Image Segmentation based on Normalized Cut Framework

Group member: Harry Huang
Harry Chao
Jeffrey Liu

Outline

1. Introduction to image segmentation
2. Normalized cut framework
3. Similarity measurement
4. Experimental result
5. Conclusion
6. Future work
7. Reference
1. Introduction to image segmentation

- Image → Meaningful segments
- It’s an old research topic while still no robust solution
- No benchmark to evaluate the performance

Popular techniques

- **1. Feature-space based techniques**
  Feature → Clustering
  (Ex: mean shift, normalized cut, K-mean) [2,3]

- **2. Image-domain based techniques**
  Go through the image by some criterion
  (Ex: split & merge, region growing, watershed) [1]

- **3. Edge-based techniques**
  Edge detection → Edge linking
2. Normalized cut framework

- Similarity measurement + normalized cut

- What is a cut? Related with “Graph theory”

\[ G = (V, E) \]
\[ A \cup B = V, A \cap B = \emptyset \]
\[ \text{Cut}(A, B) = \sum_{u \in A \times B} w(u, v) \]

Using “Cut” for segmentation

- 1. We model each pixel in the image as the vertex of a graph, and the arc between two vertices is the “similarity” of these two pixels.

- 2. Cut is the total number of “arc quantities” to separate a connective part into two disjoint parts.
Cut types

- Cut types: (1) Minimum cut
  (2) Ratio cut
  (3) Normalized cut

- The formula:
  \[ N_{\text{cut}}(A, B) = \frac{\text{cut}(A, B)}{\text{asso}(A, V)} + \frac{\text{cut}(A, B)}{\text{asso}(B, V)} \]
  \[ N_{\text{asso}}(A, B) = \frac{\text{asso}(A, A)}{\text{asso}(A, V)} + \frac{\text{asso}(A, B)}{\text{asso}(B, V)} \]
  \[ N_{\text{cut}}(A, B) = 2 - N_{\text{asso}}(A, B) \]

Fast algorithm

- The minimum N-cut:
  “the eigenvector with the second smallest eigenvalue”

  \[ \min N_{\text{cut}} = \min_y \frac{y^T(D - W)y}{y^TDy} \quad [8] \]
  \[ (D - W)y = \lambda Dy \]

  For an MxN image, W is the (MN)x(MN) similarity matrix and D is a diagonal matrix containing the total similarity between one pixel to other pixels. “y” is a (MN)x1 vector used to separate the image into “two” parts.

- We can recursively do this “cut process” on each separated part with a stopping mechanism.
Why we use it?

Why we use it? “Global view”
Find the best cutting path from the global view rather than the local view

Disadvantage of other methods:
(1) Texture-based: how to use spatial information
(2) Image-based: only low-level feature
(3) Edge linking: over segmentation

3. Similarity measurement

Color and illumination for non-texture image segmentation
- Gray level image: pixel values
- Color image: RGB, HSV
- Texture based with texton

\[ \chi^2(h_i, h_j) = \frac{1}{2} \sum_{l=1}^{L} \frac{(h_i(l) - h_j(l))^2}{h_i(l) + h_j(l)} \]

Texture and contour information
- Contour and Texture Analysis for Image Segmentation

Adaptive method
- texture and non-texture component
Texture segmentation based on texton

- Texture feature by Filter bank
- RGB for color image with mean square difference
- Color image $\rightarrow$ gray level image
  - Edge filter - 3 scales, 6 orientations
  - Bar filter - 3 scales, 6 orientations
  - Gabor filter - 3 scales and 8 orientations
  - Difference of Gaussian filter
  - Gaussian filter

Filter bank

Original image

Bar filter

Edge filter

DIP Final Project 2009 Fall
Texton

Based on the 62-dimension features of each pixel, we first classify each pixel into K bins, and each bin is a texton.

Similarity Matrix

- $h_i$, $h_j$ are histograms
  \[ x^2(h_i, h_j) = \frac{1}{2} \sum_{l=1}^{t} \frac{[h_i(l) - h_j(l)]^2}{h_i(l) + h_j(l)} \]

- Feature similarity
  \[ W_f = \exp\left(-\frac{x^2(h_i, h_j)}{\delta_f}\right) \]

- Spatial similarity
  \[ W_p = \exp\left(-\frac{(x_i - x_j)^2}{\delta_p}\right), \text{ if } \| (x_i - x_j) \| < r \]

- Similarity
  \[ W_{ij} = W_f \ast W_p \]
4. Experimental result

Experimental Result

Eagle: luminance-based
Experimental Result

people: luminance-based

DIP Final Project 2009 Fall

Experimental Result

Wini: luminance-based

DIP Final Project 2009 Fall
Experimental Result

Landscape: luminance-based

Insect: luminance-based

DIP Final Project 2009 Fall
Experimental Result

Lena: luminance-based

DIP Final Project 2009 Fall

Experimental Result

DIP Final Project 2009 Fall
Experimental Result

texture : Luminance-based

Experimental Result

texture : texton-based
Experimental Result

texture: adaptive-based

Experimental Result

texture: texton-based
5. Conclusion

- It has different results with different initial points in k-means. Also, the Matlab built-in function “eigs” is not stable.
- It’s hard to judge the performance of segmentation.
- Texton-based and Luminance-based have different advantages and disadvantages.

6. Future Work

- Stability
- Better adaptive segmentation method
- Try to merge the over-segmented parts
Reference