Graphical Models Project

Graph-cut algorithm for image segmentation
Alpha-expansion extension

Maxence Gélard

1st April 2022

1. Introduction
2. Graph-cut for image segmentation
   - Binary segmentation formalisation
   - Alpha-expansion
   - Choice of edges weights: interactive annotation
3. Evaluation and results
4. Conclusion

Introduction: Image segmentation with Graph-Cut

Motivation and goal

- **Image segmentation: assign a given label to every pixel of an image**.
    - Binary: foreground (object) / background
    - Multi-label ($\longrightarrow$ α-*expansion*)

Introduction: Image segmentation with Graph-Cut

Motivation and goal

- **Image segmentation: assign a given label to every pixel of an image**.
  - Binary: foreground (object) / background
  - Multi-label ($\longrightarrow$ α*-expansion*)
- **Energy minimisation: formalisation of the segmentation problem.**
  - Solve using *Graph-cut* techniques.
  - Take advantage of the maximum flow / minimum cut theorem to find the optimal cut.

Example of segmentation

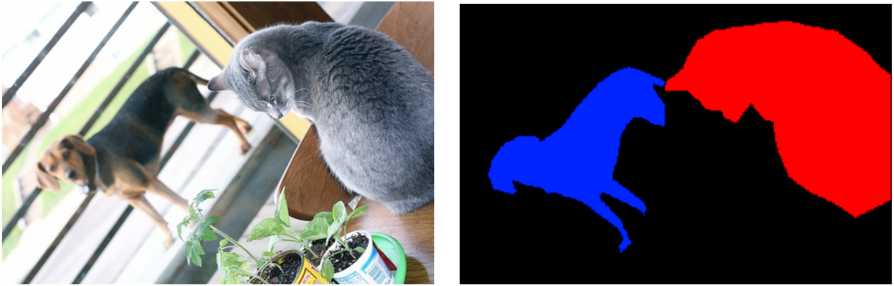


Figure: Example of multi-label segmentation

1 Introduction

2 Graph-cut for image segmentation
- Binary segmentation formalisation
- Alpha-expansion
- Choice of edges weights: interactive annotation

3 Evaluation and results

4 Conclusion

1. Introduction
2. Graph-cut for image segmentation
   - Binary segmentation formalisation
   - Alpha-expansion
   - Choice of edges weights: interactive annotation
3. Evaluation and results
4. Conclusion

Binary segmentation energy

Segmentation energy: first order MRF

- Image as a *4-connected* graph $\mathcal{G} = (\mathcal{P}, \mathcal{N})$
- Labeling matrix $A \in \{0, 1\}^{|\mathcal{P}|}$
- Define an energy: $E(A) = \sum_{p \in \mathcal{V}} R_p(A_p) + \sum_{p,q \in \mathcal{N}} B_{p,q}(A_p, A_q) 1_{A_p \neq A_q}$

Binary segmentation energy

Segmentation energy: first order MRF

- Image as a *4-connected* graph $\mathcal{G} = (\mathcal{P}, \mathcal{N})$
- Labeling matrix $A \in \{0,1\}^{|\mathcal{P}|}$
- Define an energy: $E(A) = \sum_{p \in \mathcal{V}} R_p(A_p) + \sum_{p,q \in \mathcal{N}} B_{p,q}(A_p, A_q) 1_{A_p \neq A_q}$

Weights signification (more later)

- Unary term R_p: data attachment term, penalty if disagreement between label and observation.
- Binary term $B_{p,q}$ penalty terms for similar pixels that would have different labels (that will force a form a smoothness in the label assignments matrix A).

Graph cut method

Transformation to s-t graph

- Under certain assumption (positivity and submodularity of binary potentials), graph cut can solve energy minimisation problem.
- Add source and sink nodes to be able to solve maximum flow problem
- Binary potentials: weights of edges between pixels
- Unary potentials: weights between every pixel and the sink and source nodes

Graph cut method

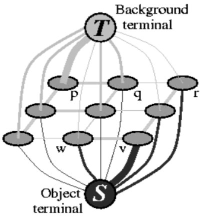


Figure: s-t graph for binary segmentation

edge	weight (cost)	for
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{N}$
$\{p,S\}$	$\lambda \cdot R_p(\text{“bkg”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p,T\}$	$\lambda \cdot R_p(\text{“obj”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \in \mathcal{O}$
	K	$p \in \mathcal{B}$

Figure: Edges weights for graph cut [BJ01]

with $K = 1 + max_{p \in \mathcal{P}} \sum_{q \in \mathcal{P}:(p,q) \in \mathcal{N}} B_{p,q}$

Algorithm running

Maximum flow: Ford-Fulkerson

Start with a flow of 0 on all edges

Until the sink and the source are disjoint in the residual graph:

- Find an s-t path in the residual graph
- On this path, use all maximum allowable flow
- Subtract this flow from the inverse arcs and add it to the forward arcs.

Label assignments

- If $v \in S$, $A_s = 1$ (background).
- If $v \in T$, $A_s = 0$ (foreground).

α-expansion: iteration of the binary case

Main idea behind α-expansion

- Initialize the labels (minimum unary cost across the labels)
- Select a class α
- Solve a binary segmentation problem: for each pixel, the graph-cut output will decide whether it keeps its current label or changes to class α.

