Image Segmentation and Registration

Dr. Christine Tanner (tanner@vision.ee.ethz.ch)
Computer Vision Laboratory, ETH Zürich

Dr. Verena Kaynig, Machine Learning Laboratory, ETH Zürich
Outline

- **Segmentation**
  - What is it?
  - Tresholding
  - K-means clustering
  - Canny edge detection
  - Graph cut
  - Region growing
  - Level set

- **Registration**
  - What is it?
  - Main components
  - Feature-based
  - Intensity-based
  - Examples
Image Segmentation

„I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have „327“? No. I have sky, house, and trees.“

--Max Wertheimer, „Laws of Organization in Perceptual Forms“ (1923)
Aim of Image Segmentation

- Partition image into a set of regions which
  - are visual distinct and
  - share certain visual properties
    - Intensity
    - Colour
    - Texture
- To simplify representation for easier analysis
Segmentation via Thresholding

Value above threshold: Object
Value below threshold: Background
Automatic Thresholding – Two Classes

1. Choose initial threshold (e.g. randomly)
2. Segment the image into object and background
3. Compute mean intensity of object ($m_1$) and background ($m_2$)
4. Set new threshold to $(m_1+m_2)/2$
5. Repeat from 2. until no change
Segmentation via K-Means Clustering

Use K means → get K classes

1) K initial means \( (m_k) \)
2) Assign each pixel \( p \) to cluster \( k \)
   • s. t. \( \text{argmin}_k |\text{intensity}(p) - m_k| \)
3) Recalculate cluster mean
4) Repeat from 2) until convergence
K-Means Clustering for Color Images

- So far K-means clustering in 1D (intensities)
- Same process for nD by using vector distances
  - e.g. color images (RGB described as 3D-vector)
K-Means Clustering – Number of Clusters

10 clusters
K-Means Clustering - Initialization

6 clusters
K-Means Clustering - Summary

- Result depends on initialization
- Number of clusters is very important
- No spatial considerations
Edge Detection

Why edges?

http://www.funnyjunksite.com
Edge Detection

Task:
Segment the image by finding relevant edges

(http://en.wikipedia.org/wiki/Canny_edge_detector)
Task:
Segment the image by finding relevant edges

Simple Way:
- Smooth the image
- Calculate magnitude of image gradient
- Threshold
Canny Edge Detection

- Aim for “optimal” edge detection algorithm
  - Good detection
    - find all relevant edges
  - Good localization
    - find edge at the right location
  - Minimal response
    - find only relevant edges
Canny Edge Detection

Optimal edges?
- good detection: yes
- good localization: NO
- minimal response: yes/no
Canny Edge Detection – Edge Direction

Improve localization
Classify according to gradient direction
Canny Edge Detection – Thining

Non-maximum suppression
Thin edge in gradient direction: i.e. keep maximum response on 1D profile orthogonal to edge
Canny Edge Detection – Thresholding

Hysteresis thresholding

- To find relevant edges
- Keep strong edges (response > $T_{high}$)
- Keep weaker edges connected to strong edges (response > $T_{low}$ and connectable to $T_{high}$ pixels)
Canny Edge Detection – Result

Initial
Final
Canny Edge Detection - Steps

1. Convolve image with Gaussian filter
2. Compute edges and estimate edge direction
3. Find edge locations using non-maximal suppression
4. Trace edges using hysteresis thresholding
Canny Edge Detection - Summary

- Affected by noise
- No automatic threshold selection
- Useful as preprocessing step
Hybrid Methods

- So far methods based on
  - properties of single pixels (e.g. thresholding, K-mean clustering)
  - relationship between neighbouring pixels (e.g. Canny edge detection)

- Combine these!
Graph Cut

\[ E(y) = \sum_{p \in P} E_d(y_p) + \]

Unary term: cost to not assign label \( y_p \) to pixel \( p \)

Source: label 1
Sink: label 2
Graph Cut

Cut graph such that cost $E(y)$ is minimal

$$E(y) = \sum_{p \in P} E_{d}(y_{p}) + \lambda \sum_{p \in P, q \in N_{2}(p)} E_{s}(y_{p}, y_{q})$$

