Image Segmentation – Part 2

Soma Biswas
Department of Electrical Engineering,
Indian Institute of Science, Bangalore.
Many possible partitions of the image!

Q) How do we pick the correct segmentation?
Normalized Cuts

- Cuts – unnatural bias for partitioning out small sets of points
- New measure: Normalized cut or balanced cut

\[ N_{\text{cut}}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \]

Finds better cut

Min-cut 1

Min-cut 2

better cut
Normalized Cuts

- Volume of set (or association):
  \[ vol(A) = assoc(A,V) = \sum_{u \in A, t \in V} w(u,t) \]

- Define normalized cut: “a fraction of the total edge connections to all the nodes in the graph”:
  \[ Ncut(A, B) = \frac{cut(A, B)}{assoc(A,V)} + \frac{cut(A, B)}{assoc(B,V)} \]

- Cut that partitions out small isolated points will have high \( Ncut \) value
- Cut1 value across node 1 will be 100% of the total connection from that node
Graph-based Image Segmentation

Image (I) → Graph Affinities (W) → Eigenvector X(W) → Discretization

\[ \text{Ncut}(A,B) = \text{cut}(A,B) \left( \frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right) \]

\[ (D - W)X = \lambda DX \]

\[ X_A(i) = \begin{cases} 1 & \text{if } i \in A \\ 0 & \text{if } i \notin A \end{cases} \]
Types of Image Segmentation

- Image analysis
  - Segmentation
    - Basic
      - Thresholding
    - Region growing
    - Active contours
    - Advanced
      - Statistical models
Clustering

- Process of partitioning data (pixels) into subsets called clusters having similar values
  - Intensity, RGB color values, texture properties, etc.
- Subsets close to each other in Euclidean space

- Assume $K$ clusters $C_1, C_2, \ldots, C_K$ with means $m_1, m_2, \ldots, m_K$.
- **Least squares error measure** measures how close the data are to their assigned clusters

$$D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \| x_i - m_k \|^2.$$

- Could consider all possible partitions into $K$ clusters and select the one that minimizes $D$ – computationally infeasible
- Is $K$ known in advance?
K-means Clustering

Form K-means clusters from a set of n-dimensional vectors.

1. Set $i_c$ (iteration count) to 1.

2. Choose randomly a set of $K$ means $m_1(1), m_2(1), \ldots, m_K(1)$.

3. For each vector $x_i$ compute $D(x_i, m_k(i_c))$ for each $k = 1, \ldots, K$ and assign $x_i$ to the cluster $C_j$ with the nearest mean.

4. Increment $i_c$ by 1 and update the means to get a new set $m_1(i_c), m_2(i_c), \ldots, m_K(i_c)$.

5. Repeat steps 3 and 4 until $C_k(i_c) = C_k(i_c + 1)$ for all $k$. 

---

1) $K$ initial "means" (in this case $K=3$) are randomly generated within the data domain (shown in color).

2) $K$ clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3) The centroid of each of the $K$ clusters becomes the new mean.

4) Steps 2 and 3 are repeated until convergence has been reached.
K-means Result

Algorithm guaranteed to terminate, but may not find the global optimum in LS sense

Step 2 modified: partition input vectors into K random clusters and then compute their means.

Figure 10.4: Football image (left) and $K=6$ clusters resulting from a K-means clustering procedure (right) shown as distinct gray tones. The six clusters correspond to the six main colors in the original image: dark green, medium green, dark blue, white, silver, and black.
Histogram Based Method

- Assumption: Homogeneous objects in the image manifest themselves as clusters on the histogram
- Clusters mapped back to the image
- Maximal connected components of the cluster labels re the image segments
- Valleys in histogram determined -> clusters is the interval of values between the valleys
- Multi-modal histograms

![Histograms and images showing threshold ranges]
Ohlander’s Recursive Histogram-Based Technique

- Initial mask selects all pixels in image
- Histogram of masked region is computed
- Clusters determined from histogram
- Connected components in image gives segments
- Each connected component generates a mask

Figure 10.8: Recursive histogram-directed spatial-clustering scheme. The original image has four regions: grass, sky, and two trees. The current mask (shown at upper left) identifies the region containing the sky and the trees. Clustering its histogram leads to two clusters in color space, one for the sky and one for the trees. The sky cluster yields one connected component, while the tree cluster yields two. Each of the three connected components become masks that are pushed onto the mask stack for possible further segmentation.
Region Growing

- Begins at one position in the image
- Attempts to grow each region until the pixels being compared are too dissimilar to the region
- Statistical test to decide whether to grow or not

- region is a population of pixels with similar stats
- region has mean $\overline{X}$ and scatter $S^2$

$$\overline{X} = \frac{1}{N} \sum_{[r,c] \in R} I[r, c]$$  \hspace{2cm} (1)

$$S^2 = \sum_{[r,c] \in R} (I[r, c] - \overline{X})^2.$$

- use a statistical test to see if border pixel $N_1$ should be added to the region
Snakes – Active Contour Models

- Why use it when there are many methods already existing??

