COSC579: Image Segmentation

Jeremy Bolton, PhD
Assistant Teaching Professor

GEORGETOWN UNIVERSITY

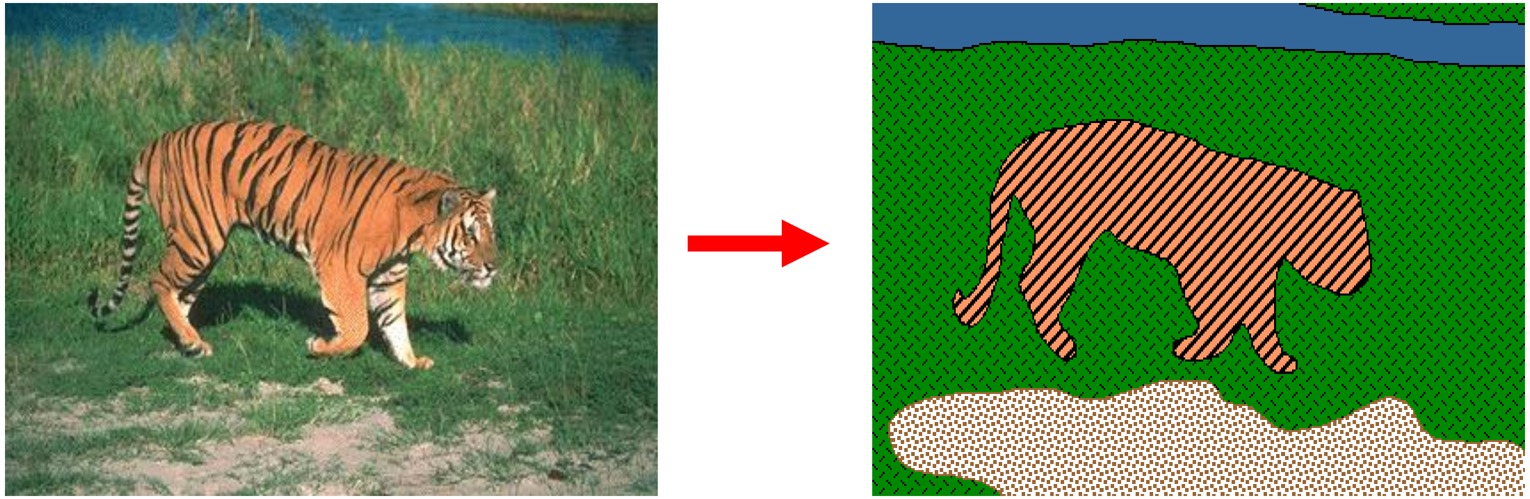
Outline

I. Motivation
   I. Spectral, Feature, and Spatial Information
      I. Basic histogram vs nearness
   II. Split and Merge Approaches
      I. Watershed
      II. Agglomerative and Divisive Clustering
   III. Graph discussion
      I. Shortest Paths: Scissors
      II. Min Cuts
   IV. Snakes
      I. Find object boundaries using contours
      II. Energy Minimization using Variational Methods

Segmentation

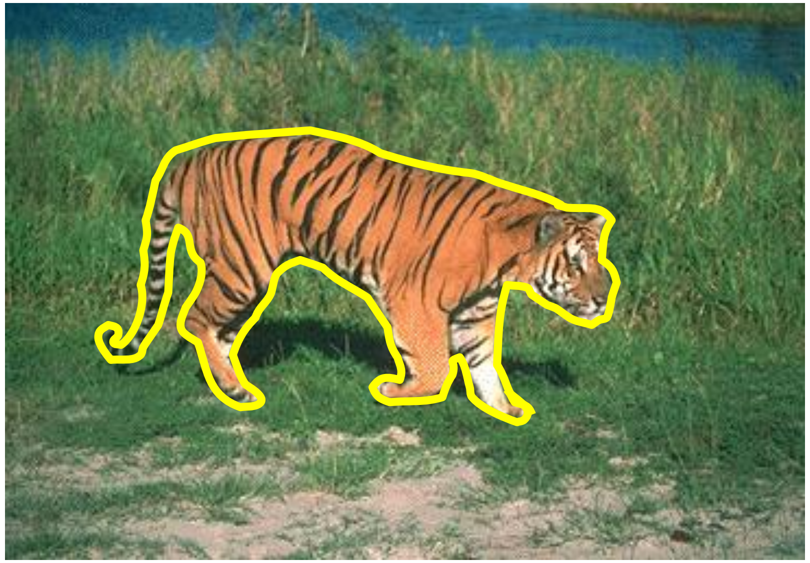
- Today's Readings
  - Szelinski CH 5

From images to objects



- What Defines an Object?
  - Subjective problem, but has been well-studied
  - Gestalt Laws seek to formalize this
    - proximity, similarity, continuation, closure
    - see [notes] by Steve Joordens, U. Toronto

Extracting objects



- How could this be done?

