Combining bottom-up + top-down image segmentation

Nils

Machine Learning & Robotics Group – TU Berlin

June 02, 2010

Outline

Introduction

Recap: Image Classification, Object Detection, Image Segmentation

- classification:

$$f(I) : \mathbb{R}^N \to \{0,1\}^C$$

- detection:

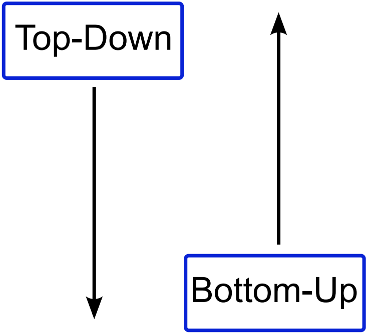
$$f(I) : \mathbb{R}^N \to \mathbb{N}^n \times \mathbb{R}^{n\times 4}$$

- segmentation:

$$f(I) : \mathbb{R}^N \to \mathbb{N}^N$$

Introduction

Bottom-Up & Top-Down Segmentation



Introduction

Bottom-Up & Top-Down Segmentation

Bottom-Up approach

Rely on continuity or region-based principles, e.g.:

- homogeneity of color
- intensity
- texture
- smoothness
- continuity of bounding contours
- ...
- combinations of the above

(a) original

(b) super-pixels

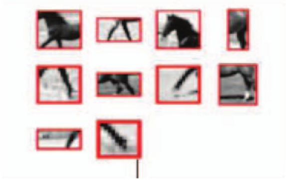
Introduction

Bottom-Up & Top-Down Segmentation

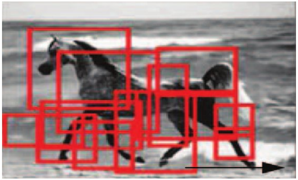
Top-down approach

Exploit class-specific information such as deformable templates, e.g. part-based models:

- variability of shapes
- appearance
- optimally fit contents of image
- delineate figure boundaries



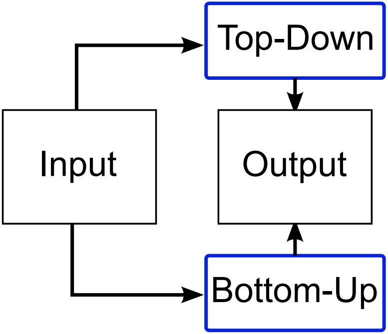
(c) parts



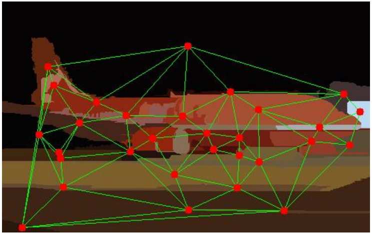
(d) applied to image

Introduction

Idea: Can you blend it?



Bottom-Up Segmentation[1] [2]



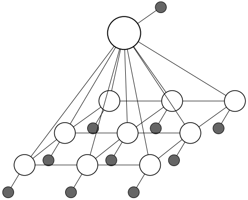


image graph

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$
$\mathcal{V}_L = \{1, 2, \dots, N\}$
$\mathcal{V}_G = \{g\}$
$\mathcal{V} = \mathcal{V}_L \cup \mathcal{V}_G$
$\mathcal{E} = \mathcal{E}_L \cup \mathcal{E}_G$

random variables (RV)

$\mathcal{X} = \{X_i\},\ i \in \mathcal{V}$

discrete labels of RV

$\mathcal{L} = \{l_1, l_2, \dots, l_M\}$

[1] Plath et al. '09, Multi-class image segmentation ...
[2] Gonfaus '10, Harmony Potentials for Joint Classification and Segmentation

Bottom-Up Segmentation

- optimal labeling $\vec{x}^*$ by inferring Maximum A Posteriori (MAP) or, equivalently, minimize global energy function[3]:

$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmin}} \ \mathrm{E}(\vec{x})$$

[3] cf. Hammersley-Clifford theorem

Bottom-Up Segmentation

- optimal labeling $\vec{x}^*$ by inferring Maximum A Posteriori (MAP) or, equivalently, minimize global energy function[3]:

