Applied Computer Vision

David Vernon
Carnegie Mellon University Africa

vernon@cmu.edu
www.vernon.eu

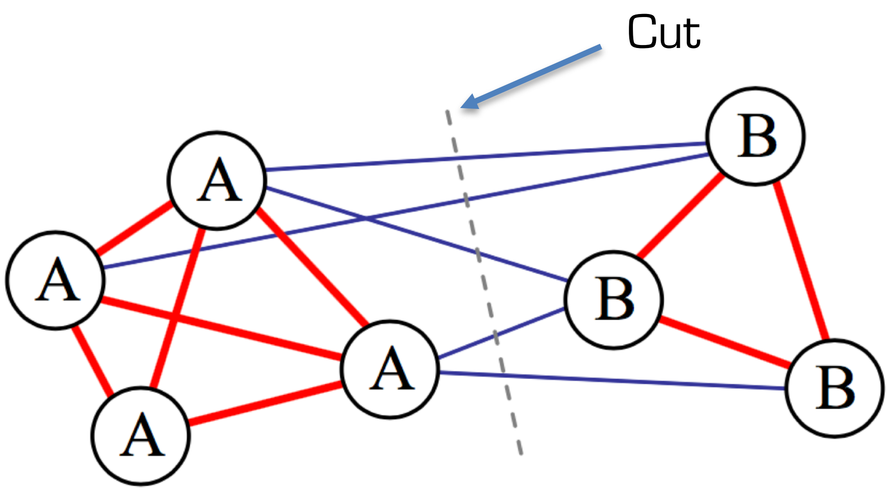
Lecture 10

Segmentation

Region-based Approaches:
Graph cuts and the GrabCut algorithm

Graph Cuts

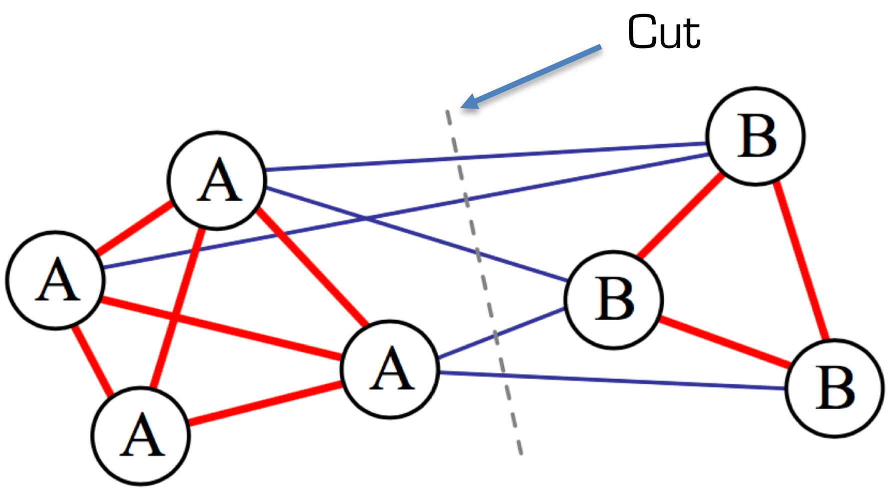
- Image as graph
  - Initially with one connected component
    (there is a path from any pixel to any other pixel)
- Segmentation as a process of finding a cut to separate the graph into two or more components



Source: R. Szeliski, *Computer Vision: Algorithms and Applications,* Springer, 2010.

Graph Cuts

- Identify affinities (similarity) between nearby pixels
- Separate groups that are connected by weak affinities



Source: R. Szeliski, *Computer Vision: Algorithms and Applications,* Springer, 2010.

Graph Cuts

- The cut between two groups A and B where the weights between two pixels (or regions) i and j measure their similarity

$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij}$$

- Using a minimum cut as a segmentation criterion usually doesn't result in reasonable clusters …

  The smallest cut usually just isolates a single pixel

Credit: R. Szeliski, *Computer Vision: Algorithms and Applications,* Springer, 2010.

Graph Cuts

A better measure: normalized cuts (Shi and Malik, 2000)