Segmentation

- Segmentation in general is a tough problem.
  - Ill-posed problem
  - Concept of “segments” is subjective
- There are many approaches to segmentation
- There is no universal approach that works well in all scenarios

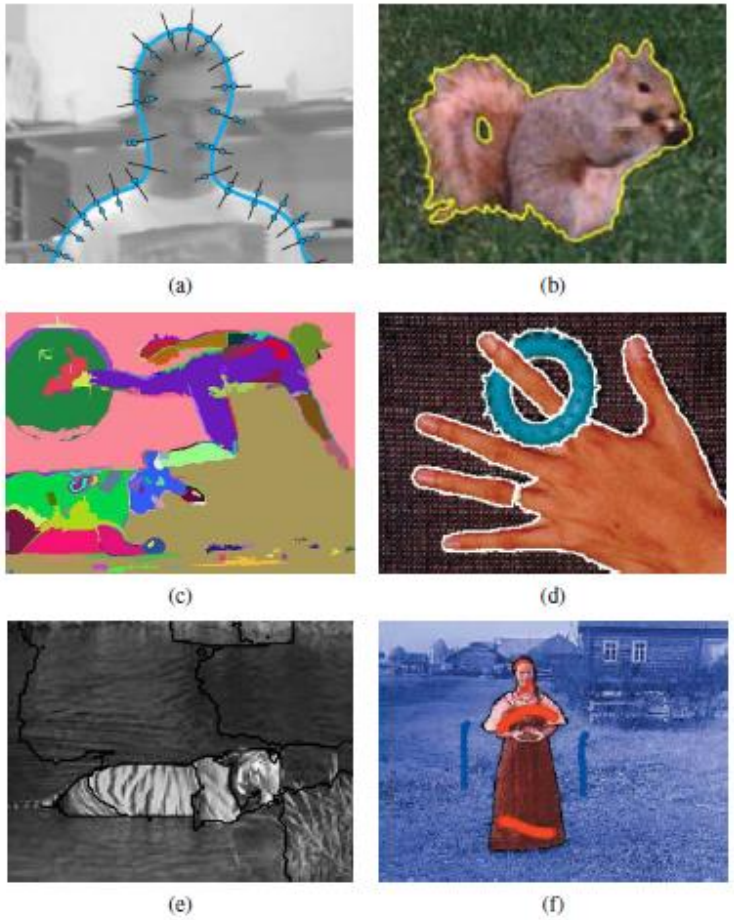


Image Segmentation

- The segmentation problem is the problem of identifying different segments in an image

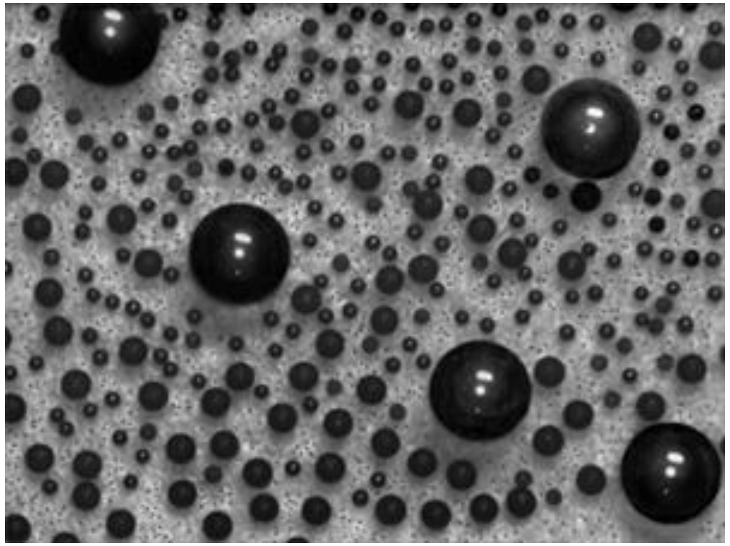
- Based on a number of factors (which will vary based on specific application)
  - Color / intensity
  - Distance / proximity
  - Features (local)

