invariant moments \( \phi_5, \phi_6, \phi_7 \) can be found in the literature).
- Robust to translation, rotation & scaling, but don’t expect wonders (still summary statistics).
### More Invariants

\[
\begin{align*}
\phi_3 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{align*}
\]

### Orientation

- **Difficult to define!**
  - Axis of least second moment
  - For mass: Axis of minimum inertia

\[
\begin{align*}
E(\rho, \theta) &= \int \int r^2 b(x, y) \, dx \, dy
\end{align*}
\]
Axis of Least Second Central Moment

• Invariance to orientation?
  – Need a common alignment
  – Compute Eigenvectors of 2nd moment matrix (Matlab: eig(A))

\[
\begin{bmatrix}
\mu_{20} & \mu_{11} \\
\mu_{11} & \mu_{02}
\end{bmatrix} = V D V^T =
\begin{bmatrix}
v_{11} & v_{12} & \lambda_1 \\
v_{21} & v_{22} & \lambda_2
\end{bmatrix}
\begin{bmatrix}
v_{11} & v_{12} \\
v_{21} & v_{22}
\end{bmatrix}
\]

Summary of Binary Image Characteristics

• Pros
  – Fast to compute, easy to store
  – Simple processing techniques
  – Can be very useful for constrained scenarios

• Cons
  – Hard to get “clean” silhouettes
  – Noise is common in realistic scenarios
  – Can be too coarse a representation
  – Cannot deal with 3D changes
Topological Features

Topological property 1:
the number of holes \((H)\)

![Figure 11.17](image) A region with two holes.

Topological property 2:
the number of connected components \((C)\)

![Figure 11.18](image) A region with three connected components.

Topological Features: Euler Number

Topological property 3:
Euler number: the number of connected components minus the number of holes
\[ E = C - H \]

![Figure 11.19](image) Regions with Euler number equal to 0 and -1, respectively.
Largest Connected Component

Topological property 4: the largest connected component.

What is Texture?

- What is texture?
Texture Types

- Defined as the smoothness or roughness of a surface.
- Visual appearance of the uniformity or lack of uniformity of brightness and color.
- There are two types of texture: random and regular.
  - Random texture cannot be exactly described by words or equations; it must be described statistically. The surface of a pile of dirt or rocks of many sizes would be random.
  - Regular texture can be described by words or equations or repeating pattern primitives. Clothes are frequently printed with regularly repeating patterns.
  - Random texture is analyzed by statistical methods.
  - Regular texture is analyzed by structural or spectral (Fourier) methods.

Statistical Approach (Intensity Statistics)

- Let \( z \) denote gray levels and let \( p(z_i), i=0,1,...,L-1 \), be the corresponding normalized histogram, where \( L \) is the number of distinct gray levels
  - The \( n \)-th central moment of \( z \): (intensity mean, variance, skewness...)
    \[
    \mu_n(z) = \sum_{k=0}^{L-1} (z_i - m)^n p(z_i) \quad \text{where} \quad m = \sum_{i=0}^{L-1} z_i p(z_i)
    \]
  - The regularity: \( R = 1 - \frac{1}{1 + \sigma^2(z)} \)
  - The uniformity: \( U = \sum_{i=0}^{L-1} p^2(z_i) \)
  - The Shannon entropy: \( e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i) \)
Example of Intensity Statistics

![Images of Smooth, Coarse, and Regular Intensity Statistics]

<table>
<thead>
<tr>
<th>Texture</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>( R ) (normalized)</th>
<th>Third moment</th>
<th>Uniformity</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth</td>
<td>82.64</td>
<td>11.79</td>
<td>0.002</td>
<td>-0.105</td>
<td>0.028</td>
<td>5.434</td>
</tr>
<tr>
<td>Coarse</td>
<td>143.56</td>
<td>74.60</td>
<td>0.017</td>
<td>-0.151</td>
<td>0.078</td>
<td>7.802</td>
</tr>
<tr>
<td>Regular</td>
<td>99.72</td>
<td>33.73</td>
<td>0.017</td>
<td>0.759</td>
<td>0.013</td>
<td>6.674</td>
</tr>
</tbody>
</table>

SFSU: CSC621-821 – Kazunori Okada (C) 2022

39

Structural Approach

- **Structural concepts:**
  - Suppose that we have a rule of the form \( S \rightarrow aS \), which indicates that the symbol \( S \) may be rewritten as \( aS \).
  - If \( a \) represents a circle [Fig. 11.23(a)] and the meaning of “circle to the right” is assigned to a string of the form \( aaaa... \) [Fig. 11.23(b)].

![Diagram of Template Matching]

FIGURE 11.23
(a) Texture primitive.
(b) Pattern generated by the rule \( S \rightarrow aS \).
(c) 2-D texture pattern generated by this and other rules.

