Texture Based Image Segmentation

Jason Chang
Massachusetts Institute of Technology
Master’s Degree Candidate
October 15, 2008
Outline

• The Goal
• Level Set Methods
• Previous Work
• Extensions / Improvements
  – Bias Field Estimation
  – Texture Based Segmentation
Image Segmentation

- Separate the image into separate regions
- Focus on Binary Segmentation (two regions, one curve)
• Curve evolution is defined by an energy functional to minimize over
• Allows for easy manipulation
• Implicitly define a curve on the image with a surface in 3D
Level Set Methods

- Define a height at every pixel in the image
Level Set Methods

- The zero level set represents the 2D curve
<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>The pixel location</td>
<td>$j \in {1, 2, \ldots, N}$</td>
</tr>
<tr>
<td>$x(j) \equiv x_j$</td>
<td>The intensity value at pixel $j$</td>
<td>$x_j \in {0, 1, \ldots, 255}$</td>
</tr>
<tr>
<td>$\phi(j) \equiv \phi_j$</td>
<td>The level set function at pixel $j$</td>
<td></td>
</tr>
<tr>
<td>$L(j) \equiv L_j = \text{sign}(\phi_j)$</td>
<td>The label assigned to pixel $j$</td>
<td>$L_j \in {+1, -1}$</td>
</tr>
<tr>
<td>$R_{\pm} = {j \mid L_j = \pm 1}$</td>
<td>The segmented regions</td>
<td></td>
</tr>
<tr>
<td>$C = {j \mid \phi_j = 0}$</td>
<td>The curve that segments the image (zero level set)</td>
<td></td>
</tr>
</tbody>
</table>
Segmentation Criterion

- Maximize mutual information between pixel intensity and labeling
  \[ J \sim U\{1, \ldots, N\} \quad I(x_j; L_j) \]

- Approximate MI by using a Kernel Density Estimate to find the PDFs
  \[ \hat{p}_{x_j|J \in R_\pm}(x_j) \equiv \hat{p}_x^\pm(x_j) = \frac{1}{h|R_\pm|} \sum_{s \in R_\pm} K\left(\frac{x_j - x_s}{h}\right) \]

- Use the Gaussian Kernel
  \[ K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \]

Junmo’s Algorithm

• Minimize the energy functional

\[ E(C) = -N \cdot \hat{I}(x_j; L_j) + \alpha \oint_C ds \]

Curve Length Penalty

• Gradient flow that minimizes the energy \( \forall j \in C \)

\[
\frac{\partial \phi_j}{\partial t} = \left[ \log \frac{\hat{p}_x^+(x_j)}{\hat{p}_x^-(x_j)} + \frac{1}{|R_+|} \int_{R_+} K \left( x_i - x_j \right) \frac{\hat{p}_x^+(x_j)}{\hat{p}_x^+(x_j)} di - \frac{1}{|R_-|} \int_{R_-} K \left( x_i - x_j \right) \frac{\hat{p}_x^-(x_j)}{\hat{p}_x^-(x_j)} di \right] \hat{N} - \alpha \kappa \hat{N}
\]

Computationally Intensive

• Approximate Gradient Descent \( \forall j \in C \)

\[
\frac{\partial \phi_j}{\partial t} \approx \log \frac{\hat{p}_x^+(x_j)}{\hat{p}_x^-(x_j)} \hat{N} - \alpha \kappa \hat{N}
\]

Junmo’s Algorithm

• Why is the Scalar Segmentation Algorithm Good?
  – Segmentation based on non-parametric densities
  – No training required
• What needs to be improved?
  – Does not perform well on images with lighting effects
  – Only supports grayscale image segmentation
  – Textured images cannot be segmented
• Assume the observed image is the product of an intrinsic image and a multiplicative gain field.
Bias Field Estimation

• Bias field is the log of the Gain field

\[ x_j = b_j \times g_j \]
\[ y_j = \log(x_j) = \log(b_j) + \log(g_j) = \log(b_j) + \beta_j \]

