Machine Learning – Lecture 17

Efficient MRF Inference with Graph Cuts

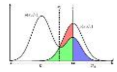
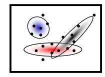
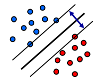
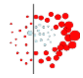
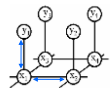
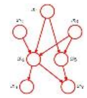
07.07.2015

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de

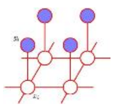
Course Outline

- Fundamentals (2 weeks)
  - Bayes Decision Theory
  - Probability Density Estimation
- Discriminative Approaches (5 weeks)
  - Linear Discriminant Functions
  - Statistical Learning Theory & SVMs
  - Ensemble Methods & Boosting
  - Decision Trees & Randomized Trees
- Generative Models (4 weeks)
  - Bayesian Networks
  - Markov Random Fields
  - Exact Inference
  - Applications

Recap: MRF Structure for Images

- Basic structure

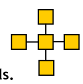


Noisy observations

"True" image content

- Two components
  - Observation model
    - How likely is it that node x_i has label L_i given observation y_i?
    - This relationship is usually learned from training data.
  - Neighborhood relations
    - Simplest case: 4-neighborhood
    - Serve as smoothing terms.

    ⇒ Discourage neighboring pixels to have different labels.
    - This can either be learned or be set to fixed "penalties".

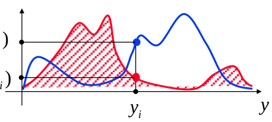


Recap: How to Set the Potentials?

- Unary potentials
  - E.g. color model, modeled with a Mixture of Gaussians

$$\phi(x_i, y_i; \theta_\phi) = -\theta_\phi \log \sum_k p(k \mid x_i) \mathcal{N}(y_i \mid \bar{y}_k, \Sigma_k)$$

⇒ Learn color distributions for each label



Recap: How to Set the Potentials?

- Pairwise potentials
  - Potts Model

$$\psi(x_i, x_j; \theta_\psi) = \theta_\psi \delta(x_i \neq x_j)$$

    - Simplest discontinuity preserving model.
    - Discontinuities between any pair of labels are penalized equally.
    - Useful when labels are unordered or number of labels is small.
  - Extension: "contrast sensitive Potts model"

$$\psi(x_i, x_j, g_{ij}(y); \theta_\psi) = \theta_\psi g_{ij}(y) \delta(x_i \neq x_j)$$

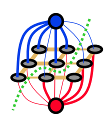
  - where,

$$g_{ij}(y) = e^{-\beta \|y_i - y_j\|^2} \qquad \beta = 2 / avg\left(\|y_i - y_j\|^2\right)$$

    - Discourages label changes except in places where there is also a large change in the observations.

Topics of This Lecture

- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Graph construction
  - Extension to non-binary case
  - Applications



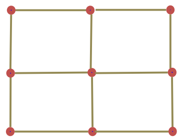
Example: Image Segmentation

Machine Learning, Summer '15

E: $\{0,1\}^N \rightarrow \mathbb{R}$ $\quad N$ = number of pixels
$0 \rightarrow$ fg
$1 \rightarrow$ bg

$$E(X) = \sum_i c_i(bg)x_i + c_i(fg)(1-x_i) + \sum_{ij} c_{ij}[x_j(1-x_i) + x_i(1-x_j)]$$

Image (D)



Slide credit: Pushmeet Kohli

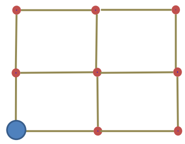
Example: Image Segmentation

Machine Learning, Summer '15

E: $\{0,1\}^N \rightarrow \mathbb{R}$ $\quad N$ = number of pixels
$0 \rightarrow$ fg
$1 \rightarrow$ bg

$$E(X) = \boxed{\sum_i c_i(bg)x_i + c_i(fg)(1-x_i)} + \sum_{ij} c_{ij}[x_j(1-x_i) + x_i(1-x_j)]$$

Unary Cost c_i(bg)



Dark (negative) Bright (positive)