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

where

Association within a cluster: sum of all the weights ***within a cluster*** A

$$assoc(A,A) \;=\; \textstyle\sum_{i \in A, j \in A} w_{ij}$$

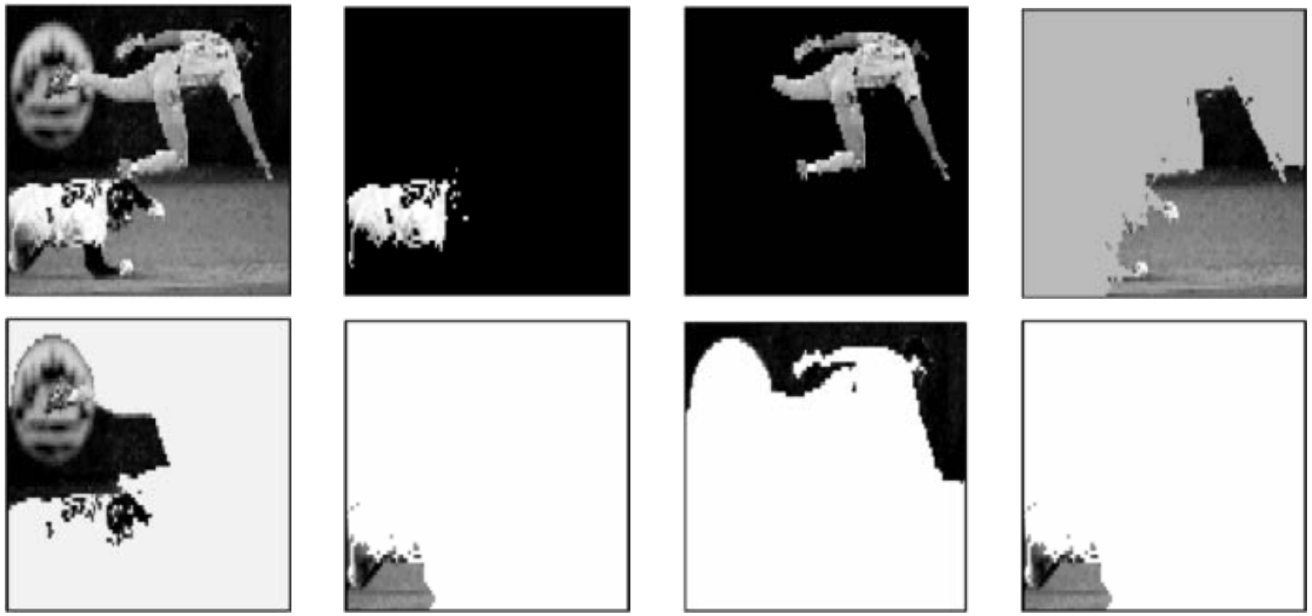
$$assoc(A,V) = assoc(A,A) + cut(A,B)$$

Sum of ***all*** the weights associated with nodes in A

Credit: R. Szeliski, *Computer Vision: Algorithms and Applications,* Springer, 2010.

Graph Cuts

- Computing the optimal normalized cut is an NP-complete problem
- Shi and Malik (2000) introduced an effective solution by transforming it to a generalized eigenvalue problem



Normalized cuts segmentation (Shi and Malik 2000) © 2000 IEEE

Credit: R. Szeliski, *Computer Vision: Algorithms and Applications,* Springer, 2010.

Graph Cuts

Segmentation can be formulated as an energy minimization problem, with two different approaches

- Regularization (i.e. variational) techniques (e.g. snakes)
- Techniques based on binary Markov random fields (e.g. graph cut techniques)

Exploit user interaction or external constraints

Graph Cuts

The energy function is encoded as a maximum flow problem and the minimum cut determines the region boundary

- The Min-cut / Max-flow problem

- s/t graph cut problem

- Solved using standard polynomial-time algorithms
  - Ford Fulkerson 1956
  - More recent optimized approaches: Boykov and Komolgorov 2004

Credit: R. Szeliski, *Computer Vision: Algorithms and Applications,* Springer, 2010.

Graph Cuts

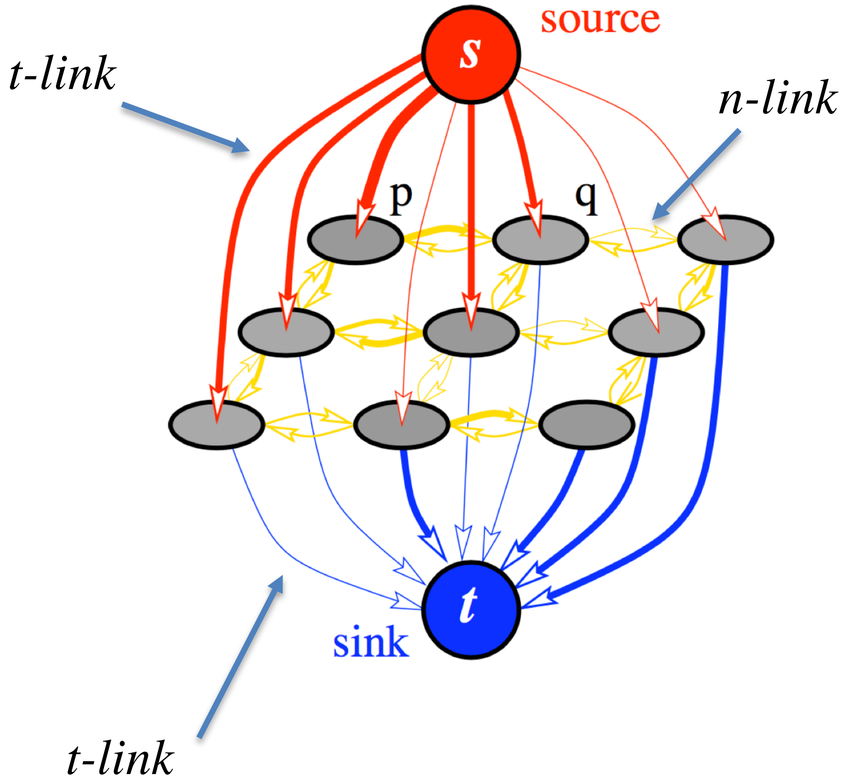
Basics

- Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be a graph
  which consists of a set of nodes $\mathcal{V}$ and
  a set of directed edges $\mathcal{E}$ that connect them
- The node set $\mathcal{V} = \{s, t\} \cup \mathcal{P}$ contains two special terminal nodes
  - source, s
  - sink, t

  and a set of non-terminal nodes $\mathcal{P}$
- Each edge is assigned some nonnegative weight or cost $w(p, q)$
- The weight of a directed edge (p, q) may differ from its reverse edge (q, p)

Source: Boykov and Veksler 2006

Graph Cuts



Edge weights / costs are reflected by thickness

Source: Boykov and Veksler 2006

Graph Cuts

Basics

- A set of all (directed) n-links is denoted $\mathcal{N}$

- The set of all graph edges $\mathcal{E}$ consists of
  n-links in $\mathcal{N}$ and
  t-links $\{(s,p),(p,t)\}$ for all non-terminal nodes $p \in \mathcal{P}$

