Announcements

• Midterm 2 in two next week (4/10):
  100 min. Bring a calculator: All hand derivation/calculations
  – 6 questions on Lecture 6 (1), 7(1), 8 (2), 9 (2).
  –
  –
  –
  – Closed book, Closed lecture note photo-copies
  – Cheat Sheets: Bring “hand-written” notes of any length (make sure write down all formulas we covered in the lectures).
  – For prep: study all algorithms covered so that you can manually solve it with a small input image. Solve by hand completely all examples used in lecture and their obvious variations. You can bring your handwritten notes, remember! Pay attention to details and recall reasons why you do these steps.

• Senior Oral Presentation
  – Your project presentation participation. Make sure you participate!
  – Send your request in iLearn forum “Senior Oral Presentation”.
  – Bring a hard copy of filled-in form to the class. The form can be obtained at the CS department office.

• Project: 3D visualization can use ready-made tools like ImageJ. Share your findings at iLearn for extra credits! Think also of quantification part. Come show your current progress during my office hours for feedbacks
Medical Image Segmentation I: Thresholding, Region Growing Watershed, Classification

CSC621-821
Biomedical Imaging and Analysis
Dr. Kazunori Okada

Overview

• Last lecture
  – Advanced Image Processing
  – Edge-Preserving Smoothing (combine smoothing and sharpening)
  – Morphological Operations (post segmentation cleaning)
  – Connected-Components Labeling (counting # of regions)
  – Segmentation: Overview

• Today’s lecture
  – Medical Image Segmentation I
  – Thresholding
  – Region Growing
  – Watersheds
  – Classification
Two Basic Properties

- Segmentation algorithms for gray scale images are generally based on one of two basic properties of gray-scale values:
  - **Discontinuity**
    - The approach is to partition an image based on abrupt changes in gray-scale levels.
    - The principal areas of interest within this category are detection of isolated points, lines, and edges in an image.
  - **Similarity**
    - The principal approaches in this category are based on thresholding, region growing, and region splitting/merging.

This Lecture

- **Similarity-based methods**
- **Thresholding** (only intensity)
- **Region Growing** (intensity & space)
- **Watershed** (intensity & space)
- **Classification** (combine intensity and space in a feature space)
What is Thresholding?

- Most basic segmentation method
- Grayscale image ⇒ Binary mask
- Thresholding can be a first step in other advanced segmentation approaches
- Single value thresholding can be given mathematically as follows:

\[
g(x, y) = \begin{cases} 
1 & \text{if } f(x, y) > T \\
0 & \text{if } f(x, y) \leq T 
\end{cases}
\]

Thresholding is Useful

- Imagine a poker playing robot that needs to visually interpret the cards in its hand

Original Image  
Thresholded Image
Or not

• If you get the threshold wrong the results can be disastrous

Threshold Too Low  |  Threshold Too High

How to pick the right threshold value???

Histogram-based Thresholding

• 3 Steps:
  – **Make Histogram**: Use intensity histogram of an image to guide the estimation of threshold value!
  – **Choose Threshold Value** $T$: Partitioning the histogram such that different gray value distributions are separated (partition a bimodal (two-peak) histogram)
  – **Do Segmentation by Thresholding with** $T$: Applying the threshold function to entire image

• The success of this technique very strongly depends on specific histogram shapes!!!
Bimodal Intensity Histograms

Ideal histogram, light object on dark background

Actual observed histogram with noise

Bimodal Histogram = only two things in images

More Complicated Cases

- How to separate those?

  Two distinct modes
  Overlapping modes
  Multiple modes

- Threshold selection is difficult in the general case
  - Domain knowledge often helps
  - E.g. Is foreground object light or dark?
  - E.g. Fraction of text on a document page (⇒ histogram quantile)
  - E.g. Size of objects/structure elements
Basic Algorithm

• The basic global threshold, \( T \), can be calculated as follows:

1. Select an initial estimate \( T_0 \) for \( T \) (typically the average grey level in the image).
2. Segment the image using \( T_0 \) to produce two groups of pixels: \( G_1 \) consisting of pixels with grey levels \( > T_0 \) and \( G_2 \) consisting pixels with grey levels \( \leq T_0 \).
3. Compute the average grey levels of pixels in \( G_1 \) to give \( \mu_1 \) and \( G_2 \) to give \( \mu_2 \).

\[
\mu = \frac{1}{N} \sum_{x \in G} g(x)
\]

4. Compute a new threshold value:

\[
T = \frac{\mu_1 + \mu_2}{2}
\]

5. Repeat steps 2 – 4 until the difference in \( T \) in successive iterations is less than a predefined limit \( T_\infty \).