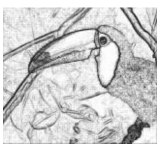
Slide credit: Pushmeet Kohli

Example: Image Segmentation

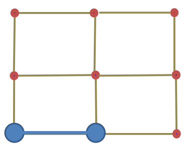
Machine Learning, Summer '15

E: $\{0,1\}^N \rightarrow \mathbb{R}$ $\quad N$ = number of pixels
$0 \rightarrow$ fg
$1 \rightarrow$ bg

$$E(X) = \sum_i c_i(bg)x_i + c_i(fg)(1-x_i) + \boxed{\sum_{ij} c_{ij}[x_j(1-x_i) + x_i(1-x_j)]}$$



Discontinuity Cost (c_{ij})



Slide credit: Pushmeet Kohli

Example: Image Segmentation

Machine Learning, Summer '15

E: $\{0,1\}^N \rightarrow \mathbb{R}$ $\quad N$ = number of pixels
$0 \rightarrow$ fg
$1 \rightarrow$ bg

$$E(X) = \sum_i c_i(bg)x_i + c_i(fg)(1-x_i) + \sum_{ij} c_{ij}[x_j(1-x_i) + x_i(1-x_j)]$$

Global Minimum ($\mathbf{x}^*$)

$$X^* = \arg\min E(X)$$

How to minimize $E(\mathbf{x})$?

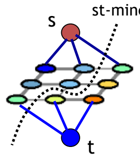
Slide credit: Pushmeet Kohli

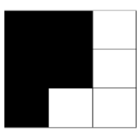
Graph Cuts - Basic Idea

Machine Learning, Summer '15

- Construct a graph such that:
  1. Any st-cut corresponds to an assignment of $\mathbf{x}$
  2. The cost of the cut is equal to the energy of $\mathbf{x}$: $E(\mathbf{x})$

$E(\mathbf{x})$





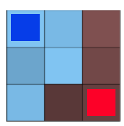
Solution

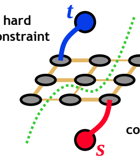
Slide credit: Pushmeet Kohli

Graph Cuts for Binary Problems

Machine Learning, Summer '15

- Idea: convert MRF into source-sink graph





Minimum cost cut can be computed in polynomial time

(max-flow/min-cut algorithms)

Slide credit: Yuri Boykov

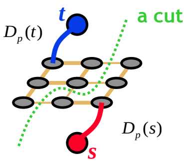
[Boykov & Jolly, ICCV'01]

Simple Example of Energy

unary potentials — pairwise potentials

$$E(L) = \sum_p D_p(L_p) + \sum_{pq \in N} w_{pq} \cdot \delta(L_p \neq L_q)$$

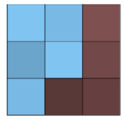
t-links — n-links

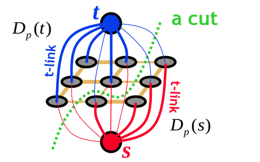


$$L_p \in \{s, t\}$$

(binary object segmentation)

Adding Regional Properties





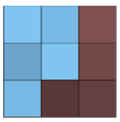
Regional bias example

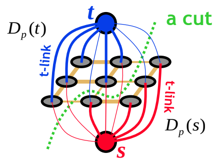
Suppose μ_s and μ_t are given "expected" intensities of object and background

$$p(I_p \mid s) \propto \exp\left(-\| I_p - \mu_s \|^2 / 2\sigma_s^2\right)$$
$$p(I_p \mid t) \propto \exp\left(-\| I_p - \mu_t \|^2 / 2\sigma_t^2\right)$$

NOTE: hard constrains are not required, in general.

Adding Regional Properties





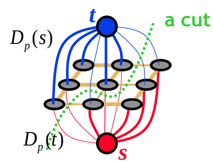
"expected" intensities of object and background μ_s and μ_t can be re-estimated

$$p(I_p \mid s) \propto \exp\left(-\| I_p - \mu_s \|^2 / 2\sigma_s^2\right)$$
$$p(I_p \mid t) \propto \exp\left(-\| I_p - \mu_t \|^2 / 2\sigma_t^2\right)$$

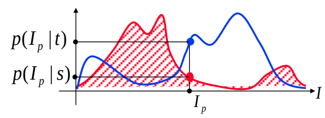
EM-style optimization

Adding Regional Properties

- More generally, unary potentials can be based on any intensity/color models of object and background.