Source: Boykov and Veksler 2006

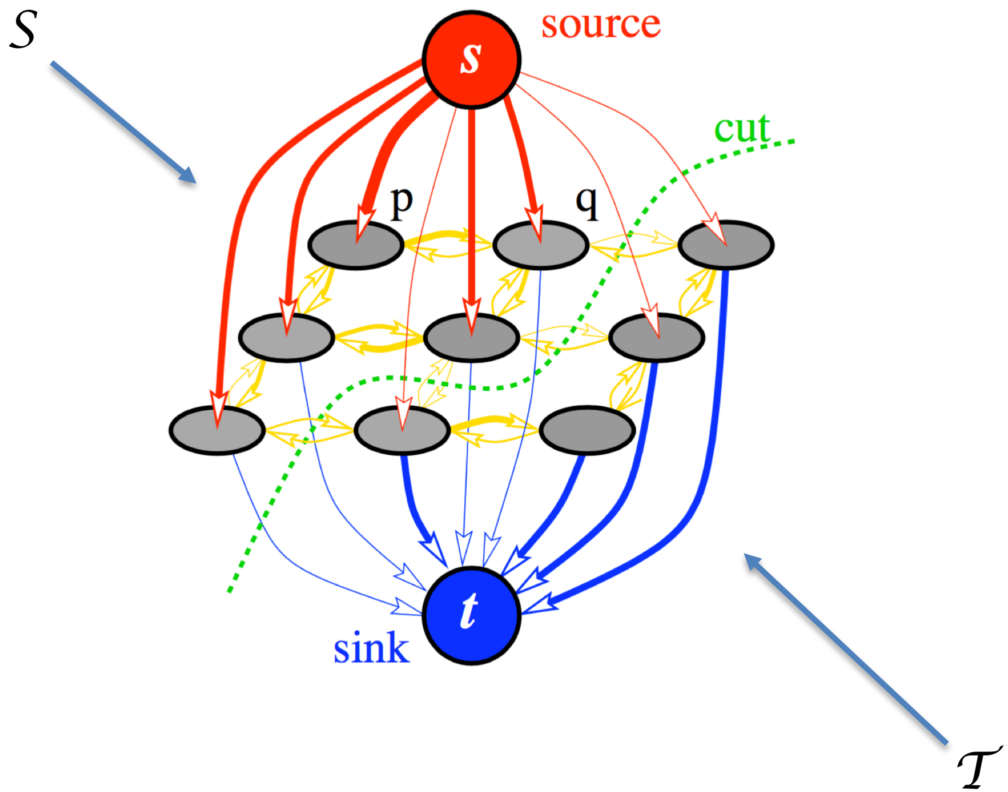
Graph Cuts

The Min-Cut and Max-Flow Problem

- An s/t cut C is a partitioning of the nodes in the graph into two disjoint subsets $\mathcal{S}$ and $\mathcal{T}$ such that the source s is in $\mathcal{S}$ and the sink t is in $\mathcal{T}$

Source: Boykov and Veksler 2006

Graph Cuts



Source: Boykov and Veksler 2006

Graph Cuts

The Min-Cut and Max-Flow Problem

- The cost of a cut $C = \{\mathcal{S}, \mathcal{T}\}$ is the sum of the costs/weights of “boundary” edges (p, q) such that $p \in \mathcal{S}$ and $q \in \mathcal{T}$

- If (p, q) is a boundary edge, we say cut C severs edge (p, q)

- The minimum cut problem is to find a cut that has the minimum cost among all cuts

