BIL 717
Image Processing

Semantic Segmentation

Erkut Erdem
Hacettepe University
Computer Vision Lab (HUCVL)
Review - Markov Random Fields

Unary potential

\[
\begin{array}{cc}
0 & 1 \\
0 & 0 & K \\
1 & K & 0 \\
\end{array}
\]

Pairwise Potential

Example: “label smoothing” grid

\[
\text{Energy}(y; \theta, \text{data}) = \sum_{i} \psi_1(y_i; \theta, \text{data}) + \sum_{i,j \in \text{edges}} \psi_2(y_i, y_j; \theta, \text{data})
\]

D. Hoiem
Review - Solving MRFs with graph cuts

Main idea:

• Construct a graph such that every st-cut corresponds to a joint assignment to the variables \( y \)

• The cost of the cut should be equal to the energy of the assignment, \( E(y; \text{data}) \).

• The minimum-cut then corresponds to the minimum energy assignment, \( y^* = \arg\min_y E(y; \text{data}) \).

* Requires non-negative energies
Review - Solving MRFs with graph cuts

Source (Label 0)

Cost to assign to 1

Sink (Label 1)

Cost to split nodes

Cost to assign to 0

Energy($y; \theta, \text{data}$) = \sum_{i} \psi_{1}(y_{i}; \theta, \text{data}) + \sum_{i,j \in \text{edges}} \psi_{2}(y_{i}, y_{j}; \theta, \text{data})
Review - Solving MRFs with graph cuts

\[
\text{Energy}(y; \theta, \text{data}) = \sum_i \psi_1(y_i; \theta, \text{data}) + \sum_{i,j \in \text{edges}} \psi_2(y_i, y_j; \theta, \text{data})
\]
Code for Image Segmentation

\[ E(x) = \sum_{i} c_i x_i + \sum_{i,j} d_{ij} |x_i - x_j| \]

Global Minimum \((x^*)\)

How to minimize \(E(x)\)?
Review - How does the code look like?

```
Graph *g;

For all pixels p
    /* Add a node to the graph */
    nodeID(p) = g->add_node();

    /* Set cost of terminal edges */
    set_weights(nodeID(p), fgCost(p),
                bgCost(p));

end

for all adjacent pixels p,q
    add_weights(nodeID(p), nodeID(q),
                cost(p,q));
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));
// is the label of pixel p (0 or 1)
```
Graph *g;

For all pixels p
    /* Add a node to the graph */
    nodeID(p) = g->add_node();
    /* Set cost of terminal edges */
    set_weights(nodeID(p), fgCost(p), bgCost(p));
end

for all adjacent pixels p,q
    add_weights(nodeID(p), nodeID(q), cost(p,q));
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));
// is the label of pixel p (0 or 1)
Graph *g;

For all pixels p

    /* Add a node to the graph */
    nodeID(p) = g->add_node();

    /* Set cost of terminal edges */
    set_weights(nodeID(p), fgCost(p), bgCost(p));

end

for all adjacent pixels p,q
    add_weights(nodeID(p), nodeID(q), cost(p,q));
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));
// is the label of pixel p (0 or 1)
Review - How does the code look like?

```c
Graph *g;

For all pixels p

    /* Add a node to the graph */
    nodeID(p) = g->add_node();

    /* Set cost of terminal edges */
    set_weights(nodeID(p), fgCost(p), bgCost(p));

end

define label_p = g->is_connected_to_source(nodeID(p));
// is the label of pixel p (0 or 1)
```

Graph representation:
- Source (0)
- Sink (1)
- Nodes with weights:
  - bgCost(a1)
  - fgCost(a1)
  - bgCost(a2)
  - fgCost(a2)

Edges:
- cost(p,q)

Pixel labels:
- a1 = bg
- a2 = fg
Review - Random Fields in Vision

4-connected; pairwise MRF

$E(x) = \sum_{i,j \in N_4} \theta_{ij}(x_i, x_j)$

Order 2

higher(8)-connected; pairwise MRF

$E(x) = \sum_{i,j \in N_8} \theta_{ij}(x_i, x_j)$

Order 2

MRF with global variables

$E(x) = \sum_{i,j \in N_8} \theta_{ij}(x_i, x_j) + \theta(x_1, \ldots, x_n)$

Order 2

Higher-order MRF

C. Rother
Review - MRF with global potential

GrabCut model [Rother et. al. ‘04]

\[ E(x, \theta^F, \theta^B) = \sum_i F_i(\theta^F)x_i + B_i(\theta^B)(1-x_i) + \sum_{i,j \in N} |x_i - x_j| \]

\[ F_i = -\log \Pr(z_i|\theta^F) \quad \text{and} \quad B_i = -\log \Pr(z_i|\theta^B) \]

\[ R^{\theta^F/B} \text{ Gaussian Mixture models} \]