$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmin}} \ \mathrm{E}(\vec{x})$$

energy function of graph $\mathcal{G}$

$$\mathrm{E}(\vec{x}) = \sum_{i \in \mathcal{V}} \phi_L(x_i) + \sum_{(i,j) \in \mathcal{E}_L} \psi_L(x_i, x_j) + \sum_{(i,g) \in \mathcal{E}_G} \psi_G(x_i, \vec{x}_g)$$

[3] cf. Hammersley-Clifford theorem

Bottom-Up Segmentation

- optimal labeling $\vec{x}^*$ by inferring Maximum A Posteriori (MAP) or, equivalently, minimize global energy function[3]:

$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmin}} \; \mathrm{E}(\vec{x})$$

energy function of graph $\mathcal{G}$

$$\mathrm{E}(\vec{x}) = \sum_{i \in \mathcal{V}} \phi_L(x_i) + \sum_{(i,j) \in \mathcal{E}_L} \psi_L(x_i, x_j) + \sum_{(i,g) \in \mathcal{E}_G} \psi_G(x_i, \vec{x}_g)$$

local unary potential

$$\phi_L(x_i) = -k \log \mathrm{P}(x_i | O_i)$$

[3] cf. Hammersley-Clifford theorem

Bottom-Up Segmentation

- optimal labeling $\vec{x}^*$ by inferring Maximum A Posteriori (MAP) or, equivalently, minimize global energy function[3]:

$$\vec{x}^* \quad = \quad \underset{\vec{x}}{\operatorname{argmin}} \; \mathrm{E}(\vec{x})$$

energy function of graph $\mathcal{G}$

$$\mathrm{E}(\vec{x}) = \sum_{i \in \mathcal{V}} \phi_L(x_i) + \sum_{(i,j) \in \mathcal{E}_L} \psi_L(x_i, x_j) + \sum_{(i,g) \in \mathcal{E}_G} \psi_G(x_i, \vec{x}_g)$$

local unary potential

$$\phi_L(x_i) = -k \log \mathrm{P}(x_i | O_i)$$

local potential

$$\psi_L(x_i, x_j) = k' \mathrm{I}[x_i \neq x_j]$$

[3] cf. Hammersley-Clifford theorem

Bottom-Up Segmentation

- optimal labeling $\vec{x}^*$ by inferring Maximum A Posteriori (MAP) or, equivalently, minimize global energy function[3]:

$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmin}} \ \mathrm{E}(\vec{x})$$

energy function of graph $\mathcal{G}$

$$\mathrm{E}(\vec{x}) = \sum_{i \in \mathcal{V}} \phi_L(x_i) + \sum_{(i,j) \in \mathcal{E}_L} \psi_L(x_i, x_j) + \sum_{(i,g) \in \mathcal{E}_G} \psi_G(x_i, \vec{x}_g)$$

local unary potential

$$\phi_L(x_i) = -k \log \mathrm{P}(x_i | O_i)$$

local potential

$$\psi_L(x_i, x_j) = k' \mathrm{I}[x_i \neq x_j]$$

global potential

$$\psi_G(x_i, x_g) = \tilde{k} \mathrm{I}[x_i \notin \vec{x}_g]$$

[3] cf. Hammersley-Clifford theorem

Bottom-Up Segmentation

- optimal labeling $\vec{x}^*$ by inferring Maximum A Posteriori (MAP) or, equivalently, minimize global energy function[3]:

$$\vec{x}^* = \underset{\vec{x}}{\operatorname{argmin}} \ \mathrm{E}(\vec{x})$$

energy function of graph $\mathcal{G}$

$$\mathrm{E}(\vec{x}) = \sum_{i \in \mathcal{V}} \phi_L(x_i) + \sum_{(i,j) \in \mathcal{E}_L} \psi_L(x_i, x_j) + \sum_{(i,g) \in \mathcal{E}_G} \psi_G(x_i, \vec{x}_g)$$

local unary potential

$$\phi_L(x_i) = -k \log \mathrm{P}(x_i | O_i)$$

global unary potential

$$\phi_G(\vec{x}_g) = -\hat{k} \log \mathrm{P}(\vec{x}_g | O_g)$$

local potential

$$\psi_L(x_i, x_j) = k' \mathrm{I}[x_i \neq x_j]$$

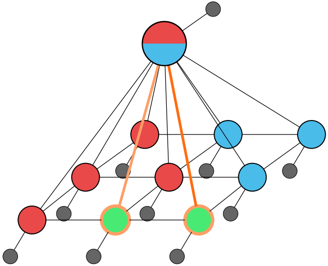
global potential

$$\psi_G(x_i, x_g) = \tilde{k} \mathrm{I}[x_i \notin \vec{x}_g]$$

[3] cf. Hammersley-Clifford theorem

Bottom-Up Segmentation

- SVM for
  - $\phi_L(x_i)$
  - $\phi_G(x_g)$
- inference:
  - message passing
  - graph cut