Source: Boykov and Veksler 2006

Graph Cuts

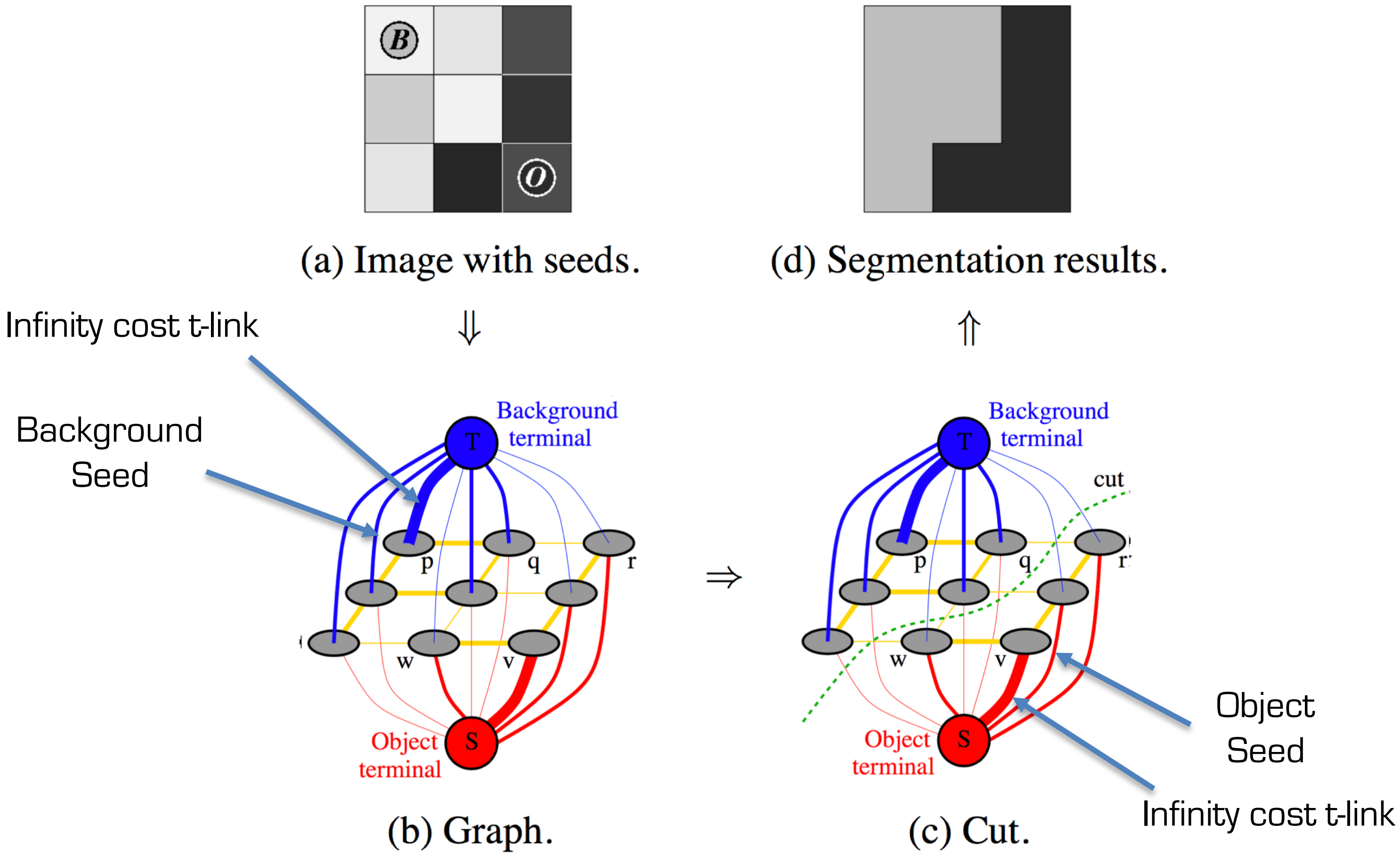
The Min-Cut and Max-Flow Problem

- A fundamental result in combinatorial optimization is that the minimum s/t cut problem can be solved by finding the maximum flow from the source s to the sink t
- Thus, the min-cut and max-flow problems are equivalent
- The maximum flow value is equal to the cost of the minimum cut

Source: Boykov and Veksler 2006

Graph Cuts

Using seeds



Edge weights / costs are reflected by thickness

Source: Boykov and Funka-Lea 2006

Graph Cuts

Example

- The image is divided into one “object” region and one “background” region

- In general, cuts can generate binary segmentation with arbitrary topological properties

    - Many isolated blobs
    - Many holes

Graph Cuts

Goal

- Compute the best cut that would give an “optimal” segmentation
- In combinatorial optimization, the cost of a cut is defined as the sum of the costs of the edges it severs
- Severed n-links are located at the segmentation boundary
  - their total cost represents the cost of the segmentation boundary
- Severed t-links can represent the regional properties of segments
- Thus, a minimum cost cut may correspond to a segmentation with desirable balance of boundary and regional properties
- Infinity cost t-links make it possible to impose hard constraints on segments

Graph Cuts

Segmentation Energy

- Consider an arbitrary set of pixels $\mathcal{P}$ and some neighbourhood system represented by a set $\mathcal{N}$ of all (unordered) pairs $\{p, q\}$ of neighbourhood elements in $\mathcal{P}$

- Let $A = (A_1, \ldots, A_p, \ldots, A_{|\mathcal{P}|})$ be a binary vector whose components A_p specify assignments to pixels p in $\mathcal{P}$

- Each A_p can be either "obj" or "bkg" (object or background)

- Vector A defines a segmentation

Graph Cuts

Segmentation Energy

Soft constraints on boundary & region properties of A are given by cost function $E(A)$:

$\lambda \geq 0$ specifies relative importance of regional properties

$$E(A) = \lambda \cdot R(A) + B(A)$$

where

$$R(A) = \sum_{p \in \mathcal{P}} R_p(A_p)$$

Regional term: assumes the individual penalties for assigning pixel p to object and background are given: R_p("obj") and R_p("bkg")

$$B(A) = \sum_{\{p,q\} \in \mathcal{N}} B_{p,q} \cdot \delta_{A_p \neq A_q}$$

Boundary term: $B_{p,q} \geq 0$ is a penalty for discontinuity between p and q

and

$$\delta_{A_p \neq A_q} = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{if } A_p = A_q \end{cases}$$

$B_{p,q}$ cost is zero if p and q are part of the same region

Graph Cuts

Segmentation Energy

Regional term $R_p(\cdot)$ might reflect how the intensity of pixel p fits into given intensity models (e.g. histograms) of the object and background:

$$R_p(\text{"obj"}) = -\ln \Pr(I_p|\text{"obj"})$$
$$R_p(\text{"bkg"}) = -\ln \Pr(I_p|\text{"bkg"})$$

Minus the log of the probability that image intensity of p occurs, given that it is an object pixel

Minus the log of the probability that image intensity of p occurs, given that it is an background pixel

Note: the use of negative log-likelihoods is motivated by the Maximum A Posterior estimation of a Markov Random Field (MAP-MRF) formulation in the original approach by [Greig et al. 1989]; see Szeliszki 2010, Section 3.7.2

Graph Cuts

Segmentation Energy

Boundary term $B(A)$ is the penalty for a discontinuity between p and q

$B_{p,q}$ is large when pixels p and q are similar (e.g. in intensity)
$B_{p,q}$ is small when pixels p and q are very different

$B_{p,q}$ can also decrease as a function of distance between p and q

Costs $B_{p,q}$ may be based on local intensity gradient, Laplacian zero-crossing, gradient direction, or other criteria.