**Problems with common existing methods:**

- No prior used and so cannot separate image into constituent components.
- Not effective in presence of noise and sampling artifacts (e.g. medical images).

*Ref: Snakes: Active Contour Models, M. Kass, A. Witkin and D. Terzopoulos, IJCV 1998*
Main Idea

- The problem of finding object boundary is cast as an energy minimization problem.
- A higher level process or a user initializes any curve close to the object boundary.
- The snake deforms and moves towards the desired object boundary.
- The way the contours slither while minimizing their energy, hence the name “snakes”.
- In the end it completely “shrink-wraps” around the object.
- Snakes are active contour models: they lock onto nearby edges, localizing them accurately.
Snakes

- The contour is defined in the \((x,y)\) plane of an image as a parametric curve \(v(s) = (x(s), y(s))\).

- Contour is said to possess an energy \(E_{\text{snake}}\) which is defined as the sum of the three energy terms:

\[
E_{\text{snake}} = E_{\text{internal}} + E_{\text{external}} + E_{\text{constraint}}
\]

- The energy terms are defined cleverly in a way such that the final position of the contour will have a minimum energy \(E_{\text{min}}\).

- Therefore our problem of detecting objects reduces to an energy minimization problem.

What are these energy terms??
Internal Energy

- Depends on the intrinsic properties of the curve.
- Sum of elastic energy and bending energy.

**Elastic Energy** ($E_{\text{elastic}}$):

- Curve treated as elastic rubber band possessing elastic potential energy.
- It discourages stretching by introducing tension.

\[
E_{\text{elastic}} = \frac{1}{2} \int_s \alpha(s) |v_s|^2 \, ds \\
\quad v_s = \frac{dv(s)}{ds}
\]

- Weight $\alpha(s)$ controls elastic energy along different parts of the contour.
- Responsible for shrinking of the contour.
- Considered constant $\alpha$ for many applications.
The snake is also considered to behave like a thin metal strip giving rise to bending energy. It is defined as sum of squared curvature of the contour.

\[ E_{bending} = \frac{1}{2} \int \beta(s) \left| v_{ss} \right|^2 \, ds \]

\( \beta(s) \) plays a similar role to \( \alpha(s) \).

Total internal energy of the snake can be defined as

\[ E_{int} = E_{elastic} + E_{bending} = \int \left( \frac{1}{2} (\alpha \left| v_s \right|^2 + \beta \left| v_{ss} \right|^2) \right) ds \]
External Energy of the Contour

- It is derived from the image.
- Define a function $E_{\text{image}}(x,y)$ so that it takes on its smaller values at the features of interest, such as boundaries.

$$E_{\text{ext}} = \int_s E_{\text{image}}(v(s)) \, ds$$

Key rests on defining $E_{\text{image}}(x,y)$. Some examples

- $E_{\text{image}}(x, y) = -|\nabla I(x, y)|^2$
Total Energy and Solution

- Find a contour $v(s)$ that minimize the energy functional

$$E_{\text{snake}} = \int_s^1 \left( \alpha(s) \left| v_s \right|^2 + \beta(s) \left| v_{ss} \right|^2 \right) + E_{\text{image}}(v(s)) ds$$

- Using variational calculus and by applying Euler-Lagrange differential equation we get following equation

$$\frac{\partial P}{\partial x} - \alpha x'' + \beta x''' = 0$$

- Each term corresponds to a force produced by the respective energy terms. The contour deforms under the action of these forces.
Solution

\[ E(\mathbf{x}(s)) = \int_0^1 F(s, \mathbf{x}(s), \mathbf{x}'(s), \mathbf{x}''(s)) \, ds \]

\[
\left\{ \begin{array}{l}
\mathbf{x} = \mathbf{x}(s) \\
\mathbf{x}' = \frac{\partial \mathbf{x}}{\partial s} \\
\mathbf{x}'' = \frac{\partial^2 \mathbf{x}}{\partial s^2}
\end{array} \right.
\]