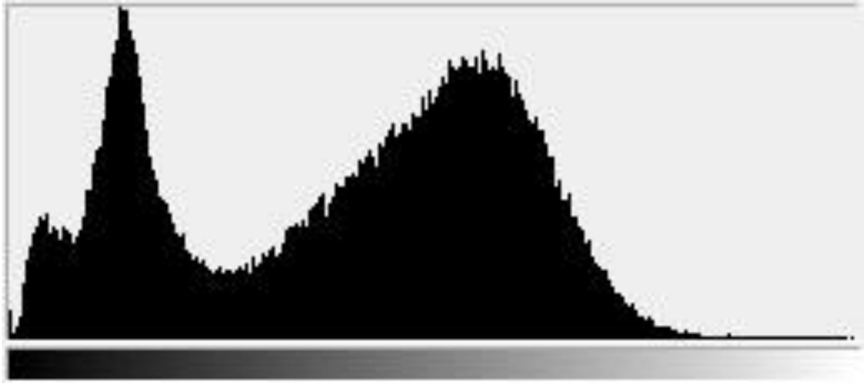
Similarity

- Based only on color or intensity
  - no image coordinate information
  - very basic
- Based on Features
  - May provide contextual information (or other based on feature computed)
- Common Approaches
  - Modes in Histograms (intensity or some color space)
    - Quantizing / binning required
  - Metrics in a Vector Space (multi-dimensional)

Histogram-based segmentation

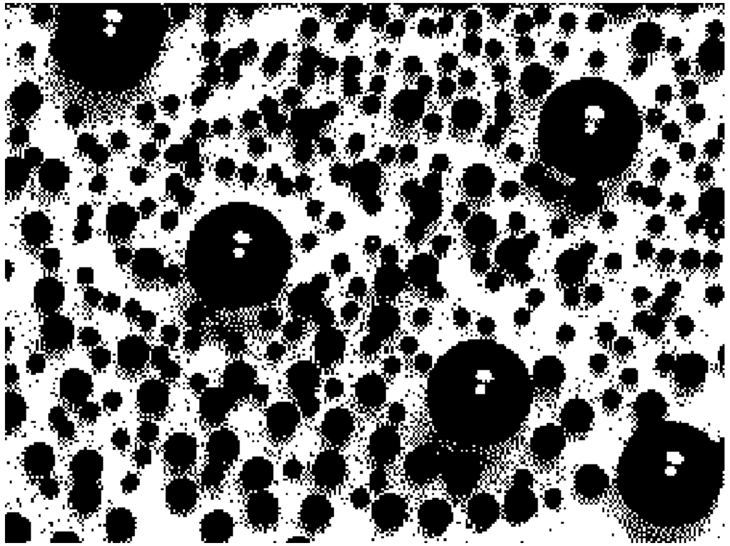
- Goal
  - Break the image into K regions (segments)
  - Solve this by reducing the number of colors to K and mapping each pixel to the closest color

