Lecture: Shape Models
FMAN30: Medical Image Analysis

Anders Heyden
2017-11-17
Content

- Representing shapes
- Point distribution models
- Modes of variation
- Projection of a shape to a model
- Segmentation using shape models
- Discussion
Representing shapes

• A continuous curve representing the contour of the object
  – Result from segmentation algorithm
  – Needs to be represented in some way
  – Fourier series, splines, interpolation …

• A sampled set of points along the contour
  – Compact and simple representation
  – A continuous curve can be obtained by interpolation
  – Suitable for shape models
  – Needs to be evenly distributed
  – Advantageous if adapted to the shape
Example of shape representation

Figure 8.12  A contour representing a hand, with possible landmark points marked.
Building a shape model

• Assume that we have a set of $M$ examples of shapes
• This could be a number of pre-segmented shapes
• Select a number of ($N$) landmark points along the shape boundary;

\[ \mathbf{x} = (x_1, x_2, \ldots, x_N) \]

• Each point contains the coordinates of the corresponding landmark point:

\[ x_i = (x_i, y_i) \]
Aligning training data pairwise

- The training data needs to be aligned and evenly distributed, e.g. using arc-length.
- Even distribution can be obtained from interpolation and re-sampling (see Image Analysis course).
- Alignment can be made by finding the best similarity transformation (translation, rotation and scaling) minimizing the distance between corresponding points.
- For two point sets, this can be solved by Procrustes analysis.

\[
\min_{s,R,t} \sum_{i=1}^{N} |x_i^1 - (sR x_i^2 + t)|^2
\]
Aligning the training data set

1. Align each shape to the first
2. Calculate the mean of the transformed shapes (by calculating the mean value for each point)
3. Align the mean shape to the first (to guarantee convergence)
4. Align each shape to the mean shape
5. Update the mean shape
6. Iterate 3 to 5 until convergence

Convergence here means that the mean shape doesn’t change significantly from the previous iteration.
The mean shape

• The mean shape is given by

$$\bar{x}_j = \frac{1}{M} \sum_{i=1}^{M} x_{ij}$$

• Observe that
  – $i$ denotes shape number
  – $j$ denotes point number
Covariance matrix

- Calculate the difference with respect to the mean shape for each shape

\[ \mathbf{d}x^i = \mathbf{x} - \bar{\mathbf{x}} \]

- Calculate the \((2N)\times(2N)\) covariance matrix

\[
S = \frac{1}{M} \sum_{i=1}^{M} \mathbf{d}x^i (\mathbf{d}x^i)^T
\]
Principal variations of the covariance matrix

• Calculate the eigen-vectors and eigen-values of $S$:

$$S'p^i = \lambda_i p^i$$

• The eigen-vectors describe different modes of variations in the training data

• The eigen-values describe the significance of the variation

• Assume that the eigen-values are ordered decreasingly

• The eigen-vector corresponding to the highest eigenvalue describes the most likely variation in the training data.
Using the eigen-vectors as a new basis

• The 2N eigen-vectors constitute a new basis

\[ P = (p^1 p^2 \ldots p^{2N}) \]

• Every shape, \( x \), can now be expressed as

\[ x = \bar{x} + Pb \]

• Here \( b \) denote coordinates in the new basis \( P \)

• The coordinates in \( b \) can be interpreted as the amount of variation in the different variation modes
Selecting a sub-basis

• Usually the first modes describe the relevant variations in the training set and the last modes only capture noise in the training data

• Select the first $t$ modes of variation for the shape descriptor

\[
P_t = (p^1 p^2 \ldots p^t) \]

\[
b_t = (b^1, b^2, \ldots, b^t)^T
\]

• Now a shape $\mathbf{x}$ can be approximated by

\[
\mathbf{x} \approx \bar{\mathbf{x}} + P_t b_t
\]
How to calculate the new coefficients

• Assume that

\[ x \approx \bar{x} + P_t b_t \]

\[ x - \bar{x} \approx P_t b_t \]

• Multiply by the transpose of the basis matrix

\[ P_t^T (x - \bar{x}) \approx P_t^T P_t b_t \]

• The least-squares solution is given by

\[ b_t = (P_t^T P_t)^{-1} P_t^T (x - \bar{x}) = P_t^T (x - \bar{x}) \]
Representation of a general shape

• Given a shape model
• Estimate
  – The mean shape: $\bar{x}$
  – The transformation matrix: $P_t$
  – The shape parameter vector: $b_t$
  – The pose parameters: $s, R, t$
• Now the shape could be written as

$$y = (sR(\bar{x} + P_t b_t) + t)$$

• Let $x = \bar{x} + P_t b_t$ denote points in the shape space and $y$ points in the actual image space
Segmentation using a shape model

• Given a shape model
• Given an image with an edge map
• The mean shape and the type of transformation matrix are known from the shape model
• We need to estimate the pose and the shape parameter vector
• Optimize some criteria over the pose and shape parameters!
Fitting a shape model to edge data

- Initialize $s$, $R$, $t$ and $b_t$
- This gives an initial $y = (y_1, y_2, \ldots, y_N)$
- At each landmark, find the closest edge point orthogonal to the contour, giving new points $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N)$
- Adjust the pose parameter to the best fit to the new points (e.g. Procrustes) based on the current estimate of $b_t$
- Transfer the new points $\hat{y}$ back to the shape space using the inverse similarity transformation, giving new points in the shape space, $\hat{x}$
- Calculate the displacement vector $dx = \hat{x} - x$
- Determine the model adjustment $db_t$ that best approximates $dx$ and update $b_t$
- Iterate until convergence
Fitting a shape model to edge data (cont)

- We need to solve

\[ x + \text{dx} = \bar{x} + P_t (b_t + \text{db}_t) \]

- Using

\[ x = \bar{x} + P_t b_t \]

- Gives

\[ \text{db}_t = P_t^T \text{dx} \]

- Iterate until convergence
Illustration

Figure 8.16 Searching an approximate model fit for target points to which landmarks may move. Courtesy N.D. Efford, School of Computer Studies, University of Leeds.
Initialization

• Segment based on the edge map using your favourite segmentation algorithm
• Estimate the pose parameters using Procrustes
• Estimate the shape parameters using

\[ b_t = P_t^T (x - \bar{x}) \]

Note: It could be a good idea to restrict the shape parameters according to

\[-3 \sqrt{\lambda_t} \leq b_t \leq 3 \sqrt{\lambda_t}\]
Other initialization methods

- Segment the image based on a simple segmentation methods, e.g. thresholding
- Calculate the centre of gravity and the moments and axes of inertia for the binary image obtained from the segmentation as well as the total area
- Select the transformation parameters such that the centre of gravity, axes of inertia and total area coincide with a transformed mean shape
- Use these transformation parameters along with $b=0$ as an initialization
- Points along the segmented shape are now easily obtained from the transformed points on the mean shape.
Illustration

1. Use transformed mean shape as a fist approximation
2. Determine profiles for each model point
3. Compare gray-levels to obtain new points, Y
4. Estimate the model parameters corresponding to the new points by minimizing

\[ |Y - T_{x_{tr}, y_{tr}, s, \theta}(\bar{x} + P_t b)|^2 \]

5. Iterate from 2 until convergence
Combine with a scale pyramid
Results
Representation

\[ X_i = (x_1, \ldots, x_n, y_1, \ldots, y_n) \]
Shape model