• This algorithm works very well for finding thresholds when the histogram is suitable.
Thresholding Example 1

Thresholding Example 2
Optimal Threshold Theory

- The value that separates two peaks as best as it can
- Search for the threshold $T$ that minimizes the within-class variance $\sigma_{\text{within}}$ of the two classes separated by $T$

$$\sigma_{\text{within}}^2(T) = n_1(T)\sigma_1^2(T) + n_2(T)\sigma_2^2(T)$$

where

$$n_1(T) = \left\{ I_{(x,y)} \leq T \right\}, \quad n_2(T) = \left\{ I_{(x,y)} > T \right\}$$

- This is the same as maximizing the between-class variance $\sigma_{\text{between}}$

$$\sigma_{\text{between}}^2(T) = \sigma^2 - \sigma_{\text{within}}^2(T)$$

$$= n_1(T)n_2(T)[\mu_1(T) - \mu_2(T)]^2$$

---

Otsu '75 Algorithm

1. Pre-compute a gray value histogram $h$.
2. For each potential threshold $T \leftarrow 1, \ldots, 256$
   1. Separate the pixels into two clusters $G_1$ and $G_2$ according to $T$
   2. Look up $n_1, n_2$ in $h$ and compute both cluster means
   3. Compute $\sigma_{\text{between}}^2$ and update
3. Choose $T^* = \arg \max_T [\sigma_{\text{between}}^2(T)]$
Example 3

- Let’s say we want to isolate the contents of the bottles
- Think about what the histogram for this image would look like
- What would happen if we pick a certain threshold value??
Problem with Single Value Algorithm

- Single value thresholding only works for bimodal histograms (two peaks)
- Images with other kinds of histograms need more than a single threshold

Variant Thresholding Methods: Break

- Single-valued, One-sided
  \[ F_T[i, j] = \begin{cases} 
    1, & \text{if } F[i, j] \geq T \\
    0, & \text{otherwise} 
  \end{cases} \]

- Two-valued, Two-sided
  \[ F_T[i, j] = \begin{cases} 
    1, & \text{if } T_1 \leq F[i, j] \leq T_2 \\
    0, & \text{otherwise} 
  \end{cases} \]

- Set membership
  \[ F_T[i, j] = \begin{cases} 
    1, & \text{if } F[i, j] \in Z \\
    0, & \text{otherwise} 
  \end{cases} \]
Un-even Illumination Example

- Uneven illumination can really fail a single valued thresholding scheme

Adaptive Thresholding

- An approach to handling situations in which single value thresholding will not work is to divide an image into sub images and threshold these individually
- Since the threshold for each pixel depends on its location within an image this technique is said to be **adaptive**
Adaptive Thresholding Example

- The image below shows an example of using adaptive thresholding with the image shown previously.

- As can be seen success is mixed.
- But, we can further subdivide the troublesome sub images for more success.

Adaptive Thresholding Example Cond.

- These images show the troublesome parts of the previous problem further subdivided.
- After this subdivision successful thresholding can be achieved.
Adaptive Thresholding Algorithms

- Estimate a local threshold within a small neighborhood window $W$ [Niblack'86]
  \[ T_W = \mu_W + k \cdot \sigma_W \]
  where $k \in [-1, 0]$ is a user-defined parameter.

- Improved version to suppress background noise for document binarization [Sauvola'00]
  \[ T_W = \mu_W \left[ 1 + k \cdot \left( \frac{\sigma_W}{R} - 1 \right) \right] \]
  where $R$ is the dynamic range of $\sigma$ and $k > 0$.
  - Typical values: $R=128$ for 8-bit images and $k \approx 0.5$.