This yields the Euler-Lagrange equation for extrema in \(E\):

\[
\frac{\partial F}{\partial \mathbf{x}} - \frac{d}{ds} \left( \frac{\partial F}{\partial \mathbf{x}'} \right) + \frac{d^2}{ds^2} \left( \frac{\partial F}{\partial \mathbf{x}''} \right) = 0
\]

\[
F = P(\mathbf{x}) + \frac{a}{2} \mathbf{x}'^2 + \frac{\beta}{2} \mathbf{x}''^2
\]

\[
\frac{\partial F}{\partial \mathbf{x}} = \frac{\partial P}{\partial \mathbf{x}} \quad \frac{\partial F}{\partial \mathbf{x}'} = a \mathbf{x}' \quad \frac{\partial F}{\partial \mathbf{x}''} = \beta \mathbf{x}''
\]

\[
\frac{\partial P}{\partial \mathbf{x}} - a \mathbf{x}'' + \beta \mathbf{x}''' = 0
\]

Energy Gradient

Unfortunately, these equations are difficult to solve analytically because \(\mathbf{x}\) must be known before \(\partial P/\partial \mathbf{x}\) can be found.
Semi-Implicit Relaxation Methods

\[ a \frac{\partial^2 u_j^t}{\partial s^2} - \beta \frac{\partial^4 u_j^t}{\partial s^4} = \frac{\partial P}{\partial u_j^t} \]

\[ \frac{\partial u}{\partial t} \rightarrow \frac{u_j^{t+1} - u_j^t}{\delta t} \quad \frac{\partial^2 u}{\partial s^2} \rightarrow \frac{u_{j+1}^{t+1} + u_{j-1}^{t+1} - 2u_j^{t+1}}{\delta s^2} \]

\[ \frac{\partial^4 u}{\partial s^4} \rightarrow \frac{u_{j+2}^{t+1} - 4u_{j+1}^{t+1} + 6u_j^{t+1} - 4u_{j-1}^{t+1} + u_{j-2}^{t+1}}{\delta s^4} \]

\[ u^{t+1} = M^{-1}\left(u^t + \delta t \frac{\partial P}{\partial u^t}\right) \]

\[ p = \beta \frac{\delta t}{\delta s^4} \quad q = -a \frac{\delta t}{\delta s^2} - 4\beta \frac{\delta t}{\delta s^4} \quad r = 1 + 2a \frac{\delta t}{\delta s^2} + 6\beta \frac{\delta t}{\delta s^4} \]
Discretizing

- the contour $v(s)$ is represented by a set of control points
  
  \[ v_0, v_1, \ldots, v_{n-1} \]

- The curve is piecewise linear obtained by joining each control point.

- Force equations applied to each control point separately.

- Each control point allowed to move freely under the influence of the forces.

- The energy and force terms are converted to discrete form with the derivatives substituted by finite differences.
Limitations

- Extremely sensitive to parameters.
- No external force acts on points which are far away from the boundary – small capture range
- Convergence is dependent on initial position.
- Fails to detect concave boundaries.

- New kind of snake that permits the snake to start far from the object, and yet still draws it towards the object, and forces it into boundary concavities.
- New external force field, called gradient vector flow, (GVF)
- Field computed as a spatial diffusion of the gradient of an edge map derived from the image.
- Diffuse forces to exist far from the object, and crisp force vectors near the edges.  (http://www.iacl.ece.jhu.edu/static/gvf/)
Statistical Shape and Appearance Models

**Question:** Why do we need them?

**Answer:** Make sense out of images

- Recognise and understand structures
- Analyze complex and variable structures
- Overcome noisy and incomplete data

Ref: *Active Appearance Models, T. Cootes, G. Edwards, C. Taylor, TPAMI 2001*
Main Idea

- “Interpretation through synthesis” approach
- “Explain” novel images by generating similar synthetic images using a parameterized model of appearance.
- Robust segmentation by using the model to constrain solutions to be valid examples of the class of images modelled.

- Other applications: Model codes the appearance of a given image in terms of a compact set of parameters
- Useful for higher-level interpretation of the scene.
- Matches shape and texture simultaneously, resulting in an algorithm that is rapid, accurate, and robust.
- Models can match to any of a class of deformable objects (e.g., any face with any expression, rather than one person’s face with a particular expression).
Model

- Single flexible shape model (a Point Distribution Model)
- Training set of annotated images where corresponding points have been marked on each example.
- Face model, we require face images marked with points defining the main features – shape and texture
Active Shape Model

We apply Procrustes analysis to align the sets of points (each represented as a vector, $x$) and build a statistical shape model.