Bottom-Up Segmentation

- class occurances
- patch/super-pixel confidence
- detect object/non-object areas

Bottom-Up Segmentation

- class occurances
- patch/super-pixel confidence
- detect object/non-object areas
- CANNOT detect instances
  $\Rightarrow$ top-down segmentation

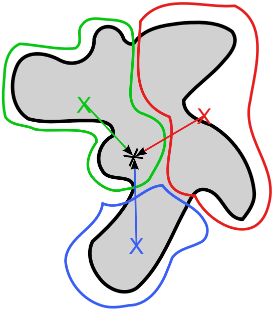
Top-Down Segmentation
Outline

- adaptation of sparse TD algorithm to deal with patches[4]
- construct TD from BU
- for each patch:
  1. feature vector (color, shape, etc.)
  2. occupied part of groundtruth
  3. offset to center of gravity of groundtruth

[4] Leibe '08, Robust Object Detection with Interleaved Categorization and Segmentation

Top-Down Segmentation

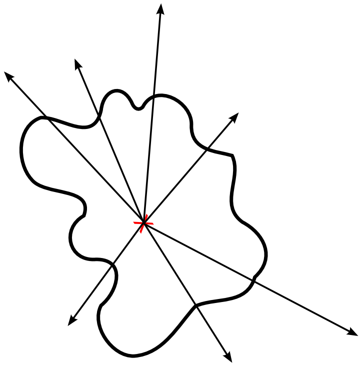
Training



- for each positive training patch, save offset to groundtruth center

Top-Down Segmentation

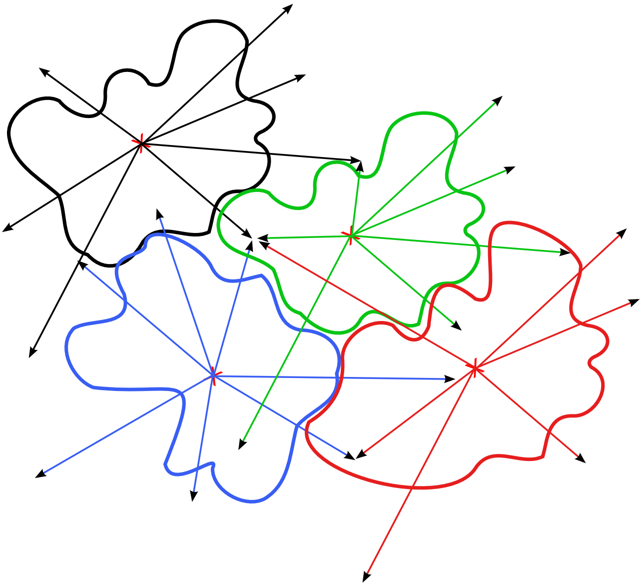
Training



- cluster training patch based on their feature vectors (visual codebook)
- repeat for every object class