Results

Original image

Global threshold selection (Otsu)

Local threshold selection (Niblack)
Inhomogeneity Correction

- Some images contain a smooth gradient (MRI, documents)
  \[ \implies \text{Try to fit that gradient with a polynomial function} \]

![Original image](image1.png) ![Fitted surface](image2.png)

Surface Fitting Algorithm

- Polynomial surface of degree \( d \), \((x,y)\) pixel coordinates

\[
\text{Least-squares estimation, e.g. for } d=3 (m=10) \text{ with image with } n \text{ pixels}
\]

\[
\begin{bmatrix}
1 & x_0 & y_0 & x_0^2 & y_0^2 & \cdots & x_0^3 & y_0^3 \\
1 & x_1 & y_1 & x_1^2 & y_1^2 & \cdots & x_1^3 & y_1^3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
1 & x_n & y_n & x_n^2 & y_n^2 & \cdots & x_n^3 & y_n^3
\end{bmatrix}
\begin{bmatrix}
b_0 \\
b_1 \\
\vdots \\
b_m
\end{bmatrix} =
\begin{bmatrix}
I_0 \\
I_1 \\
\vdots \\
I_n
\end{bmatrix} = \mathbf{A} \mathbf{b}
\]

\[
\text{Solution with pseudo-inverse:}
\]

\[
b = (A^T A)^{-1} A^T \mathbf{I}
\]

\[
\text{Matlab (using SVD):}
\]

\[
b = \mathbf{I} \backslash \mathbf{A}
\]
Surface Fitting Algorithm Cond.

- Iterative Algorithm
  1.) Fit parametric surface to all points in region.
  2.) Subtract estimated surface.
  3.) Apply global threshold (e.g. with Otsu method)
  4.) Fit surface to all background pixels in original region.
  5.) Subtract estimated surface.
  6.) Iterate further if needed…

- The first pass also takes foreground pixels into account.
  - This is corrected in the following passes.
  - Basic assumption here: most pixels belong to the background.

Various Thresholding Results
Various Thresholding Results

Original image

Global (Otsu)

Local (Sauvola)

Polynomial + Global

Issue: Spatial Proximity

- Thresholding can segment pixels, which are far apart but accidentally have similar intensity, as parts of the same object. Why?

- Thresholding only utilized intensity information ignoring the chance that pixels/voxels located nearby tend to have similar values!

- We should insert spatial proximity into consideration…

   In order to make sure segmented objects are connected!
Region-based Segmentation

Definition

• Image segmentation is the partitioning of an image into non-overlapping, contiguous regions that are uniform/homogeneous with respect to some characteristics.

Uniformity Predicate

• The uniformity of a connected region of pixels can be defined by a uniformity predicate. A logical statement that is true only if pixels in the region are sufficiently similar in terms of gray level, color or some other property.

\[
P(R) = \begin{cases} 
  \text{TRUE} & \text{if } |f(j,k) - f(m,n)| \leq \Delta \\
  \text{FALSE} & \text{otherwise}
\end{cases}
\]

For all neighboring (j,k) and (m,n) in R.

where \((j,k)\) and \((m,n)\) are the coordinates of neighboring pixels in region R.

\[
P(R) = \begin{cases} 
  \text{TRUE} & \text{if } |f(j,k) - \mu_R| \leq \Delta \\
  \text{FALSE} & \text{otherwise}
\end{cases}
\]

For all \((j,k)\) in R.

where \(f(j,k)\) is the grey level of a pixel from region R with coordinates \((j,k)\) and \(\mu_R\) is a mean grey level of all pixels in R except the pixel at \((j,k)\).
Region-based Segmentation

- **Formal Definition**: Let $R$ be a set of all pixels from the entire image region, and let $P(R_i)$ denote a uniformity predicate over sub-region $R_i$ that is a subset of $R$. Then, segmentation is a process that partitions $R$ into $n$ sub-regions, $R_1, R_2, \ldots, R_n$, such that

1. $\bigcup_{i=1}^{n} R_i = R$  
   The regions cover the whole image (complete)

2. $R_i$ is a connected region, $i = 1, 2, \ldots, n$.  
   Each region must be a connected component