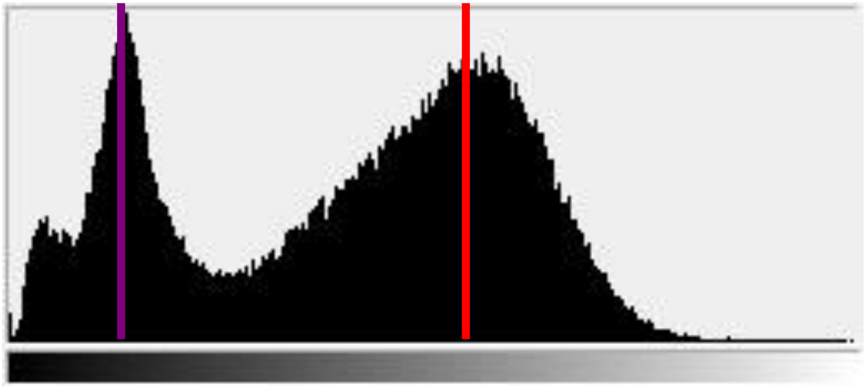




Histogram-based segmentation

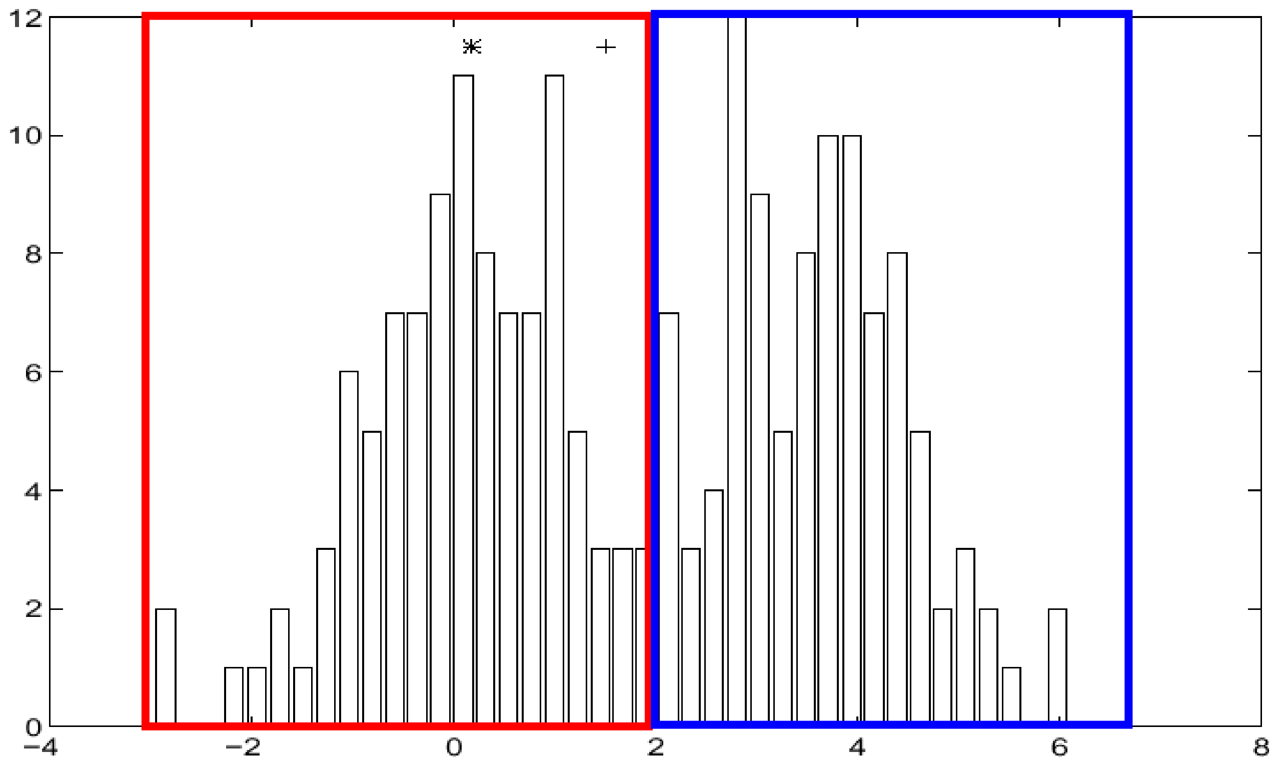
- Goal
  - Break the image into K regions (segments)
  - Solve this by reducing the number of colors to K and mapping each pixel to the "closest" color





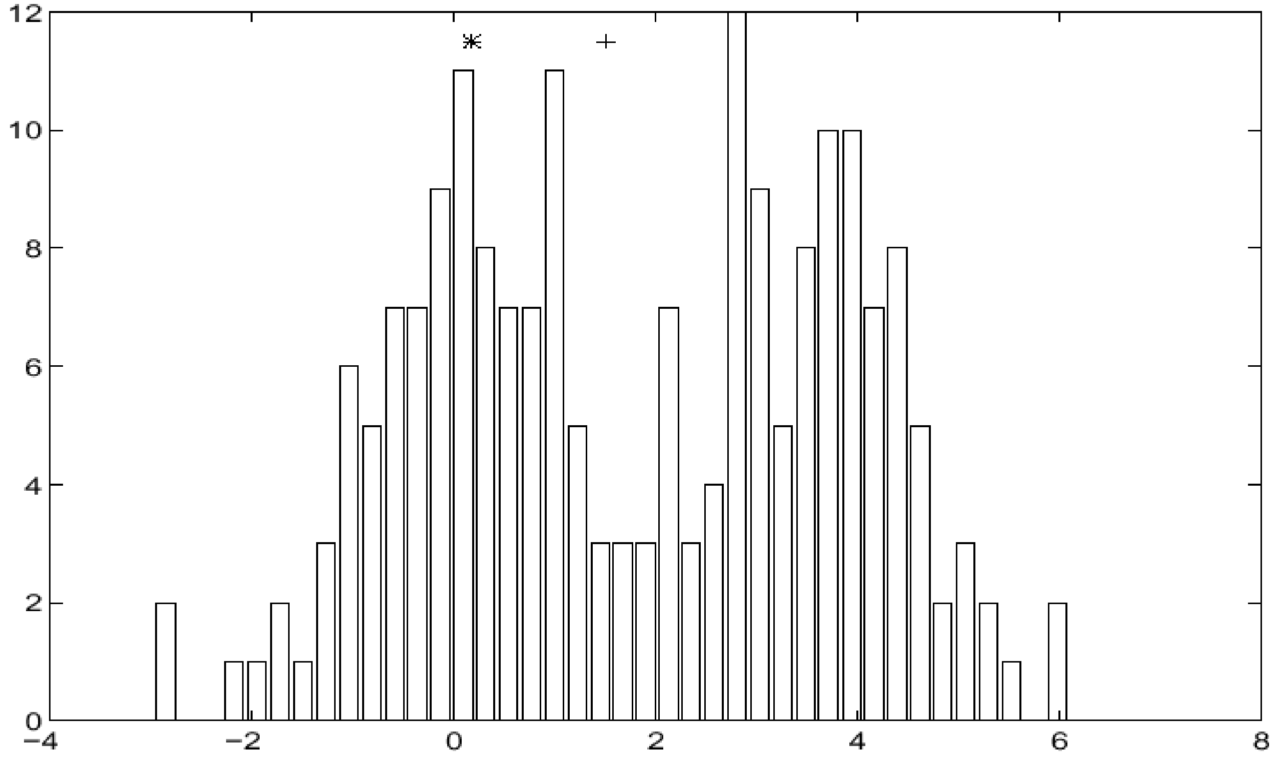
Here's what it looks like if we use K = two colors