**Problem:** for unknown \( x, \theta^F, \theta^B \) the optimization is NP-hard! [Vicente et al. ‘09]

C. Rother
**Review - GrabCut: Iterated Graph Cuts**

[Rother et al. Siggraph '04]

Learning of the colour distributions

Graph cut to infer segmentation

\[
\min_{\theta^F, \theta^B} E(x, \theta^F, \theta^B)
\]

Most systems with global variables work like that e.g. [ObjCut Kumar et. al. '05, PoseCut Bray et al. '06, LayoutCRF Winn et al. ’06]

C. Rother
Review - Random Fields in Vision

4-connected; pairwise MRF

\[ E(x) = \sum_{i,j \in N_4} \theta_{ij}(x_i, x_j) \]

Order 2

higher(8)-connected; pairwise MRF

\[ E(x) = \sum_{i,j \in N_8} \theta_{ij}(x_i, x_j) \]

Order 2

MRF with global variables

\[ E(x) = \sum_{i,j \in N_8} \theta_{ij}(x_i, x_j) + \theta(x_1, \ldots, x_n) \]

Order 2

Higher-order MRF

C. Rother
Review - Why Higher-order Functions?

In general $\theta(x_1, x_2, x_3) \neq \theta(x_1, x_2) + \theta(x_1, x_3) + \theta(x_2, x_3)$

Reasons for higher-order RFs:

1. Even better image(texture) models:
   - Field-of Expert [FoE, Roth et al. ‘05]
   - Curvature [Woodford et al. ‘08]

2. Use **global** Priors:
   - Connectivity [Vicente et al. ‘08, Nowozin et al. ‘09]
   - Better encoding label statistics [Woodford et al. ‘09]
   - Convert global variables to global factors [Vicente et al. ‘09]
Semantic Segmentation

• Joint recognition & segmentation
  – segmenting all the objects in a given image and identifying their visual categories

• aka scene parsing or image parsing

• Early studies aim at segmenting out a single object of a known category
  – Borenstein & Ullman, 2002, Liebe & Schiele, 2003,
Early Studies of Semantic Segmentation

• Given an image and object category, to segment the object

• Segmentation should (ideally) be
  • shaped like the object e.g. cow-like
  • obtained efficiently in an unsupervised manner
  • able to handle self-occlusion
Early Studies of Semantic Segmentation
Early Studies of Semantic Segmentation
Early Studies of Semantic Segmentation

Using Normalized Cuts, Shi & Malik, 1997
Early Studies of Semantic Segmentation

Using Normalized Cuts, Shi & Malik, 1997

Borenstein and Ullman, ECCV 2002
Jigsaw approach: Borenstein and Ullman, 2002
Implicit Shape Model - Liebe and Schiele, 2003

Interest Points → Matched Codebook Entries → Probabilistic Voting

R. Fergus
Random Fields for segmentation

\[ l = \text{Image pixels (observed)} \]
\[ h = \text{foreground/background labels (hidden) – one label per pixel} \]
\[ \theta = \text{Parameters} \]

\[ p(h \mid I, \theta) \]

Posterior
Random Fields for segmentation

\( I = \) Image pixels (observed)
\( h = \) foreground/background labels (hidden) – one label per pixel
\( \theta = \) Parameters

\[
p(h \mid I, \theta) \propto p(I, h \mid \theta) = p(I \mid h, \theta) p(h \mid \theta)
\]

1. Generative approach models joint
   \( \rightarrow \) Markov random field (MRF)

2. Discriminative approach models posterior directly
   \( \rightarrow \) Conditional random field (CRF)
Generative Markov Random Field

\[
p(h, I | \theta) = \frac{1}{Z(\theta)} \left[ \prod_i \phi_i(I | h_i, \theta_i) \prod_{ij} \psi_{ij}(h_i, h_j | \theta_{ij}) \right]
\]

Where:
- \( p(h, I | \theta) \) is the joint probability of the labels and the image.
- \( p(I | h, \theta) \) is the likelihood of the image given the labels.
- \( p(h | \theta) \) is the prior distribution of the labels.
- \( Z(\theta) \) is the normalization constant.
- \( \phi_i(I | h_i, \theta_i) \) is the unary potential for each label.
- \( \psi_{ij}(h_i, h_j | \theta_{ij}) \) is the pairwise potential between labels.