$\mathbf{x} = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T$
**Procrustes Algorithm**

- **Input:** Landmark points with known correspondence
- **Required:** To find mapping between the two point sets
- **Normalization:** (translation and scaling); Have to find rotation
  - Centering so that center shifts to a common origin
  - Scaling - Sum of squared distance of the points = 1

For point-sets by $X_i \in \mathbb{R}^2$, $i = 1, 2, ..., N_1$ and $Y_j \in \mathbb{R}^2$, $j = 1, 2, ..., N_2$

$$D_{Procrustes}(X, Y) = \sum_{i=1}^{N} \left\| \frac{(X_i - \mu_X)}{\sigma_X} - R(\theta) \frac{(Y_i - \mu_Y)}{\sigma_Y} \right\|^2$$

$$\theta = \arctan \left( \frac{\sum_{i=1}^{N} [X_i^c(2)Y_i^c(1) - X_i^c(1)Y_i^c(2)]}{\sum_{i=1}^{N} [X_i^c(1)Y_i^c(1) + X_i^c(2)Y_i^c(2)]} \right)$$

$$X_i^c = \frac{X_i - \mu_X}{\sigma_X}, \text{ and } Y_i^c = \frac{Y_i - \mu_Y}{\sigma_Y}.$$
Texture Model

- Warp each training image so the points match those of the mean shape, obtaining a “shape-free patch”
- “Texture” is the pattern of intensities or colors across an image patch.

\[ g \rightarrow \frac{g - \mu_g \mathbf{1}}{\sigma_g} \]
PCA (Principal Component Analysis)

- **Orthogonal transformation** converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called **principal components**.

- First principal component has the largest possible **variance** (accounts for maximum variability in the data), followed by the next component and so on.

- All principal components are orthogonal to (i.e., uncorrelated with) the preceding components.

\[
\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \quad \Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n, \quad \Phi_i = \Gamma_i - \Psi
\]

The vectors \( u_k \) and scalars \( \lambda_k \) are the eigenvectors and eigenvalues, respectively, of the covariance matrix

\[
C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T
\]

\[
= AA^T \quad A = [\Phi_1 \Phi_2 \ldots \Phi_M].
\]
Variation Model

Variation model

Example:

\[ \mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{Pb} \]

Labeled brain MR image

Varying the most significant parameter \( b_1 \)

Varying the second most significant parameter \( b_2 \)

Cootes et. Al., Statistical shapes tutorial
Variation Model – Appearance Model

Applying PCA to the normalized data

\[ g \approx \bar{g} + P_g b_g \]

- \( \bar{g} \) Mean normalized grey-level vector
- \( P_g \) Set of orthogonal modes of variation
- \( b_g \) Set of grey level parameters

\[ x \approx \bar{x} + Q_s c \]
\[ g \approx \bar{g} + Q_g c \]

Varying \( c \) changes both shape and texture
Parameters

First two modes of shape variation

First two modes of gray-level variation

400 faces
68 points
10,000 intensity values

First four modes of appearance variation
3.2 Combining shape and texture parameters

Example:

- Shape variation, texture fixed
  \[ x \approx \bar{x} + P_s b_s \quad \text{vary } b_s \]

- Texture variation, shape fixed
  \[ g \approx \bar{g} + P_g b_g \quad \text{vary } b_g \]

- Shape and texture correlated
  \[ x \approx \bar{x} + Q_s c \quad \text{vary } c \]
  \[ g \approx \bar{g} + Q_g c \]

*Brain MR, Cootes et. al. statistical shapes tutorial*
Active Appearance Model Search

- **Input:** Appearance model, a rough initial estimate of the position, orientation, and scale, new image to be interpreted

- **Goal:** Require efficient scheme for adjusting the model parameters, so that the synthetic example generated matches the image as closely as possible.

Current Model Texture:

\[ g_m = \bar{g} + Q_g c. \]

Current difference between model and image:

\[ r(p) = g_s - g_m \]

Model parameters

Sum of squares of elements (scalar measure):

\[ E(p) = r^T r \]

First order Taylor Expansion

\[ r(p + \delta p) = r(p) + \frac{\partial r}{\partial p} \delta p, \]
Active Appearance Model Search

Suppose during matching our current residual is \( r \). We wish to choose \( \delta p \) so as to minimize \( |r(p + \delta p)|^2 \). By equating (3) to zero, we obtain the RMS solution,

\[
\delta p = -Rr(p) \quad \text{where} \quad R = \left( \frac{\partial r}{\partial p}^T \frac{\partial r}{\partial p} \right)^{-1} \frac{\partial r}{\partial p}^T.
\] (4)

- Recalculate \( dr/dp \) at every step is an expensive operation.
- Assume it to be approximately fixed.
- Estimate it once from a training set using numeric differentiation.
- Displace each parameter from the known optimal value on typical images and compute a weighted average of the residuals.
- Precompute \( R \) and use it in all subsequent searches with the model.
- Images used can be examples from the training set.
Results

- Train on 100 hand-labeled face images
- RMS error between model and target points was about 0.8% of face width
Thank You

*Slide Credit:*
Amyn Poonawala
R.Nika Razansky
Dhruv Batra