3. $R_i \cap R_j = \emptyset$ for all $i$ and $j$, $i \neq j$  
   The regions have no overlaps (disjoint)

4. $P(R_i) = \text{true}$ for $i = 1, 2, \ldots, n$.  
   Each region must be uniform as per $P$

5. $P(R_i \cup R_j) = \text{false}$ for $i \neq j$  
   A union of any regions must be non-uniform

Region Growing

- **Concept**
  - Simplest region-based segmentation
  - Group pixels into larger regions in **bottom-up fashion**
  - **Pixel aggregation**: starting with a set of “seed” points, grow regions by appending to each seed points their neighboring pixels that have similar properties (such as gray level, texture, color, shape).
  - Region growing based techniques are better than the edge-based techniques in noisy images where edges are difficult to detect.
Region Growing Algorithm

1. Given a set of seed pixels and an image
2. For each seed pixel, continue to...
3. Add a pixel to a region if and only if:
   - It has not been assigned to any other region.
   - It is a neighbor of that region.
   - The new region created by addition of the pixel is still uniform.

Region Growing Process

1. Initialize $p$ by a seed point
2. Add $n$-neighbors of $p$ to list $L$ if $P(p, R) = $True
3. Pop the top item of $L$ and replace $p$ by it
4. Add $p$ to a region $R$ if $p$ is not evaluated yet and if $P(p, R) = $True else mark $p$ as boundary
5. Go to 2 until $L$ is empty
   - Two Regions $R$ and $\neg R$
Region Growing Process Details

• $P$ is a predicate that defines whether an element belongs to a region, i.e.:
  – Compares the new element with the neighbor value.
  – Compares the new element with the mean value of region.

• $L$ is ordered, for example, according to the value computed for evaluating $P$.

• We can define several seed points:
  – If a point touches more than one $R_i$, a measure defines to which region it belongs to.

Exercise 1

Suppose that we have the image given below.

(a) Use the region growing idea to segment the object. The seed for the object is the center of the image. Region is grown in horizontal and vertical directions, and when the difference between two pixel values is less than or equal to 5.

Table 1: Show the result of Part (a) on this figure.

<table>
<thead>
<tr>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>69</td>
<td>79</td>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td>59</td>
<td>10</td>
<td>60</td>
<td>64</td>
<td>59</td>
<td>56</td>
<td>60</td>
</tr>
<tr>
<td>10</td>
<td>58</td>
<td>10</td>
<td>60</td>
<td>70</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>10</td>
<td>60</td>
<td>59</td>
<td>65</td>
<td>67</td>
<td>10</td>
<td>65</td>
</tr>
</tbody>
</table>

Use 4-connected neighbor
Use uniformity predicate with neighboring pixels
Exercise 2

(b) What will be the segmentation if region is grown in horizontal, vertical, and diagonal directions?

Table 2: Show the result of Part (b) on this figure.

<table>
<thead>
<tr>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>69</td>
<td>70</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>59</td>
<td>10</td>
<td>60</td>
<td>64</td>
<td>59</td>
<td>56</td>
<td>60</td>
</tr>
<tr>
<td>10</td>
<td>69</td>
<td>10</td>
<td>60</td>
<td>70</td>
<td>10</td>
<td>62</td>
</tr>
<tr>
<td>10</td>
<td>60</td>
<td>59</td>
<td>85</td>
<td>67</td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Use 8-connected neighbor
Use uniformity predicate with neighboring pixels

Limitation of Region Growing 1

- Ambiguities around edges of adjacent regions may not be resolved correctly.
- Different choices of seeds may give different segmentation results.
- Results are completely dependent on the choice of the predicate $P$
- In another word, you must choose right predicate/define what you mean by uniform!
- And give sensible seed points
Limitation of Region Growing 2

- Region growing satisfies the 2\textsuperscript{nd} and 4\textsuperscript{th} of the region-based segmentation criteria (connected and uniform), but not the others.