Template Matching

SFSU: CSC621-821 – Kazunori Okada (C) 2022

40
Spectral Approach

- Extract frequency information about texture patterns via Fourier transform of an image
- $S(r, \theta)$ is called spectrum function which is the result of Fourier transform of an image. It is described in polar coordinate, $r$ is radius/frequency and $\theta$ is angle
  - For each direction $\theta$, $S(r, \theta)$ may be considered a 1-D function $S_\theta(r)$.
  - For each frequency $r$, $S_r(\theta)$ is a 1-D function.
  - A global description:
    \[
    S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \quad S(\theta) = \sum_{r=1}^{R_0} S_r(\theta)
    \]
    Summed over all angles \quad Summed over all radius

Example of Spectral Signatures

![Example of Spectral Signatures](image_url)

**FIGURE 11.24** (a) Image showing periodic texture. (b) Spectrum. (c) Plot of $S(r)$. (d) Plot of $S(\theta)$. (e) Another image with a different type of periodic texture. (f) Plot of $S(\theta)$. (Courtesy of Dr. Dragana Brzakovic, University of Tennessee.)
After Feature Extraction

• You can do …

• Classification/Recognition
• Detection (this is a little bit different but another application of the classification)

Classification

• A task to differentiate input types
Pattern Classes & Object Recognition

- **Pattern classes**
  - A pattern class is a family of patterns that share some common properties
  - Yellow Fish and Green Fish
  - Malignant and Benign

- **Pattern recognition**
  - to assign a given pattern to its respective class

- **Training/Learning**:
  - to find a function that assigns a class to arbitrary input pattern from a set of training examples:
    - \{(pattern1, class1), (pattern2, class2), ...\}

Type of Classifiers

- **Supervised**: know the answer
  - Nearest-neighbor classifier
  - k-nearest-neighbor classifier
    - same class as majority of k-closest training set
  - Bayes classifier
  - Neural Net, Convolutional NN
  - Support Vector Machine, Random Forest etc.

- **Unsupervised**: don’t know the answer
  - K-mean clustering
  - Parzen window
  - Autoencoder
Feature Space

- First agree on a set of features that you use for your classifier (fea1, fea2, fea3,..)

Feature Space

```
\begin{align*}
\text{width/curvature} \\
\text{lightness/circularity}
\end{align*}
```

Decision Function & Boundary

- Class assigning function!

```
C = \{ w(x) = a + bx \}
```

Tune decision boundary model
Complexity
Overfitting
"Less performance in training data better performance in novel patterns"
How do we find decision boundary from data?
In a simple 1D example
Make a histogram of a feature (length/volume) for each class (e.g., salmon or sea bass/malignant or benign)
And find the best threshold function!

Feature Extraction vs Classifier
Small number of features → simpler decision regions, easier to train and quicker response
Ideal Feature Extraction → Trivial Classifier
Perfect Classifier → Simple Features
General purpose recognition system is a very difficult challenge (DNN does good job)
Practice defines what is the best approach
For example: how many features do you need? What architecture of DNN? Study
Classifiers Overview: Break

• Pros
  – Being non-iterative, classifiers are relatively computationally efficient during runtime
  – Can be applied to any types of images

• Cons
  – Do not perform any spatial modeling
  – Require manual interaction to obtain training data
  – Preparing training data is tedious and time-consuming
  – Performing training is iterative and often time-consuming

Detection

• It is a task to locate a region of interest
  – Find a fish in an image of aquarium
  – Find a tumor in a CT image

• In medical imaging, this is one of the most important task
  – Computer-aided detection of X
  – Automating a semi-automatic segmentation (automatic seeding)
Template matching

• Simplest way for detection
  1. have a template (small image showing the region of interest)
  2. raster scan the input image,
  3. at each pixel location, compute a similarity value of the template and corresponding image region
  4. After finish scan, find a pixel location that gave the highest similarity

What’s wrong with it?

• How to compute similarity?
  – Many similarity functions
  – You can extract invariant features first then compare

• How to select template?
  – Object appearance may change (tumor)

• How many templates should I use?
How to improve?

• Notice that template matching is essentially a binary nearest neighbor classifier
• So you can use more advanced classifier to detect your target
  – Random Forest, SVM, Adaboost
  – Deep Neural Net
• Machine learning…
  – YOLO
• Cons: Training of your detectors can take looooong time…

After Registration

• Registration is the first step for comparing images
• After alignment you can
  – Detect any abnormal changes (anomaly detection)
  – Analyze the extent of changes (change quantification)
  – Provide a basis for a follow-up study to look for recurrence of cancers
  – Provide a basis for therapy monitoring to see if lesions are improving
### Change Detection / Anomaly Detection

- Given an image \( A(\mathbf{x}) \) and \( B(\mathbf{x}) \) or \( (A(\mathbf{x},t) \) and \( A(\mathbf{x},t+dt)) \),
- find a set of locations \( \mathbf{y} \in \mathbf{x} \) that indicates a change between \( A \) and \( B \)
  - Difference Image
  - Ratio Image
  - Local Comparison with model (training)
  - General classification