Finding Modes in a Histogram



- How Many Modes Are There?
  - Easy to see (sometimes), hard to compute

Mean Shift [Comaniciu & Meer]

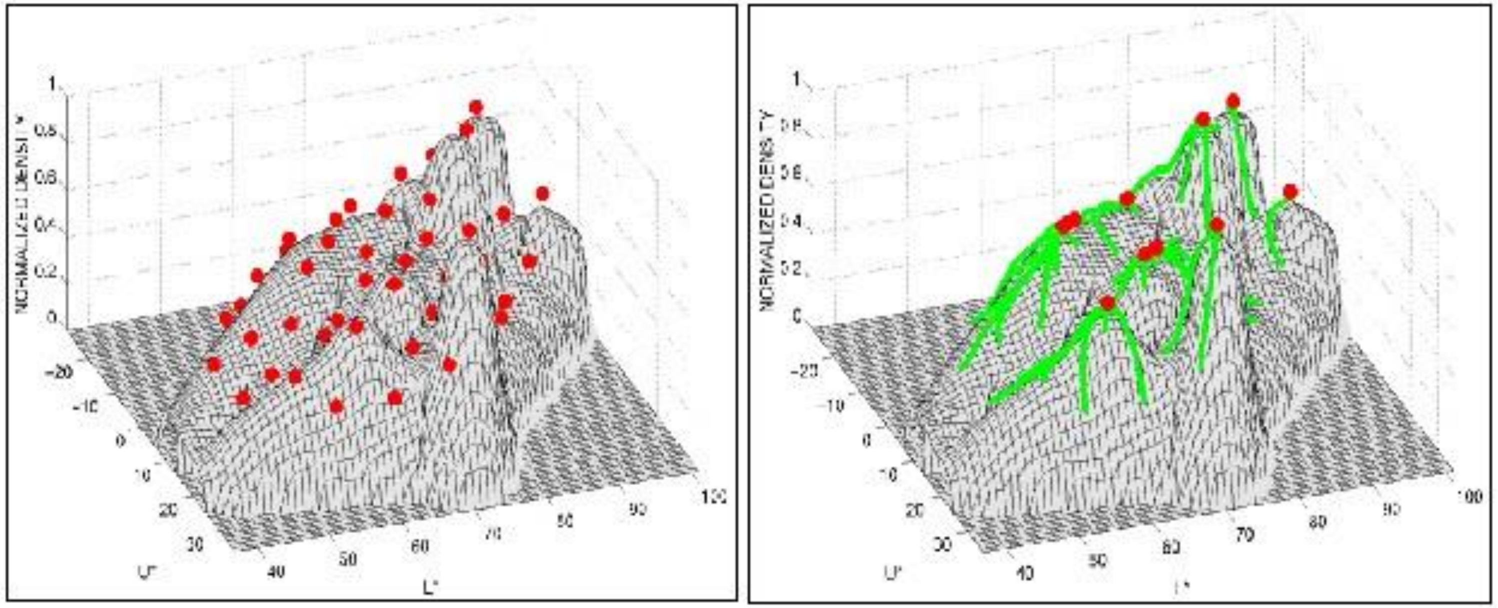


- Iterative Mode Search
  1. Initialize random seed, and window W
  2. Calculate center of gravity (the "mean") of W: $\sum_{x \in W} xH(x)$
  3. Translate the search window to the mean
  4. Repeat Step 2 until convergence

GEORGETOWN UNIVERSITY

Mean-Shift

- Approach
  - Initialize a window around each point
  - See where it shifts—this determines which segment it's in
  - Multiple points will shift to the same segment