Review: Region-based Segmentation

1. \( \bigcup_{i=1}^{n} R_i = R \)
   - The regions cover the whole image (complete)

2. \( R_i \) is a connected region, \( i = 1, 2, \ldots, n \)
   - Each region must be a connected component

3. \( R_i \cap R_j = \emptyset \) for all \( i \) and \( j, i \neq j \)
   - The regions have no overlaps (disjoint)

4. \( P(R_i) = \text{true} \) for \( i = 1, 2, \ldots, n \)
   - Each region must be uniform as per ‘P’

5. \( P(R_i \cup R_j) = \text{false} \) for \( i \neq j \)
   - A union of any regions must be non-uniform
Limitation of Region Growing 2

- Region growing satisfies the 2\textsuperscript{nd} and 4\textsuperscript{th} of the region-based segmentation criteria (**connected and uniform**), but **not the others**.
- It fails to satisfy the 1\textsuperscript{st} criterion (**cover all pixels**) because the seeds defined by the user may not be sufficient to create a region for every pixel.
- The 3\textsuperscript{rd} and 5\textsuperscript{th} criteria (**neighboring regions have no overlaps and are not uniform**) is not satisfied because the regions grown from two nearby seeds are always regarded as distinct, even if those seeds are defined in a part of the image that should be segmented as a single region.

Split and Merge Algorithm

- A complete segmentation (covering all five criteria) is possible if we adopt a **top-down approach**
- **Region Splitting**
  - The entire image is considered initially to be a single region
  - If P(image) is FALSE (not uniform), we **divide** the image into four quadrants
  - If P is FALSE for any quadrant, we further **subdivide** that quadrant into sub quadrants and so on
Region Merging: Break

• Region Merging
  - Region merging is the opposite of region splitting.
  - Start with small regions (e.g. 2x2 or 4x4 regions) and
  - Merge the regions that have similar characteristics (such as gray level, variance).
  - Typically, splitting and merging approaches are used iteratively.

1. Set the entire image as an initial region $R$
2. Split into four disjoint regions $R_i$ for which $P(R_i) = \text{FALSE}$
3. Merge any adjacent regions $R_j$ and $R_k$ for which $P(R_j \cup R_k) = \text{TRUE}$
4. Stop when no further merging or splitting is possible.

Boundary Based Segmentation

• Region Growing focused on defining regions
• When multiple regions are put nearby, boundary of these regions can be confusing to this algorithm…
• Can we focus more on boundaries???
Watershed Segmentation Idea

- Visualize an image in 3D: spatial coordinates and gray levels.
- In such a topographic interpretation, there are 3 types of points:
  - Points belonging to a regional minimum
  - Points at which a drop of water would fall to a single minimum. (\textit{catchment basin} or watershed of that minimum.)
  - Points at which a drop of water would be equally likely to fall to more than one minimum. (\textit{divide lines} or \textit{watershed lines}.)

Watershed Segmentation Algorithm

- The objective is to find the watershed lines.
- The idea is simple:
  - Suppose that a hole is punched in each regional minimum and then the entire topography is flooded from below by letting water rise through the holes at a uniform rate.
  - When rising water in distinct catchment basins is about to merge, a dam is built to prevent merging. These \textit{dam boundaries} correspond to the watershed lines.
Watershed Segmentation Algorithm

We find the location of local peaks in intensity values!!!

Watershed Segmentation Example

FIGURE 10.44
(Continued)
(c) Result of further flooding.
(f) Beginning of merging of water from two catchment basins (a short dam was built between them).
(g) Longer dams, (h) Final watershed segmentation lines. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)
Watershed Process

- Start with all pixels with the local minimum value in the input image
  - Initialize a set of regions by these pixels as initial centers of the catchment basins
- For each successive intensity level k:
  - Search all pixels with intensity k
  - For each pixel of intensity k
    - If adjacent to exactly one existing region, add these pixels to that region
    - Else if adjacent to more than one existing regions, mark as boundary
    - Else start a new region
  - Increase k by one

Application: Mammography

Segmentation in digital mammography. (a) Digitized mammogram and radiologist’s boundary for biopsy-proven malignant tumor. (b) Result of watershed algorithm. (c) Suspicious regions determined by automated method. (Images provided courtesy of CE Priebe.)
Edge Detection by Watershed

Watershed algorithm might be used on the gradient image instead of the original image. (Edge Detection)

Due to noise and other local irregularities of the gradient, over-segmentation (breaking images into too many small regions) might occur.
Markers to Avoid Over-segmentation

A solution is to limit the number of regional minima. Use markers to specify the only allowed regional minima.