The figure illustrates the graph model for an image plane, with labels representing foreground and background, and pixels connected by potential functions.
Conditional Random Field

Dependency on $I$ allows introduction of pairwise terms that make use of image.

For example, neighboring labels should be similar only if pixel colors are similar $\rightarrow$ Contrast term

e.g. Kumar and Hebert 2003
Levin & Weiss [ECCV 2006]

\[ E(h; I) = \sum_i \lambda_i |h - h_{F_i,I}| + \sum_{ij} w(i, j) |h_i - h_j| \]

Consistency with fragments segmentation

Segmentation alignment with image edges

Resulting min-cut segmentation
Semantic Segmentation

Joint Object recognition & segmentation

Goal: Detect and segment test image:

Large set of example segmentation:

Up to 2.000.000 shape templates

\[ E(x,w): \{0,1\}^n \times \{\text{Exemplar}\} \rightarrow \mathbb{R} \]

\[ E(x,w) = \sum_i |T(w)_i - x_i| + \sum_{ij \in N_4} \theta_{ij}(x_i,x_j) \]

“Hamming distance”

[Lempitsky et al. ECCV '08]
Semantic Segmentation
Joint Object recognition & segmentation

UIUC dataset; 98.8% accuracy

[Lempitsky et al. ECCV '08]
C. Rother
Semantic Segmentation
Joint Object recognition & segmentation

\[ E(x, \omega) = \sum_{i} \theta_{i}(\omega, x_{i}) + \sum_{i} \theta_{i}(x_{i}) + \sum_{i} \theta_{i}(x_{i}) + \sum_{i,j} \theta_{ij}(x_{i}, x_{j}) \]

\( x_{i} \in \{1, \ldots, K\} \) for \( K \) object classes

Location
Class (boosted textons)

sky
grass

[TextonBoost; Shotton et al, ‘06]
Semantic Segmentation
Joint Object recognition & segmentation

(a)  
(b) 69.6%
Class+ location
(c) 70.3%
+ edges
(d) 72.2%
+ color

[TextonBoost; Shotton et al, '06]
C. Rother
Semantic Segmentation
Joint Object recognition & segmentation

Good results …
Semantic Segmentation

Joint Object recognition & segmentation

Failure cases…
Nonparametric Scene Parsing via Label Transfer (Liu et al. TPAMI’12)

A non-parametric formulation

input

retrieved images and their annotations

result

groundtruth

window
tree
sky
road
pole
car
building
unlabeled
Nonparametric Scene Parsing via Label Transfer

- Framework consists of three main modules:
  1. **Scene retrieval**: finding nearest neighbors (k-NN approach)
  2. **Dense scene alignment**: dense scene matching (SIFT Flow)

![Diagram of the nonparametric scene parsing system via label transfer]

- **Scene retrieval**:
  - Establish scene retrieval: nearest neighbors
  - **Dense scene alignment**:
    - Establish dense scene alignment
    - **Label transfer**:
      - Warp the annotations from the database
      - Use a Markov random field (MRF) model to label the input image
      - Choose the nearest neighbors with the top matching scores
      - Reconcile multiple voting candidates to the query image

- **System pipeline**
  - There are three key algorithmic components:
    - **Query image**
    - **Scene retrieval**
    - **Nearest neighbors**
    - **Dense scene alignment**
    - **Voting candidates with dense flows**
    - **Label transfer**
    - **Parsing result**
Dense Scene Alignment via SIFT Flow

- SIFT Flow (Liu et al., ECCV 2008)
  - Finds semantically meaningful correspondences among two images by matching local SIFT descriptors.
Dense Scene Alignment via SIFT Flow

- SIFT Flow (Liu et al., ECCV 2008)
  - Finds semantically meaningful correspondences among two images by matching local SIFT descriptors

\[
E(w) = \sum_{p} \min(||s_1(p) - s_2(p + w(p))||_1, t) + \text{data term}
\]

\[
\sum_{p} \eta(|u(p)| + |v(p)|) + \text{small displacement term}
\]

\[
\sum_{(p,q) \in \varepsilon} \min(\lambda|u(p) - u(q)|, d) + \min(\lambda|v(p) - v(q)|, d), \text{ smoothness term}
\]

\[w(p) = (u(p), v(p)) : \text{flow vector at point } p\]
Label Transfer

- A set of voting candidates \(\{s_i; c_i; w_i\}_{i=1:M}\) is obtained from the retrieved images with \(s_i, c_i,\) and \(w_i\) denoting the SIFT image, annotation, and SIFT flow field of the \(i\)th voting candidate.
- A probabilistic MRF model is built to integrate
  - multiple category labels,
  - prior object (category) information
  - spatial smoothness of category labels
- \(\log P(c|I, s, \{s_i, c_i, w_i\}) = \sum_p \psi(c(p); s, \{s'_i\})\)

\[
+ \alpha \sum_p \lambda(c(p)) + \beta \sum_{\{p,q\} \in \varepsilon} \phi(c(p), c(q); I) + \log Z
\]
Label Transfer