#### Difference Image

<table>
<thead>
<tr>
<th></th>
<th>8</th>
<th>10</th>
<th>8</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>240</td>
<td>11</td>
<td>10</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>210</td>
<td>205</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>220</td>
<td>98</td>
<td>88</td>
<td>46</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>143</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>110</td>
<td>0</td>
<td>-168</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>0</td>
<td>-166</td>
<td>-164</td>
<td></td>
</tr>
</tbody>
</table>

Image 1

\[
\text{Difference Image} = \text{Image 1} - \text{Image 2}
\]

Image 2

Threshold the absolute values
## Difference Image Overview

**Pros**
- Simple (some say it's the most commonly used method)
- Easy to interpret
- Robust

**Cons:**
- Difference value is absolute, so same value may have different meaning depending on the starting class

## Ratio Image

**Instead of subtraction, use the ratio**

\[ \frac{I_{m1}}{I_{m2}} \]

**If no change, you have 1**

**Threshold the image for detection**

**Pros**
- Simple
- May mitigate problems with viewing conditions, esp. sun angle

**Cons**
- Asymmetric as to the direction of change: 5/2 and 2/5
Local Comparison

• Local comparison
  – After alignment of A & B,
  – Raster-scan with \( nxn \) local window pair from A & B.
  – Extract the same features from both images
  – Compute dissimilarity at each location
  – Threshold the dissimilarity values to find change locations

• Cons
  – You don’t know if the change come from alignment errors or real changes…

Local Comparison with Model

• Training phase
  – Align N number of images into the same coordinate
  – At each pixel location, compute the standard deviation of pixel value differences for \( N(N-1)/2 \) pairs

• Then align A & B to the same N images

• Then at each pixel/voxel location
  – Compute the pixel value difference
  – If the difference is within the standard deviation of the location, no change
  – Otherwise it is a change!
General Classifier

- The local comparison can be generalized by using a general classifier
  1. Train a region classifiers
  2. Perform the classifier to subregions of A & B
  3. Then compare the resulting class labels between A & B.

- Pros
  - Avoids need for strict calibration
  - Designates type of change occurring

- Cons
  - Error is multiplicative from two parent maps

Course Overview

- You learned about
  - How medical images are generated
  - How to handle images
  - How to process images
  - How to represent images

- Things to think
  - Extending 2D algorithms to 3D
  - Could be a research work!

- Online tutorial and Java demos available on
How medical image are generated

- Lec 2-3,
- Principle of Optical Image Generation
- Biological imaging
- X-ray/CT
- MRI
- PET/SPECT
- US
- Fusion
- Different Physics exploited for different modalities

How to handle images

- Lec 1 & 4
- Image Data Structures
- Image As a Function / Lattice
- Sampling/Resolution/Quantization
- Graylevels & Colors
- Image Standard & Viewers
  - DICOM, ANALYZE
- Image Visualization
  - MPR, MIP, Interpolation
How to process images

• Lec 5-8
  • Point process vs Neighboring process
  • Enhancement: histogram equalization
  • Smoothing: average, Gaussian, anisotropic diff.
  • Sharpening/Edge detection: Laplace, LoG, Sobel
  • Mathematical Morphology
  • Connected Components Labeling
  • Fourier analysis / aliasing

How to process images

• Lec 9-12
  • Segmentation
    – Thresholding: Otsu method
    – Region growing
    – Watersheds
    – Classification
    – Level Sets
  • Registration
    – Rigid Landmark-based: Singular Value Decomposition (markers)
    – Rigid Surface-based: Iterative Closest Point algorithm
    – Rigid Intensity-based: Mutual Information Maximization
    – Nonrigid Landmark-based: Thin-Plate Splines, Cubic B-splines
    – Nonrigid Intensity-based: Demons algorithm

Computer Vision!
How to represent images

• Lec 13
• Feature extraction after segmentation
  – Boundary
  – Skelton
  – Region
  – Statistical moments
• Change analysis after registration
  – Difference image
  – Classifier

To your future

• Visualization
• GPU-parallelization
• Clinical studies with engineering minds
• Machine learning (learn from data)
• Computer vision (geometry from data)
• Data mining (pattern detection)
• Artificial Intelligence (logical reasoning)
• Deep learning
Next Class if you are interested.

- CSC872 Pattern Analysis Machine Intelligence
- AI & PR & Machine Learning in depth
- Learn about the recognition & learning
- Old syllabus
- [https://bidal.sfsu.edu/~kazokada/csc872/](https://bidal.sfsu.edu/~kazokada/csc872/)

Summary

- **Image Quantification**
  - After Segmentation
    - Feature Extraction
    - Classification/Recognition
  - After Registration
    - Change Analysis
- **Course Overview**

- **Next Week:**
  - Work on project work.
  - Group project work on 11/28 in class.
  - Project presentations in three weeks!
  - Practice! 30 min max. (overshooting times will be negatively graded).