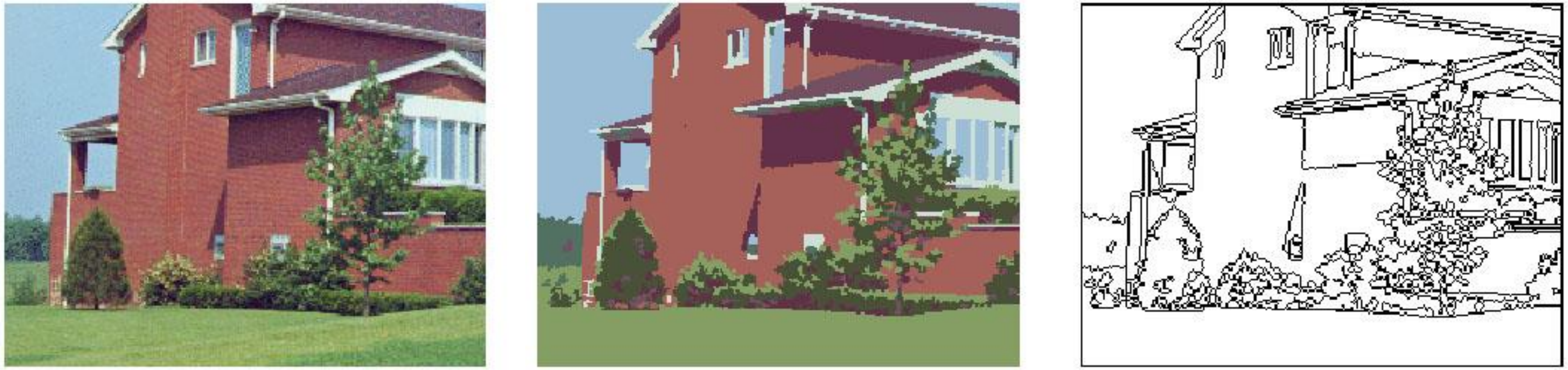


Mean shift trajectories

GEORGETOWN UNIVERSITY

Mean-shift for image segmentation

- Useful to take into **account spatial information**
  - instead of (R, G, B), run in (R, G, B, x, y) space
  - D. Comaniciu, P. Meer, Mean shift analysis and applications, *7th International Conference on Computer Vision*, Kerkyra, Greece, September 1999, 1197-1203.
    - http://comaniciu.net/Papers/MsAnalysis.pdf



More Examples: http://www.caip.rutgers.edu/~comanici/segm_images.html

GEORGETOWN UNIVERSITY

Watershed Approach

- Interpret image as a landscape where intensity represents height.
- Approach: flood the landscape and see where the water flows
  - All pixels that flow to the same minima (water basin) are in a segment
  - To adjust granularity of segments, add or drain water.

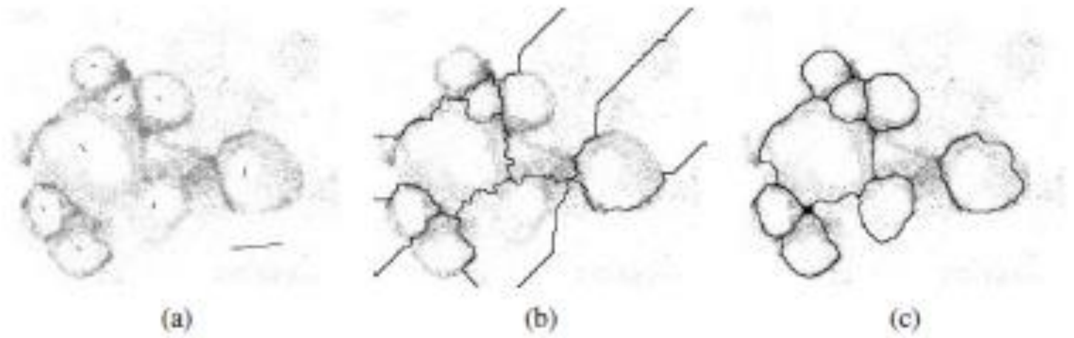
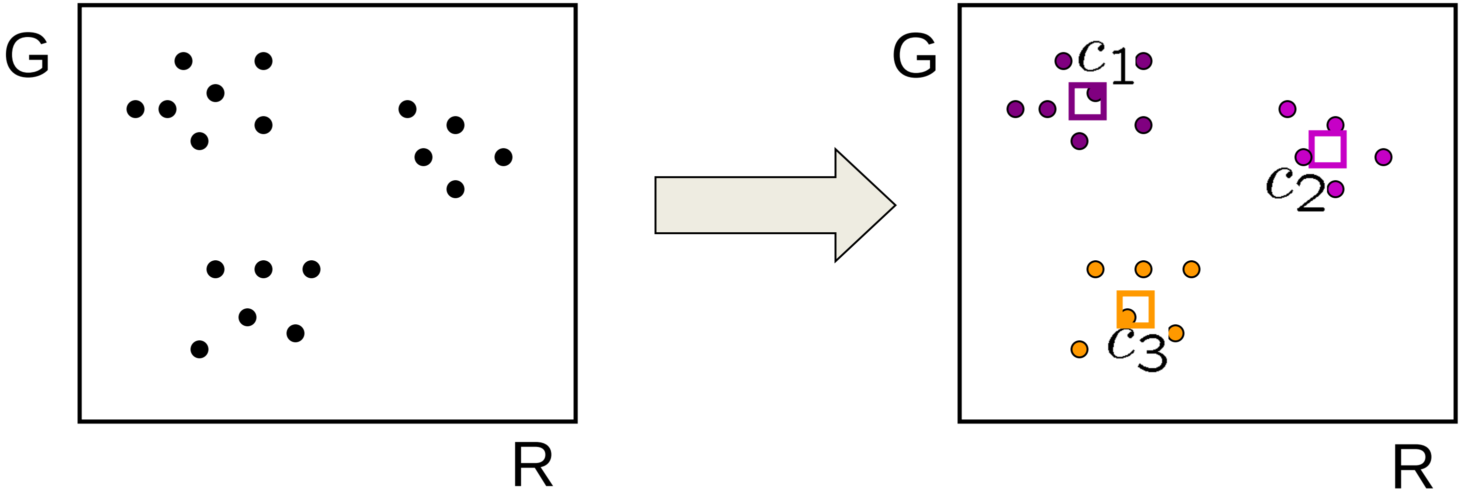