- Likelihood term:

\[
\psi(c(p) = l) = \begin{cases} 
\min_{i \in \Omega_{p,l}} \| s(p) - s_i(p + w(p)) \|, & \Omega_{p,l} \neq \emptyset, \\
\tau, & \Omega_{p,l} = \emptyset,
\end{cases}
\]

- \( \Omega_{p,l} = \{ i; c_i(p + w(p)) = l \} \) where \( l=1,...,L \) indicates the index set of the voting candidates whose label is \( l \) after being warped to pixel \( p \).

- \( \tau \) is set to be the value of the maximum difference of SIFT feature: \( \tau = \max_{s_1,s_2,p} \| s_1(p) - s_2(p) \| \)
Label Transfer

• Prior term:

\[ \lambda(c(p) = l) = -\log \text{hist}_l(p) \]

• The prior probability that the object category \( l \) appears at pixel \( p \).
  
  – obtained by counting the occurrence of each object category at each location in the training set
  
  – Location prior
Label Transfer

- Spatial smoothness term:
  \[
  \phi(c(p), c(q)) = \delta[c(p) \neq c(q)] \left( \frac{\xi + e^{-\gamma \|I(p) - I(q)\|^2}}{\xi + 1} \right)
  \]

- The neighboring pixels into having the same label with the probability depending on the image edges:
  - Stronger the contrast, the more likely it is that the neighboring pixels may have different labels.
5.1.1 Evaluation Criterion

As mentioned in Section 3, the LMO database consists of report results on the SUN database, a larger and more SIFT flow in (f). Notice the similarity between (a) and (g), (b) and (h). Our system combines the voting from multiple candidates and generates outputs the parsing of the query in Fig. 8j, which is close to the query Fig. 8a and Fig. 8b, respectively.

We use average pixel-wise recognition rate labeled with 33 object categories using the LabelMe online 2,466 training and 200 test images. The images are densely color scheme shown on the left (hue: orientation; saturation: magnitude). (g), (h), and (i) are the warped version of (c), (d), (e) with respect to the illustration purposes, we set the ground-truth annotation in Fig. 8k. For candidates containing minimum SIFT matching scores. For flow field is visualized in Fig. 8f using the same visualization scheme as in [29], where hue indicates orientation and 3

Some label transfer results are shown in Fig. 10. The input image from the test set is displayed in Fig. 10a. We show the warped images and annotations (Fig. 10a), indicating that SIFT flow successfully matches warped image (Fig. 10d) looks similar to the input annotation is listed in Fig. 10f. Notice that the gray pixels are shown in Figs. 8c, 8d, and 8e, respectively. The SIFT image structures. The scene parsing results output by our system output is "unlabeled" output. For samples 1, 5, 6, 8, and 9, the ground-truth annotation in Fig. 10 is listed in Fig. 10f. Notice that the gray pixels are "unlabeled," but our system does not generate "unlabeled" output.
Parse Results
Because the regularity of the database is the key to the success, we remove the SIFT flow matching, i.e., set the flow vector to be zero for every pixel, and obtain an average recognition rate of 61.23 percent without MRF and 67.96 percent with MRF, shown in Figs. 12d and 12f, respectively. This result is significant because SIFT flow is the bottleneck of the system in terms of speed. A fast implementation of our system consists of removing the dense scene alignment module, and simply performing a grid-to-grid label transfer (the likelihood term in the label transfer module still comes from SIFT descriptor distance).

How would different scene retrieval techniques affect our system? Other than the GIST distance used for retrieving nearest neighbors for the results in Fig. 12, we also use the spatial pyramid histogram intersection of HOG visual words and of the ground-truth annotation, with the corresponding per-class recognition rate displayed in Figs. 12g and 12h, respectively. For this database, GIST performs slightly better than HOG visual words. We also explore an upper bound of the label transfer framework in the ideal scenario of having access to perfect scene matching. In particular, we retrieve the nearest neighbors for each image using their ground truth annotation.