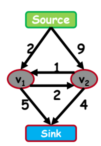
$$D_p(L_p) = -\log(p(I_p \mid L_p))$$



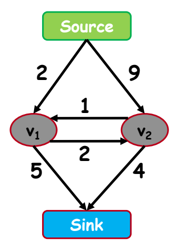
Object and background color distributions

Topics of This Lecture

- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Graph construction
  - Extension to non-binary case
  - Applications



How Does it Work? The st-Mincut Problem



Graph (V, E, C)

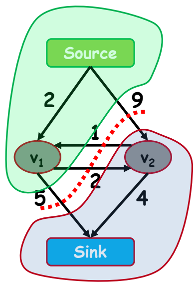
Vertices V = $\{v_1, v_2 \ldots v_n\}$

Edges E = $\{(v_1, v_2) \ldots\}$

Costs C = $\{c_{(1,2)} \ldots\}$

Machine Learning, Summer '15

The st-Mincut Problem



5 + 2 + 9 = 16

What is an st-cut?

An st-cut (S,T) divides the nodes between source and sink.

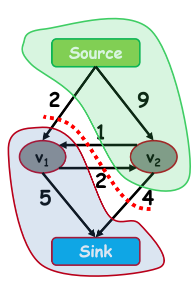
What is the cost of a st-cut?

Sum of cost of all edges going from S to T

Slide credit: Pushmeet Kohli

Machine Learning, Summer '15

The st-Mincut Problem



2 + 1 + 4 = 7

What is an st-cut?

An st-cut (S,T) divides the nodes between source and sink.

What is the cost of a st-cut?

Sum of cost of all edges going from S to T

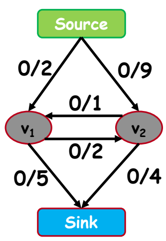
What is the st-mincut?

st-cut with the minimum cost

Slide credit: Pushmeet Kohli

Machine Learning, Summer '15

How to Compute the st-Mincut?



Solve the dual maximum flow problem

Compute the maximum flow between Source and Sink

Constraints

Edges: Flow < Capacity

Nodes: Flow in = Flow out

Min-cut/Max-flow Theorem

In every network, the maximum flow equals the cost of the st-mincut

Slide credit: Pushmeet Kohli

Machine Learning, Summer '15

History of Maxflow Algorithms

Augmenting Path and Push-Relabel

year	discoverer(s)	bound
1951	Dantzig	$O(n^2mU)$
1955	Ford & Fulkerson	$O(m^2U)$
1970	Dinitz	$O(n^2m)$
1972	Edmonds & Karp	$O(m^2 \log U)$
1973	Dinitz	$O(nm \log U)$
1974	Karzanov	$O(n^3)$
1977	Cherkassky	$O(n^2m^{1/2})$
1980	Galil & Naamad	$O(nm \log^2 n)$
1983	Sleator & Tarjan	$O(nm \log n)$
1986	Goldberg & Tarjan	$O(nm \log(n^2/m))$
1987	Ahuja & Orlin	$O(nm + n^2 \log U)$
1987	Ahuja et al.	$O(nm \log(n\sqrt{\log U}/m))$
1989	Cheriyan & Hagerup	$E(nm + n^2 \log^2 n)$
1990	Cheriyan et al.	$O(n^3/\log n)$
1990	Alon	$O(nm + n^{8/3} \log n)$
1992	King et al.	$O(nm + n^{2+\epsilon})$
1993	Phillips & Westbrook	$O(nm(\log_{m/n} n + \log^{2+\epsilon} n))$
1994	King et al.	$O(nm \log_{m/(n \log n)} n)$
1997	Goldberg & Rao	$O(m^{3/2} \log(n^2/m) \log U)$ $O(n^{2/3}m \log(n^2/m) \log U)$

n: #nodes

m: #edges

U: maximum edge weight

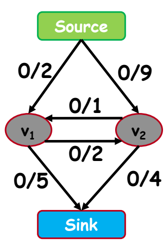
Algorithms assume non-negative edge weights