α-expansion: refinement

Add auxiliary nodes

Auxiliary node between 2 neighbors that have different labels: new edges created between the 2 neighbors and the auxiliary node (respectively $e_{p,a}$ and $e_{q,a}$) with weight corresponding to the binary decision "α or $\bar{\alpha}$"

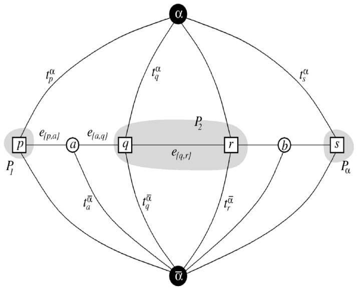


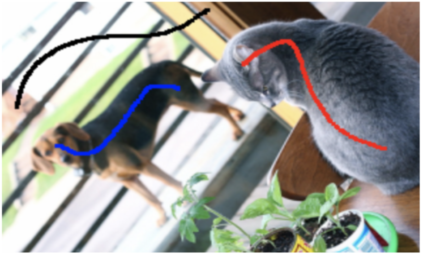
Figure: s-t graph (α-expansion) [BVZ01]

Choice of weights

Unary potentials

- Interactively select 2 sets of pixels $\mathcal{O}$ and $\mathcal{B}$ respectively corresponding to the foreground (object) and the background
- **L_2-distance**: $R_p(label) = ||p - \mu_{label}||^2$, with p the current pixel, *label* either designated the foreground or the background, and μ_{label} the mean of the distribution of the pixels assigned to this label.
- **Normal density**: $R_p(label) = \mathcal{N}(p; \mu_{label}, \Sigma_{label})$, with $\mathcal{N}$ the normal distribution, and Σ_{label} the covariance matrix of the distribution of the pixels assigned to this label.

Choice of weights



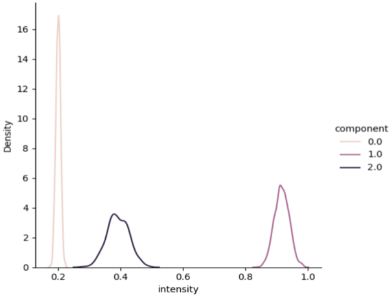


Figure: Interactive annotation

Choice of weights

Binary potentials

- positive weights such that two similar pixels will be assigned a higher weight and conversely.
- $B_{p,q}(A_p, A_q) = \omega e^{-\frac{||p-q||^2}{2\sigma^2}} 1_{A_p \neq A_q}$
- $0 = B_{p,q}(0,0) + B_{p,q}(1,1) \leq B_{p,q}(0,1) + B_{p,q}(1,0)$

Evaluation

Evaluation methods

- *COCO* dataset from Microsoft
- 25 samples with Cats and / or dogs classes, with ground truth segmentation masks.


Figure: Samples from COCO datasets

Evaluation Metric

Intersection over Union

- Metric for a given label L: $\mathcal{G}$, ground truth pixels that have labels L, $\mathcal{P}$ predicted pixels that have label L. $IoU_L = \frac{|\mathcal{G} \cap \mathcal{P}|}{|\mathcal{G} \cup \mathcal{P}|}$.
- Then, we average over all labels to get our final score: $IoU = \frac{1}{n_{labels}} \sum_{L \in labels} IoU_L$.

Segmentation results (L_2 potentials)		
Algorithm	Mean *IoU*	Max *IoU*
Binary	0.50	0.72
α-expansion	0.28	0.49

Segmentation results (normal potentials)		
Algorithm	Mean *IoU*	Max *IoU*
Binary	0.59	0.79
α-expansion	0.33	0.57

Figure: Results table

Result example

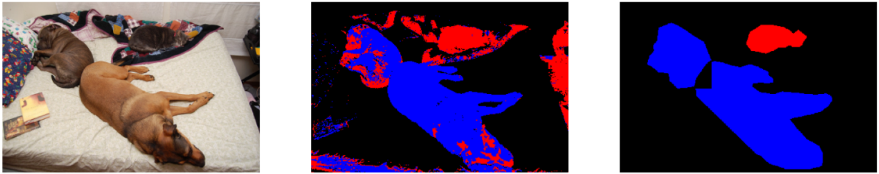
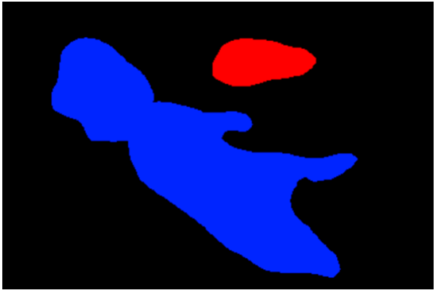


Figure: Segmentation results (left: image, center: result, right: ground truth

Mask RCNN Comparison



Mask RCNN Segmentation results		
Algorithm	Mean *IoU*	Max *IoU*
Binary	0.58	0.95
α-expansion	0.54	0.92

Figure: Results Mask-RCNN

Conclusion

Conclusion, move forward

- Successfully implemented binary and multi-labels segmentation based on the Ford-Fulkerson maximum flow algorithm.
- Main advantage: no training ($\neq$ Deep Learning), with still good quality results. But high time complexity (Ford-Fulkerson: $O(m^2U)$)
- Trade-off between data attachment (unary terms) and smoothness (binary terms): not always perfect (especially on COCO dataset)
- Possible continuation: different maximum flow algorithm such that Edmonds Karp or Dinitz algorithms.

References

Yuri Y Boykov and M-P Jolly, *Interactive graph cuts for optimal boundary & region segmentation of objects in nd images*, Proceedings eighth IEEE international conference on computer vision. ICCV 2001, vol. 1, IEEE, 2001, pp. 105–112.

Yuri Boykov, Olga Veksler, and Ramin Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Transactions on pattern analysis and machine intelligence **23** (2001), no. 11, 1222–1239.