Figure 5.13 Locally constrained watershed segmentation (Beare 2006) © 2006 IEEE: (a) original confocal microscopy image with marked seeds (line segments); (b) standard watershed segmentation; (c) locally constrained watershed segmentation.

GEORGETOWN UNIVERSITY

Segmentation as Clustering

- How to choose the representative colors?
  - This is a clustering problem!



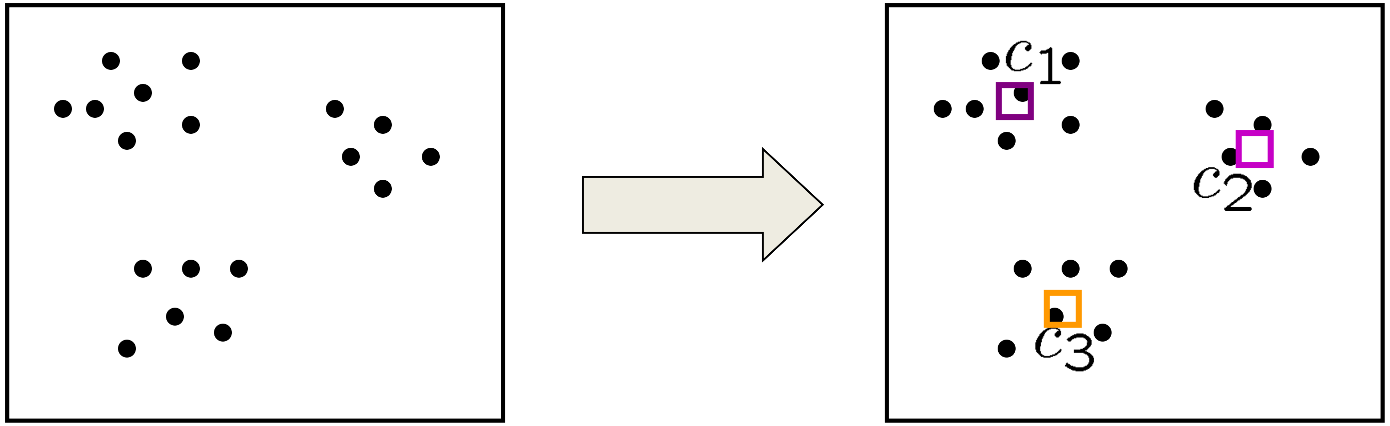
Objective

- Each point should be as close as possible to a cluster center
  - Minimize sum squared distance of each point to closest center

$$\sum_{\text{clusters } i} \quad \sum_{\text{points p in cluster } i} ||p - c_i||^2$$

Break it down into subproblems

- Suppose I tell you the cluster centers c_i
  - Q: how to determine which points to associate with each c_i?
  - A: for each point p, choose closest c_i



Suppose I tell you the points in each cluster

- Q: how to determine the cluster centers?
- A: choose c_i to be the mean of all points in the cluster

Split and Merge

- Simple approach for using spatial and intensity.
  1. Threshold
  2. Identify connected components

- Performing these operations once in sequence rarely works; however repeatedly performing these operations improves results.
  - Terrain - based model: Watershed
  - Cluster-based models
  - Graph-based

Divisive Clustering (Region Splitting)

- (Often) Based on Histogram
  - Starting at a high threshold and continuing to decrease, find a “good” threshold for the histogram that best separates the large peaks.