$$B_{p,q} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p, q)}$$

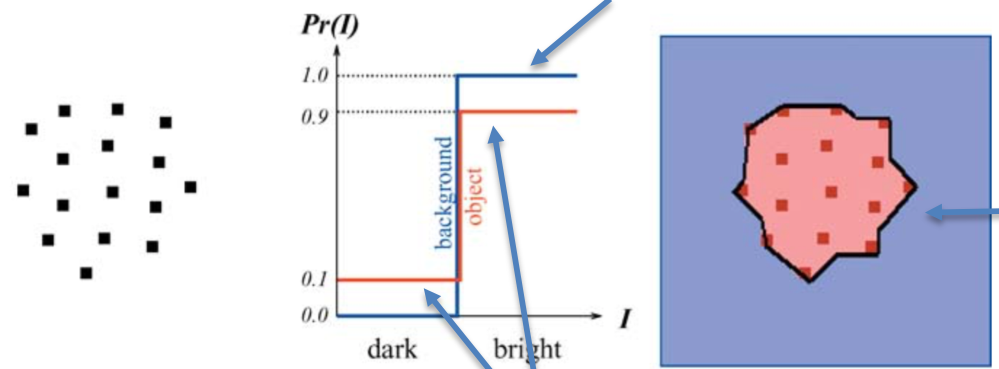
Example function

Big penalty/cost when $|I_p - I_q| < \sigma$
Small penalty/cost when $|I_p - I_q| > \sigma$

Graph Cuts

Synthetic Example

background histogram allows for only bright values



(a) Original image (b) Intensity histograms (c) Optimal segmentation

Optimal segmentation (achieved using graph cuts) finds a balance between the regional and the boundary energy terms

Object histogram allows for dark intensities of dots and bright space in between

Boundary Term:

$$B_{p,q} = \begin{cases} 1 & \text{if } I_p = I_q \\ 0.2 & \text{if } I_p \neq I_q \end{cases}$$

Regional Term: Assume the *a priori* known intensity histograms in fig. (b) above

Using these histograms in the negative log-likelihood formulae we get the regional penalties $R_p(Ap)$ for pixels with different intensities

I_p	R_p("obj")	R_p("bkg")
dark	2.3	$+\infty$
bright	0.1	0

Source: Boykov and Funka-Lea 2006

Graph Cuts

Synthetic Example

Assign weights to all

n-links

and t-links

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	$p \in \mathcal{B}$

Maximum of
sum of boundary costs
for all neighbours of p

where

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

Source: Boykov and Funka-Lea 2006

Graph Cuts

Synthetic Example

- Find the minimum cost cut C_{min} using a standard algorithm
  - e.g. based on the Ford Fulkerson method
- Determine the segmentation $A(C) = (A_1, ..., A_p, ..., A_{|\mathcal{P}|})$

$$A_p(C) = \begin{cases} \text{“obj”}, & \text{if } \{p, T\} \in C \\ \text{“bkg”}, & \text{if } \{p, S\} \in C \end{cases}$$

Graph Cuts

The Min-Cut and Max-Flow Problem

- There are many standard polynomial time algorithms for min-cut/max-flow
- The Ford Fulkerson method is perhaps the best known
- For grid graphs, such as we have in images, Boykov and Komolgorov developed a fast algorithm which has a linear time complexity
  - While the Boykov and Komolgorov algorithm is very efficient (few seconds on typical images) it is far from real-time

Source: Boykov and Veksler 2006

Graph Cuts

Hard Constraints

- Objects may not have sufficiently distinct regional properties to impose adequate constraints on the segmentation

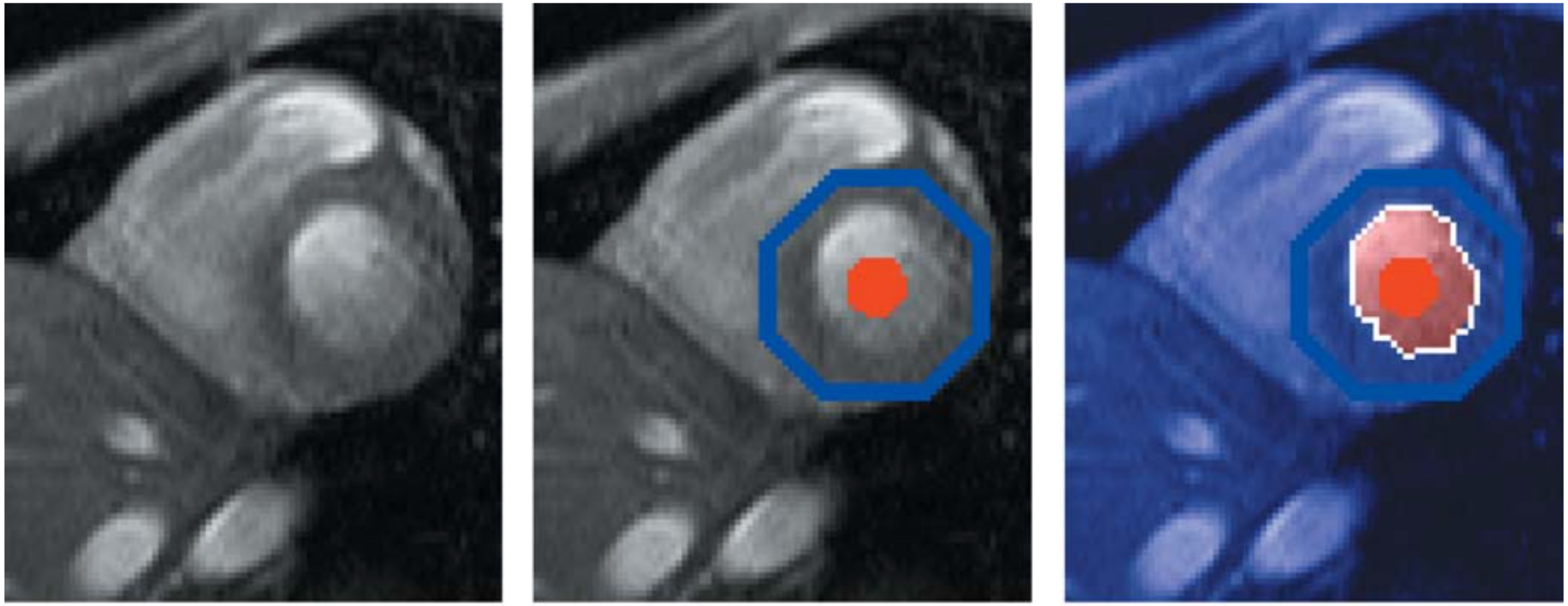
- We constrain the search space of possible solutions