Unary Term: cost to not assign label $y_{p}$ to pixel $p$

Binary Term: cost to assign label $y_{p}$ to $p$ and $y_{q}$ to $q$
Examples: K-Means refined by Graph Cut

Graph cut

Graph cut
Graph Cut - Summary

- Hybrid method: intensity and edge costs
  ✓ Provides method to solve such a problem
  ✓ Global optimum
  ➢ But high memory usage
Region Growing

- Region based perspective
  - From a (manually selected) seed (pixel or region)
  - Expand boundary to enclose homogenous region (e.g. allowed intensities within range of mean±delta)
  - Leakage – when to stop?
Level Set

- Want smooth boundary enclosing homogenous regions
- Hybrid method: intensities and curvature of boundary

http://en.wikipedia.org/wiki/Level_set_method
Top-down Methods

- So far all methods were bottom-up
- Can we use prior knowledge?
  - What objects are expected in the images?
  - E.g. searching for certain shapes and appearances
  - Important especially for noisy, low-contrast, low-resolution images
- Involves image registration → next
Outline

- **Segmentation**
  - What is it?
  - Tresholding
  - K-means clustering
  - Canny edge detection
  - Graph cut
  - Region growing
  - Level set

- **Registration**
  - What is it?
  - Main components
  - Feature-based
  - Intensity-based
  - Examples
Image Registration

- **Aim:** to establish spatial correspondences

- **Result:** motion vectors, spatial transformation between images

http://www.vcipl.okstate.edu/research_image_registration.html
Main Component: Optimization Criteria

- **Feature-based**
  - Find feature candidates
  - Match features (minimize feature difference)
  - Estimate spatial transformation

- **Intensity-based**
  - Transform source image such that similarity between images is maximized
Main Component: Spatial Transformation

- What spatial transformation is expected?
  - E.g. rigid transformation for bones
  - Helps to constrain problem

- Defines interpolation function
  - From sparse correspondences (e.g. at features)
  - To dense displacement field
# Image Registration Approaches

<table>
<thead>
<tr>
<th>Feature-Based</th>
<th>Intensity-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference image A</td>
<td>Initial transformation T</td>
</tr>
<tr>
<td>Source image B</td>
<td>Transform B</td>
</tr>
<tr>
<td>Match local image features</td>
<td>Transformation T</td>
</tr>
<tr>
<td>Point correspondences</td>
<td>Transformed source image $B^T$</td>
</tr>
<tr>
<td>Fit model to point correspondences</td>
<td>Reference image A</td>
</tr>
<tr>
<td>Transformation model</td>
<td>Calculate image matching criterion</td>
</tr>
<tr>
<td>Final transformation T</td>
<td>Update transformation T</td>
</tr>
<tr>
<td><strong>Optimum?</strong></td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Optimisation</td>
<td>Final transformation T</td>
</tr>
</tbody>
</table>
Landmarks - SIFT Features

- **SIFT** = Scale Invariant Feature Transform
- **Rotation** and **scale** invariant

Landmarks - SIFT Features

1. Scale-space extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor
1. Scale Space

- Convolve image with Gaussian kernel
- Get DoG images
- Downsample by factor of 2
- Repeat
1. Scale Space

Gaussian blurred images

Difference of Gaussian images
1. Scale Space Extrema Detection

Detect extrema of DoG images:

- Compare pixel to its 26 neighbors
  - In 3x3x3 regions
  - At current and adjacent scales
- Current pixel extrema of all neighbours?
2. Keypoint Localization

- Where exactly is the extrema?
  - Fit a 3D quadratic function to the local sample points
  - Determine the interpolated location of the extrema

- Reject extrema with low contrast
2. Keypoint Localization

- Remove edge responses
  - DoG function gives strong response at edges
  - But location along the edge is poorly determined
    - 1 large principal curvature (across edge)
    - 1 small principal curvature (along edge)
  - Eigenvalues ($\alpha > \beta$) of Hessian matrix $H$ are proportional to principle curvatures
    $$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$
  - Reject if ratio of eigenvalues ($r = \alpha/\beta$) is large
3. Orientation Assignment

- Assign one or more orientations to each keypoint
  - Histogram of local image gradient directions in neighborhood
  - Peaks in histograms (>80% of max) define dominant orientations