• Assume that the intrinsic image pixels, \( b_j \), are i.i.d. conditioned on knowing the regions \( R_{\pm} \)

• Find the MAP estimate of \( \beta \) for a given segmentation

\[ \hat{\beta}_{\text{MAP}} = \Lambda_{\beta} f(\beta), \quad [f(\beta)]_j = \sum_i \Pr[L_j = i] \frac{\partial}{\partial \beta_j} \hat{p}_y(y_j | \beta_j, L_j) \]
\[ \sum_i \Pr[L_j = i] \hat{p}_y(y_j | \beta_j, L_j) \]

• Use a fixed-point iteration to find \( \beta \)

\[ \hat{\beta}^{(k+1)} = \Lambda_{\beta} f(\hat{\beta}^{(k)}) \]

Segmentation Algorithm with Bias Field Estimation

1. Assume that the bias field, $\beta$, is zero and the intrinsic image is just the observed image
2. Segment the estimated intrinsic image
3. Estimate the bias field, $\beta$, given the current segmentation
4. Find the estimated intrinsic image, $b$, from the estimated bias field
5. Repeat from Step 2 until convergence
• Alternate between segmentation and bias field estimation
Vector Segmentation

- Extending the formulation to vector values (notice the **bold** vector $\mathbf{r}_i$, instead of the scalar $x_i$)

$$
\frac{\partial \phi_j}{\partial t} \approx \left[ \log \frac{\hat{p}_x^+(\mathbf{v}_j)}{\hat{p}_x^-(\mathbf{v}_j)} \right] \bar{N} - \alpha \kappa \bar{N}
$$

- Vector-valued images can be segmented
  - Color images are segmented using $\mathbf{v}_j = [R, G, B]$
  - Texture images can be segmented by representing each pixel with a texture vector
Steerable Pyramid

Recursively replace the gray box at the red dot

Steerable Pyramid

Original

High-Pass ($v_0$)

Low-Pass ($v_D$)

Varying Orientation

Varying Scale

$(v_1, v_2, v_3, \ldots, v_{D-1})$
Dimensionality Reduction

- Estimating PDFs using the Improved Fast Gauss Transform (Yang et al.) has complexity:

\[ O\left(D^c (M + N)\right) \]

- Problems with high dimensionality
  - Takes ~1 hour to do a KDE on 14 dimensions
  - Sparse data in 14 dimensions provide for poor estimate

Dimensionality Reduction

- We can reconstruct the original image from the outputs
  \[ \mathbf{x} = \sum_{i=0}^{D} \Theta_i \mathbf{z}_i = \sum_{i=0}^{D} \mathbf{v}_i \]

- Approximately reconstruct \( \mathbf{x} \) by using a subset of the outputs
  \[ \hat{\mathbf{x}} = \sum_{i=0}^{D-1} \mathbf{v}_i u_i \quad \mathbf{u} \in \{0, 1\}^D \]

- Define the error of the reconstruction as the MSE
  \[ e(\mathbf{x}, \hat{\mathbf{x}}) = \| \mathbf{x} - \hat{\mathbf{x}} \|_2^2 = (\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}}) \]

- Perform the following optimization
  \[ \mathbf{u}^* = \arg\min_{\mathbf{u} \in \{0, 1\}^D} e(\mathbf{x}, \hat{\mathbf{x}}) \]
  \[ s.t. \ |\mathbf{u}| = \sum_{i=0}^{D-1} u_i = d \]
Pyramid Subset Results

Oriented Stripes

Scaled Stripes

Scaled Checkerboard

Different Textures

(Using d=3)
Smoothly Varying Textures

- Try to capture a texture that varies smoothly in orientation and scale
Smoothly Varying Textures

Smoothly changing orientation
Smoothly Varying Textures

Outputs at 4 orientations and 1 scale
Smoothly Varying Textures

Junmo’s Scalar Segmentation

Vector Segmentation (Pyramid Subset)

Vector Segmentation (Smoothly Varying Textures)
Thanks!

Questions / Comments?