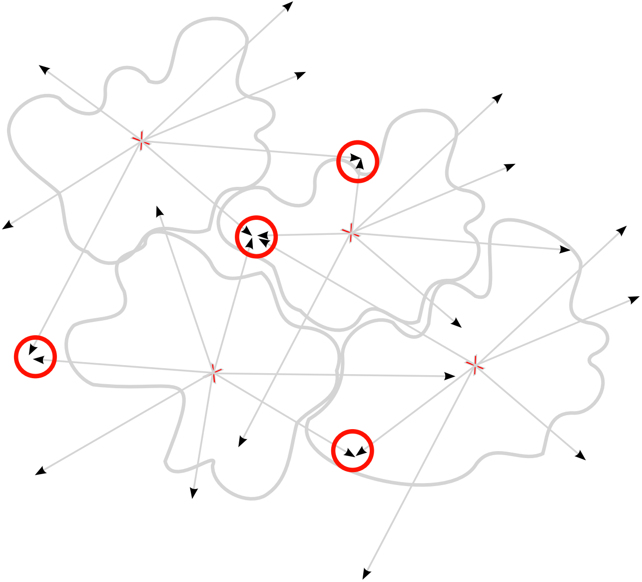
Top-Down Segmentation

Object hypotheses – quantization step



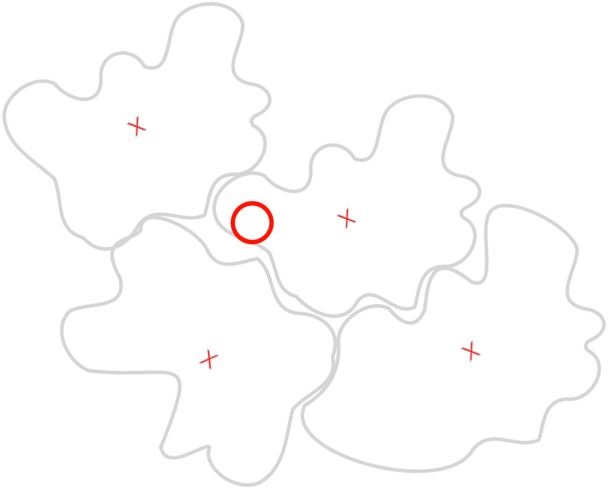
Top-Down Segmentation

Object hypotheses – spatial voting



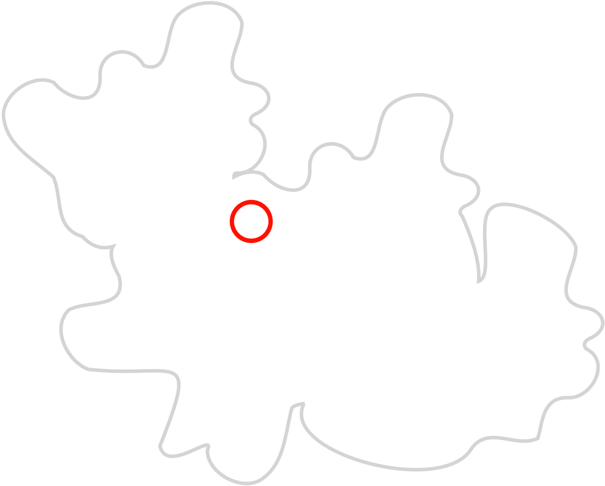
Top-Down Segmentation

Object hypotheses – find maximum



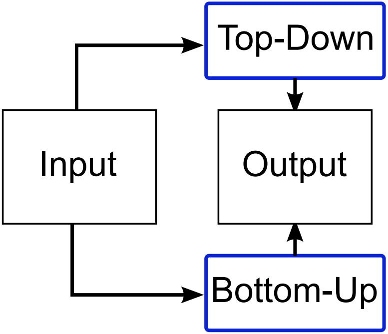
Top-Down Segmentation

Object hypotheses – done



Combining BU and TD

Outline of algorithm



Top-Down Segmentation

Outline of algorithm

1. run BU:
    1.1 determine classes
    1.2 determine labels of pixel (-regions)

Top-Down Segmentation

Outline of algorithm

1. run BU:
   1.1 determine classes
   1.2 determine labels of pixel (-regions)
2. for each foreground patch from BU:
   2.1 select NN from visual codebook
   2.2 project center offsets of matched codeword
   2.3 separately, project groundtruth patch of codeword

Top-Down Segmentation

Outline of algorithm

1. run BU:
   1.1 determine classes
   1.2 determine labels of pixel (-regions)
2. for each foreground patch from BU:
   2.1 select NN from visual codebook
   2.2 project center offsets of matched codeword
   2.3 separately, project groundtruth patch of codeword
3. run spatial voting scheme (Hough vote)
4. save maxima in voting step as object hypotheses

Top-Down Segmentation

Outline of algorithm

1. run BU:
   1.1 determine classes
   1.2 determine labels of pixel (-regions)
2. for each foreground patch from BU:
   2.1 select NN from visual codebook
   2.2 project center offsets of matched codeword
   2.3 separately, project groundtruth patch of codeword
3. run spatial voting scheme (Hough vote)
4. save maxima in voting step as object hypotheses
5. *determine agreement of TD and BU (TBD)*

Ça y est!

Questions and/or comments?