Markers’ Result

A solution is to limit the number of regional minima. Use markers to specify the only allowed regional minima. (For example, gray-level values might be used as a marker.)
Segmentation by Classification

- Classification methods: pattern recognition / machine learning techniques
- Partition an image by a membership function trained to partition data points in feature space
- Feature space is any user-specified function of images: intensity, mean, moments, etc
- Membership function is trained from a set of example data called training data
- Classifier nicely extends binary segmentation to a process that detects more than one objects!

Supervised and Unsupervised Classification

- Supervised classifier (Classification)
  - Trained with training example data with known categorical labels (music, classic vs contemporary)
  - Training data of this type must be prepared first before segmentation (manually annotated…)
  - As a segmentation, it adds more manual work
- Unsupervised classifier (Clustering)
  - Trained with data without labels (just music)
  - Can use a set of image as training data
  - A single image can be clustered and segmented at the same time (so more compatible to previous methods)
**Characteristic and Membership Function**

- A **characteristic function** is an indicator function denoting whether a pixel is inside or outside a certain pre-determined set corresponding to the $k$-th target object

\[
\chi_k(j) = \begin{cases} 
1 & \text{if } j \in S_k \\
0 & \text{otherwise}
\end{cases}
\]

- We can generalize this to “**membership function**” that is not binary valued and satisfied:

\[
0 \leq m_k(j) \leq 1, \quad \text{for all } j, k
\]

\[
\sum_{k=1}^{K} m_k(j) = 1, \quad \text{for all } j
\]

**Thresholding as Classifier**

- Thresholding is a simplistic classifier
  - Single object: $k = 1$
  - Membership function: **binary characteristic function**
  - Feature space: **intensity value**
  - Training data: **intensity histogram of the image**
  - Training procedure: *pick the optimal threshold value*

\[
g(x, y) = \begin{cases} 
1 & \text{if } (x, y) \in \{(x, y) | f(x, y) > T\} \\
0 & \text{otherwise}
\end{cases}
\]
Advanced Classifiers

• Advanced classifiers extend to...
  • More complex feature space function of images, \( S(f,x,y,z) \)
  • More complex training algorithms, \( m(S) \)

• Membership functions can be derived by using more advanced algorithms such as
  • Fuzzy clustering
  • Classifier algorithms: SVM, Adaboost, LDA, Random Forrest etc
  • Statistical algorithms using probability functions: Bayesian
  • Estimates of partial-volume fractions

For More Information
Take CSC872 Patter Analysis
Machine Intelligence!

Segmentation by Clustering

• Clustering methods divide data points into \( K \) groups in a given feature space
  – Useful for multi-target segmentation where we have more than one objects of interest
  – K-mean clustering
  – Expectation-Maximization algorithm for fitting Mixture of Gaussian model in histogram
  – Markov-Random-Field (MRF) model fitting to images

For More Information
Take CSC872 Patter Analysis
Machine Intelligence!
Segmentation of a magnetic resonance brain image. (a) Original image. (b) K-means segmentation. (c) K-means segmentation with a Markov random field (MRF) model.

**Soft Segmentation**

- The most common approach to addressing partial-volume effects is to produce segmentations that allow regions or classes to overlap, called soft segmentations.
- Soft segmentations retain more information than typical hard segmentation from the original image by allowing uncertainty in the location of the object boundaries.
- Soft segmentations based on *membership functions* can be easily converted to hard segmentations by assigning a pixel to the class with the highest membership value.
Summary

• **Medical Image Segmentation I**
  – Thresholding
  – Region Growing
  – Watersheds
  – Classification

• Midterm 2 in two weeks: 4/10. 400min. Lecture 6-9. Bring a calculator, Bring hand-written notes

• Next Lecture:
  – **Medical Image Segmentation II**
  – Level-Set, Performance evaluation

• Project: 3D visualization/iLearn/Quantification