- This is done by providing hard constraints:
  - identifying a set of pixels (seeds) that belongs to the object $\mathcal{O}$
  - identifying a set of pixels (seeds) that belong to the background $\mathcal{B}$

$$\forall p \in \mathcal{O} : \quad A_p = \text{“obj”}$$
$$\forall p \in \mathcal{B} : \quad A_p = \text{“bkg”}$$

Graph Cuts

Hard Constraints



(a) Original image (b) Initialization (c) Segmentation

$$\forall p \in \mathcal{O}: \quad A_p = \text{“obj”}$$
$$\forall p \in \mathcal{B}: \quad A_p = \text{“bkg”}$$

Graph Cuts

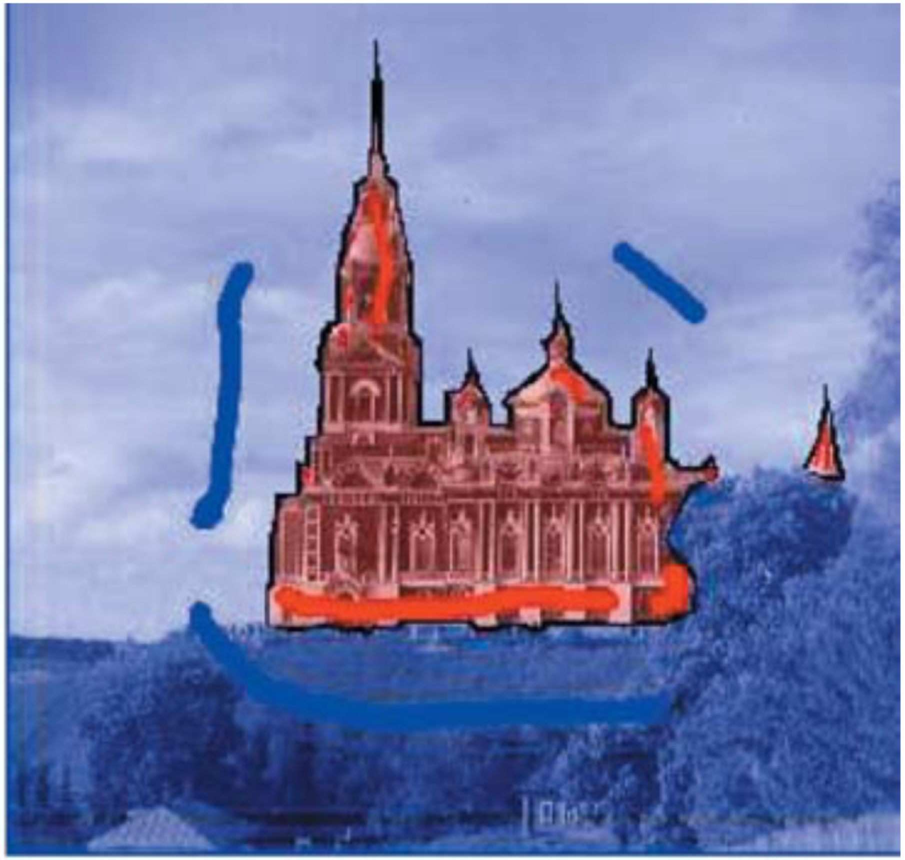
Hard Constraints

- Intensities of pixels marked as seeds can also be used to learn the histograms for “object” and “background” intensity distributions:

$$\Pr(I_p|\text{“obj”})$$
$$\Pr(I_p|\text{“bkg”})$$

Graph Cuts

(a) A woman from a village

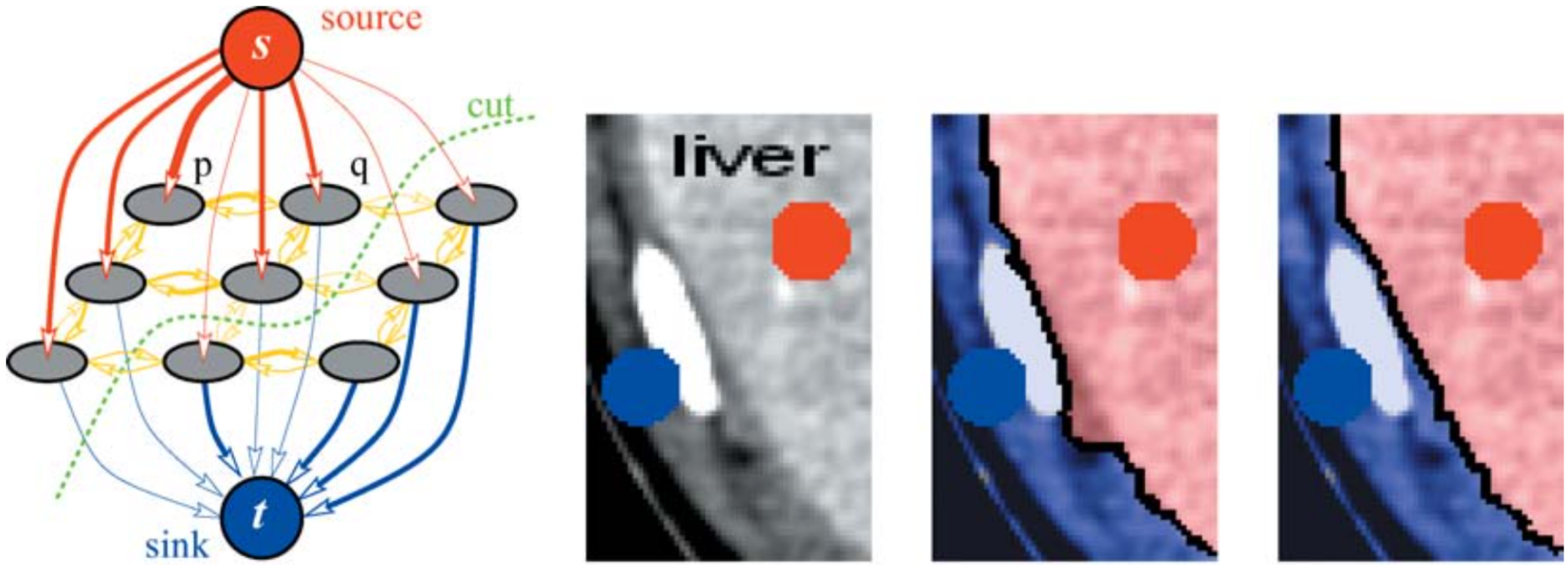


(b) A church in Mozhaisk (near Moscow)

Source: Boykov and Funka-Lea 2006

Graph Cuts

Graph cuts with directed graphs



(a) directed graph (b) image (c) undir. result (d) dir. result

Source: Boykov and Funka-Lea 2006

Graph Cuts

Graph cuts with directed graphs

- The weights of the directed edges (p, q) can depend on the sign of the intensity difference $(I_p\text{-}I_q)$
- Undirected edges depend only on the absolute value $|I_p\text{-}I_q|$
- See [Boykov and Funka-Lea 2006] for details

Graph Cuts

GrabCut

- Extension of the basic Boykov & Jolly (2001) technique by Rother, Komolgorov, and Blake (2004)
- Iteratively re-estimates the region statistics
- Region statistics are modelled as a mixture of Gaussians in colour space

Graph Cuts

GrabCut

This allows the approach to operate with minimal user input

- Single bounding box
- The background colour model
  - initialized from a strip of pixels around the box outline
- The foreground colour model
  - initialized from the interior pixels
  - quickly converges to a better estimate of the object