Slide credit: Andrew Goldberg

Machine Learning, Summer '15

Maxflow Algorithms

Flow = 0



Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

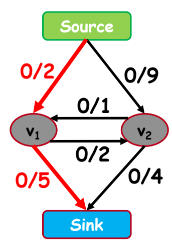
Algorithms assume non-negative capacity

Slide credit: Pushmeet Kohli

Machine Learning, Summer '15

Maxflow Algorithms

Flow = 0



Augmenting Path Based Algorithms

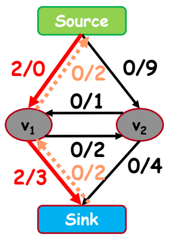
1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

Algorithms assume non-negative capacity

Slide credit: Pushmeet Kohli

Maxflow Algorithms

Flow = 0 + 2



Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

--- edge is created in the residual graph with capacity equals to the flow passed through the edge

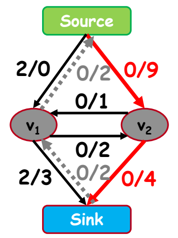
Slide credit: Pushmeet Kohli

B. Leibe

29

Maxflow Algorithms

Flow = 2



Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

--- edge is created in the residual graph with capacity equals to the flow passed through the edge

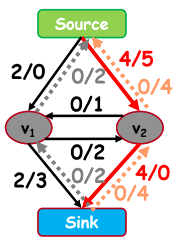
Slide credit: Pushmeet Kohli

B. Leibe

32

Maxflow Algorithms

Flow = 2 + 4



Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

--- edge is created in the residual graph with capacity equals to the flow passed through the edge

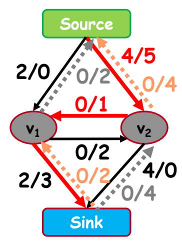
Slide credit: Pushmeet Kohli

B. Leibe

33

Maxflow Algorithms

Flow = 6



Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

--- edge is created in the residual graph with capacity equals to the flow passed through the edge

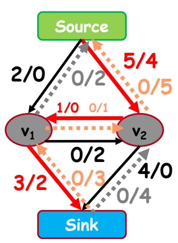
Slide credit: Pushmeet Kohli

B. Leibe

34

Maxflow Algorithms

Flow = 6



Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

--- edge is created in the residual graph with capacity equals to the flow passed through the edge

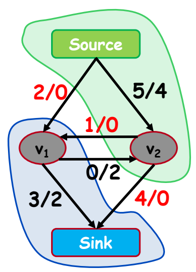
Slide credit: Pushmeet Kohli

B. Leibe

35

Maxflow Algorithms

Flow = 7



Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

Algorithms assume non-negative capacity

Slide credit: Pushmeet Kohli

B. Leibe

38

When Can s-t Graph Cuts Be Applied?

$$E(L) = \underbrace{\sum_p E_p(L_p)}_{\text{unary potentials (t-links)}} + \underbrace{\sum_{pq \in N} E(L_p, L_q)}_{\text{pairwise potentials (n-links)}} \qquad L_p \in \{s,t\}$$

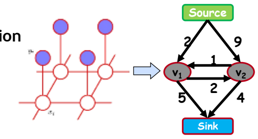
- s-t graph cuts can only globally minimize binary energies that are submodular. [Boros & Hummer, 2002, Kolmogorov & Zabih, 2004]

$E(L)$ can be minimized by s-t graph cuts	$\Leftrightarrow$	$E(s,s)+E(t,t) \le E(s,t)+E(t,s)$ Submodularity ("convexity")

- Submodularity is the discrete equivalent to convexity.
  - $\Rightarrow$ Solution will be globally optimal.