- Achieves rotation invariance
  - Future operations on images after transformation relative to this orientation, scale, location
4. Keypoint Descriptor

- Compute image gradient magnitude and orientation around keypoint
- Surrounding divided into 4x4 subregions
- Accumulate into orientation histograms (8 bins) relative to keypoint orientation
- Keypoint descriptor of length 128 (=16 subregions*8 bins)

Correspondences indicated by small distance between key point descriptors
Stitching Example

1) Detect keypoints
2) Establish correspondences between keypoints
3) Determine transformation

Transformations

All 2D affine transformations (translation, rotation, scaling, shearing) can be expressed in a 3x3 transformation matrix $A$, i.e. $\mathbf{x}' = A\mathbf{x}$.
Estimate Transformation from Keypoints

- Ordinary least squares
  - Overdetermined linear system of equation
    \[ AX = X' \quad \Rightarrow \quad \hat{A} = X'X^T(XX^T)^{-1} \]
  - Minimizes \( \sum_n (\hat{A}x_n - x_n')^2 \)
Robust Estimation

- Weighted least squares
  - Diagonal weight matrix $W$ from confidence of keypoint correspondences
    $$\hat{A} = X' W X^T (X W X^T)^{-1}$$

- Huber loss ($\xi = \hat{A} x - x'$)
  - Outliers have less influence

$$L_c(\xi) = \begin{cases} 
c|\xi| - \frac{c^2}{2} & \text{for } |\xi| > c, 
\frac{\xi^2}{2} & \text{for } |\xi| \leq c.
c\end{cases}$$
Intensity-based: Image (Dis)Similarity

- Simplest: minimize Mean Squared Differences
  - Optimal if images differ only by Gaussian noise
    \[ D(A, B) = \frac{1}{N_\Omega} \sum_{x \in \Omega} \left( A(x) - B(T(x)) \right)^2 \]
- Linear function: maximize Correlation Coefficient
  - Signals are normalized with respect to mean and standard deviation
    \[ S(A, B) = \frac{1}{N_\Omega} \left| \frac{\sum_{x \in \Omega} (A(x) - \bar{A})(B(T(x)) - \bar{B})}{\sqrt{\sum_{x \in \Omega} (A(x) - \bar{A})^2} \sqrt{\sum_{x \in \Omega} (B(T(x)) - \bar{B})^2}} \right| \in [0,1] \]
Intensity-based: Block Matching Example

- Find translation which maximizes correlation coefficient
Intensity-based: Image Similarity

- **Statistical:** maximize Mutual Information
  - Want to reduce the amount of information in combined image
  - Information measured by Mutual Information
    \[ I(A,B) = H(A) + H(B) - H(A, B) \]
    
    where
    - Entropy
      \[ H(A) = -\sum p(a) \log p(a) \]
    - Joint Entropy
      \[ H(A,B) = -\sum \sum p(a,b) \log p(a,b) \]

Joint histogram:

Probability of intensity pairs in MR/CT brain images

Registered

Misregistered by 2 mm

Misregistered by 5 mm

Example: Contrast-Enhanced MR Mammography

No registration

Rigid registration

Non-rigid registration

Maximum intensity projections of the difference between the post- and pre-contrast MR breast image
Example: Contrast-Enhanced MR Mammography

- Change in image intensity due to contrast
  - Similarity measure: Mutual information
- Deformable parametric transformation
  - Regular grid of control points interpolated by B-Splines
  - Constraint: Volume preservation
- Mean accuracy ~0.5 mm

Example: Segmentation by Registration

- Model-based segmentation
  - Assume repetitive shape/appearance
  - Use learned statistical variation to constrain problem

- Main steps
  - Register training data to common space
  - Probabilistic representation of variation
    - e.g. PCA model of shape and appearance changes
  - Register model to image data

Cootes and Taylor, “Active Shape Models – Smart Snakes”, BMVC 1992,
Outline

- Segmentation
  - What is it?
  - Threshold
  - K-means
  - Graph cut
  - Canny
  - Watersheds

- Registration
  - What is it?
  - Main components
  - Feature-based
  - Intensity-based
  - Examples
Questions?

References

**Image Segmentation**

**Image Registration**