Graph Cuts

GrabCut

Source: Rother, Komolgorov, Blake 2004

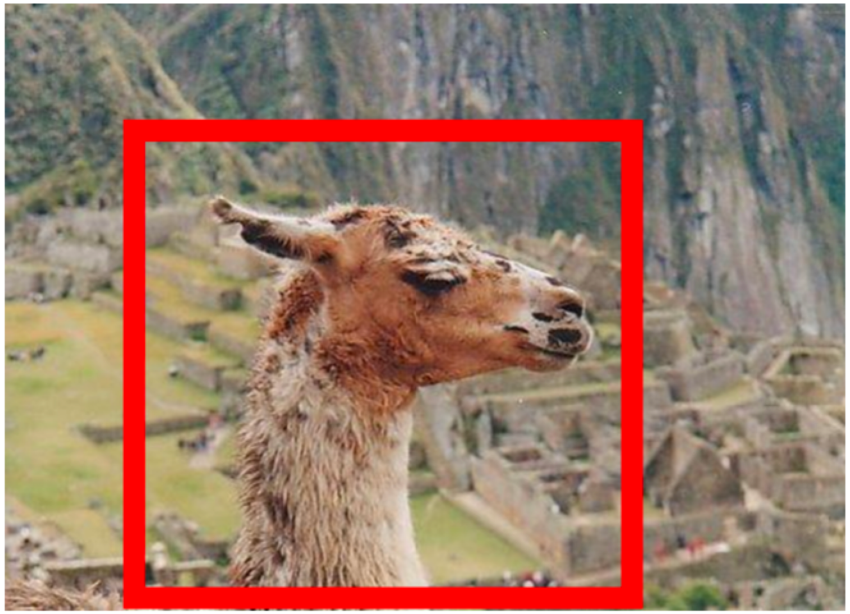
Graph Cuts

GrabCut

Source: Rother, Komolgorov, Blake 2004

Graph Cuts

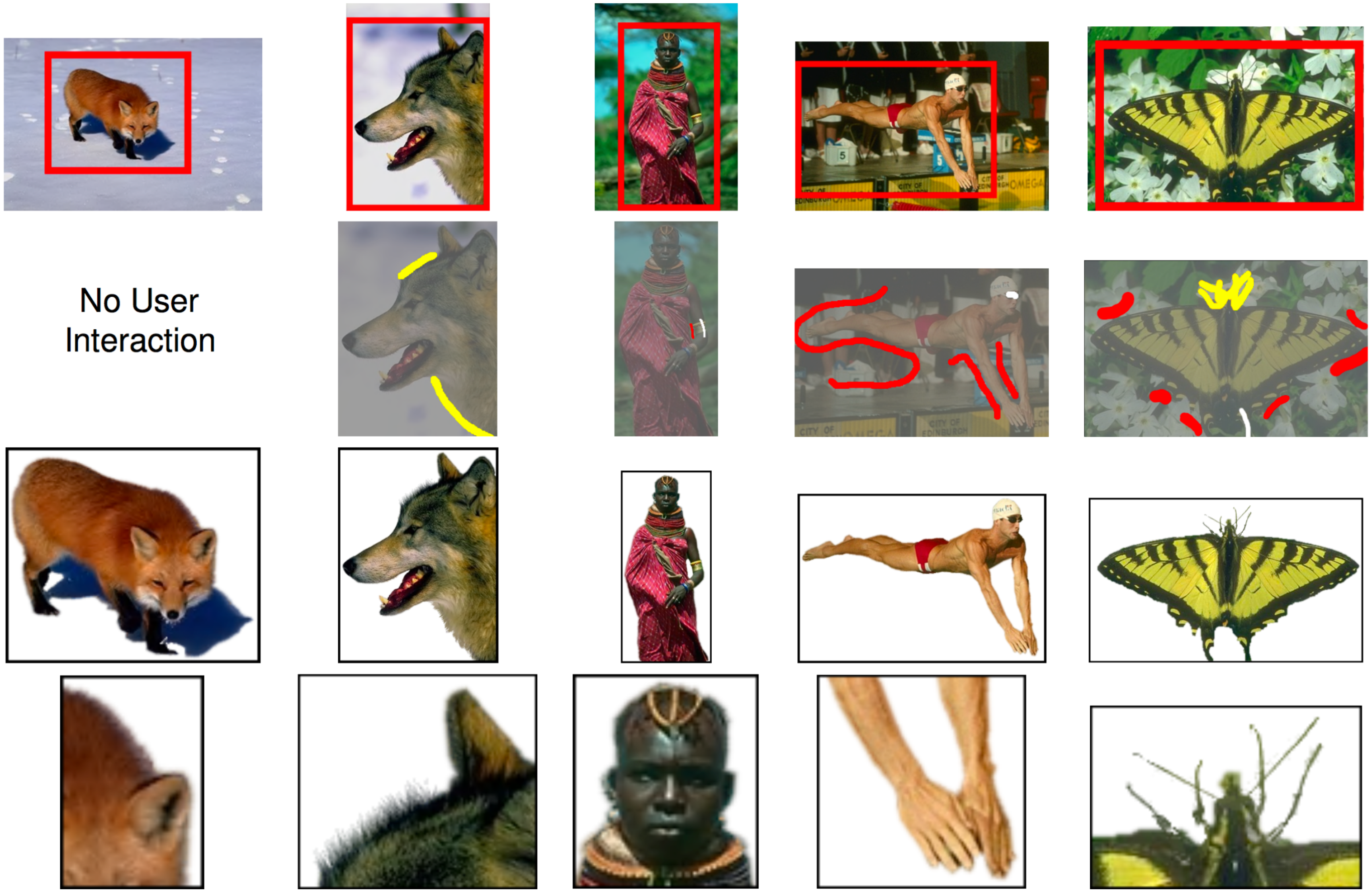
GrabCut



Source: Rother, Komolgorov, Blake 2004

Graph Cuts

GrabCut



Source: Rother, Komolgorov, Blake 2004

Demos

The following code is taken from the grabCut project in the lectures directory of the ACV repository

See:

```
grabCut.h
grabCutImplementation.cpp
grabCutApplication.cpp
```

```
/*
 * function grabCut
 * Trackbar callback - number of iterations user input
*/

void performGrabCut(int, void*) {

   extern Mat   inputImage;
   extern int   numberOfIterations;
   extern int   number_of_control_points;
   extern char* grabcut_window_name;
   Mat result;          // segmentation result
   Mat bgModel,fgModel; // the models (hard constraints)

   if (numberOfIterations < 1)  // the trackbar has a lower value of 0 which is invalid
      numberOfIterations = 1;

   /* get two control points (top left and bottom right) and rectangle */
   do {
      waitKey(30);
   } while (number_of_control_points < 2);

    /* GrabCut segmentation                                                              */
    /* see: http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#grabcut */
    grabCut(inputImage,         // input image
            result,             // segmentation result (4 values); can also be used as an input mask providing constraints
            rect,               // rectangle containing foreground
            bgModel,fgModel,    // for internal use ... allows continuation of iterative solution on subsequent calls
            numberOfIterations, // number of iterations
            GC_INIT_WITH_RECT); // use rectangle

    /* Get the pixels marked as likely foreground */
    compare(result,GC_PR_FGD,result,CMP_EQ);

    /* Generate output image */
    Mat foreground(inputImage.size(),CV_8UC3,cv::Scalar(255,255,255));
    inputImage.copyTo(foreground,result); // use result to mask out the background pixels

    imshow(grabcut_window_name, foreground);
}
```

Reading

R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

Section 5.4 Normalized cuts

Section 5.5 Graph cuts and energy-based methods

Section 3.7 Global optimization

Section 3.7.1 Regularization

Section 3.7.2 Markov random fields

Reading

s/t graph cuts and overview of energy-based graph cut methods

Boykov, Y. and Jolly, M.-P. 2001. "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images", Proc. International Conference on Computer Vision, Vol. I, pp. 105-112.

Boykov, Y. and Funka-Lea, G. 2006. "Graph Cuts and Efficient N-D Image Segmentation", International Journal of Computer Vision 70(2), 109–131.

Summary of min-cut approaches in computer vision

Boykov, Y. and Veksler, O. 2006. "Graph Cuts in Vision and Graphics: Theories and Applications", in Handbook of Mathematical Models of Computer Vision, Paragios, N., Chen, Y., Faugeras, O. D. (eds.), Springer, pp. 79-96.

GrabCut

Rother, C., Kolmogorov, V., and Blake, A. 2004. "GrabCut – Interactive Foreground Extraction using Iterated Graph Cuts, ACM Transactions on Graphics.