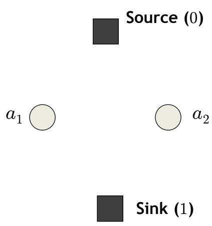
Topics of This Lecture

- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Graph construction
  - Extension to non-binary case
  - Applications



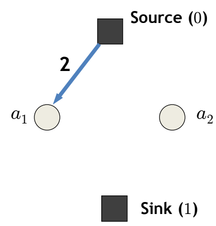
Example: Graph Construction

$E(a_1, a_2)$



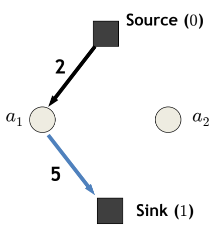
Example: Graph Construction

$E(a_1, a_2) = 2a_1$



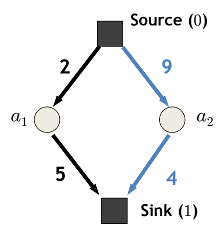
Example: Graph Construction

$E(a_1, a_2) = 2a_1 + 5(1-a_1)$



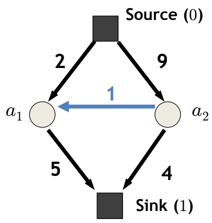
Example: Graph Construction

$E(a_1, a_2) = 2a_1 + 5(1-a_1) + 9a_2 + 4(1-a_2)$



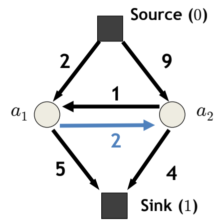
Example: Graph Construction

$E(a_1,a_2) = 2a_1 + 5(1-a_1) + 9a_2 + 4(1-a_2) + (1-a_1)a_2$



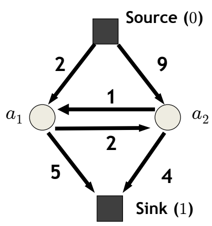
Example: Graph Construction

$E(a_1,a_2) = 2a_1 + 5(1-a_1) + 9a_2 + 4(1-a_2) + (1-a_1)a_2 + 2(1-a_2)a_1$



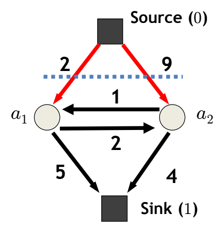
Example: Graph Construction

$E(a_1,a_2) = 2a_1 + 5(1-a_1) + 9a_2 + 4(1-a_2) + (1-a_1)a_2 + 2(1-a_2)a_1$



Example: Graph Construction

$E(a_1,a_2) = 2a_1 + 5(1-a_1) + 9a_2 + 4(1-a_2) + (1-a_1)a_2 + 2(1-a_2)a_1$



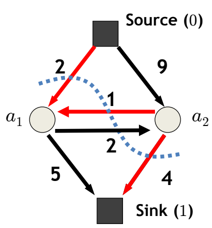
Cost of cut = 11

$a_1 = 1 \; a_2 = 1$

$E\ (1,1) = 11$

Example: Graph Construction

$E(a_1,a_2) = 2a_1 + 5(1-a_1) + 9a_2 + 4(1-a_2) + (1-a_1)a_2 + 2(1-a_2)a_1$



Cost of cut = 7

$a_1 = 1 \; a_2 = 0$

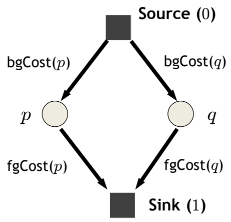
$E\ (1,0) = 7$

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);
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));

// is the label of pixel p (0 or 1)
```

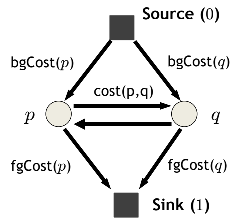
How Does the Code Look Like?


```
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);
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));

// is the label of pixel p (0 or 1)
```



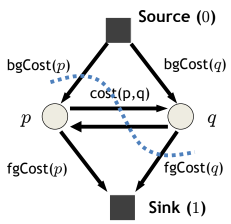
How Does the Code Look Like?


```
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);
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));

// is the label of pixel p (0 or 1)
```



How Does the Code Look Like?

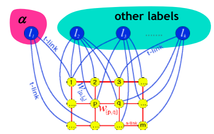

```
Graph *g;

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);
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));

// is the label of pixel p (0 or 1)
```



p = bg q = fg

Topics of This Lecture

- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Graph construction
  - Extension to non-binary case
  - Applications

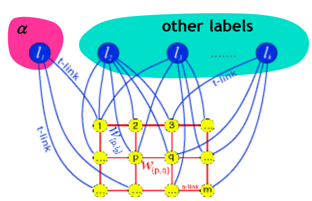


Dealing with Non-Binary Cases

- Limitation to binary energies is often a nuisance.
  ⇒ E.g. binary segmentation only…
- We would like to solve also multi-label problems.
  - The bad news: Problem is NP-hard with 3 or more labels!
- There exist some approximation algorithms which extend graph cuts to the multi-label case:
  - α-Expansion
  - $\alpha\beta$-Swap
- They are no longer guaranteed to return the globally optimal result.
  - But α-Expansion has a guaranteed approximation quality and converges in a few iterations.

α-Expansion Move

- Basic idea:
  - Break multi-way cut computation into a sequence of binary s-t cuts.

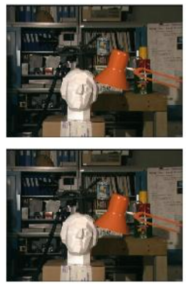


α-Expansion Algorithm

1. Start with any initial solution
2. For each label "α" in any (e.g. random) order:
   1. Compute optimal α-expansion move (s-t graph cuts).
   2. Decline the move if there is no energy decrease.
3. Stop when no expansion move would decrease energy.

Slide credit: Yuri Boykov

Example: Stereo Vision



Original pair of "stereo" images

Depth map

Slide credit: Yuri Boykov

α-Expansion Moves

- In each α-expansion a given label "α" grabs space from other labels

initial solution

- -expansion
- -expansion
- -expansion
- -expansion
- -expansion
- -expansion
- -expansion

For each move, we choose the expansion that gives the largest decrease in the energy: $\Rightarrow$ binary optimization problem

Slide credit: Yuri Boykov

Topics of This Lecture

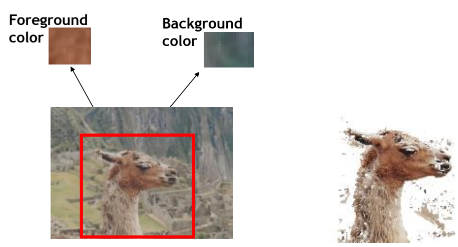
- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Extension to non-binary case
  - Applications

GraphCut Applications: "GrabCut"

- Interactive Image Segmentation [Boykov & Jolly, ICCV'01]
  - Rough region cues sufficient
  - Segmentation boundary can be extracted from edges
- Procedure
  - User marks foreground and background regions with a brush.
  - This is used to create an initial segmentation which can then be corrected by additional brush strokes.

Slide credit: Matthieu Bray

GrabCut: Data Model



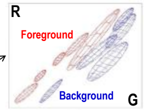
Global optimum of the energy

- Obtained from interactive user input
  - User marks foreground and background regions with a brush
  - Alternatively, user can specify a bounding box

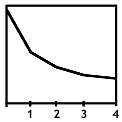
Slide credit: Carsten Rother

Iterated Graph Cuts

Result



Color model
(Mixture of Gaussians)



Energy after
each iteration

GrabCut: Example Results

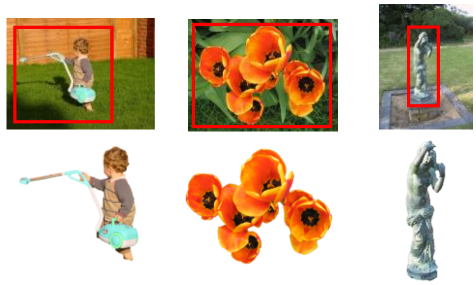


Image source: Carsten Rother

References and Further Reading

- A gentle introduction to Graph Cuts can be found in the following paper:
  - Y. Boykov, O. Veksler, Graph Cuts in Vision and Graphics: Theories and Applications. In *Handbook of Mathematical Models in Computer Vision*, edited by N. Paragios, Y. Chen and O. Faugeras, Springer, 2006.
- Try the Graph Cut implementation at
  http://pub.ist.ac.at/~vnk/software.html