Agglomerative-clustering (Region Merging)

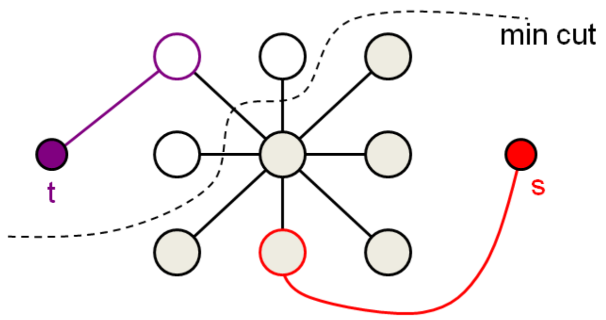
- Approach
  1. Seed (many) coordinates in the image as cluster centers
  2. If a neighboring pixel is “similar enough” then add it to the cluster
  3. If a neighboring pixel is to be added to a cluster, but is already a member of another cluster, then merge clusters
  4. Repeat until some stopping criterion is met

Contours as Segment Edges

- Segmentation as identifying the **boundaries** of segments
- Segmentation: intrinsic balance between spectral (or feature) and spatial similarities
  - Pose as energy minimization with two terms: similarity and nearness

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

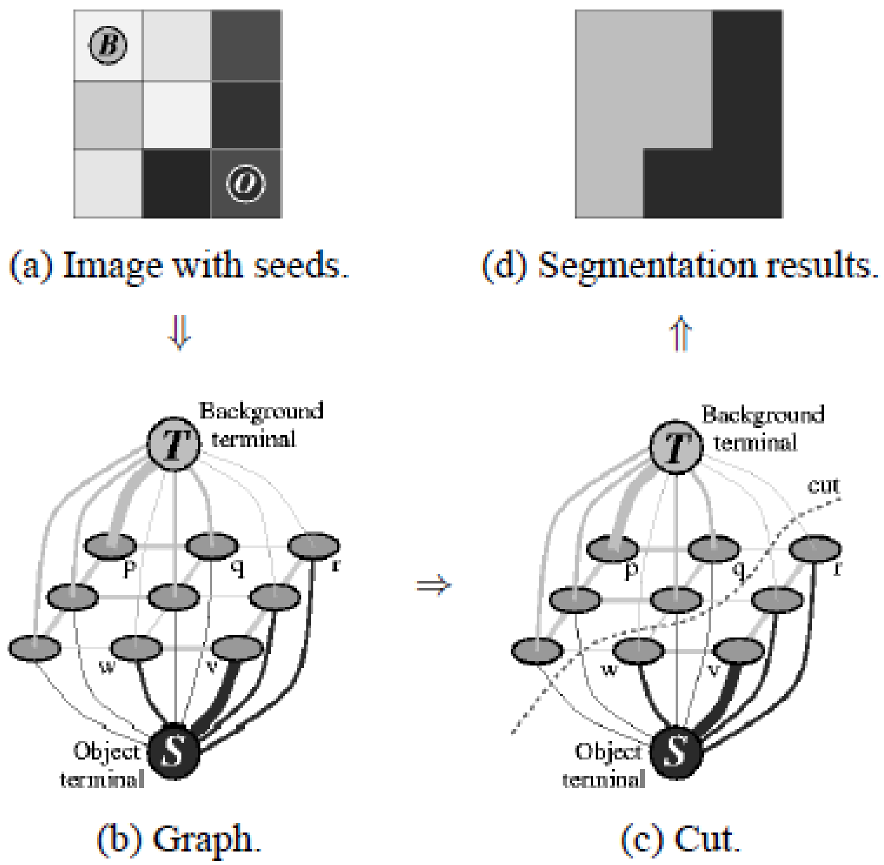
- Interpret image as a graph
  - Pixels are nodes
    - Have observed spectra, features, etc
  - Neighboring pixels have connecting edges
    - Nearness measured in image coordinates
  - Approaches are often interactive:
    - How are representatives of each cluster determined?
    - How are terminals defined?!?



GEORGETOWN UNIVERSITY

Interactive "Seeding" of Graph

- Problem with Segmentation: segment representative is unknown.
  - Solution: Interactively choose "Seed"



Weights on the Graph

- Links
  - N-link: link to image neighbors

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

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{otherwise.} \end{cases}$$

$B_{\{p,q\}}$ is large when pixels p and q are similar

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

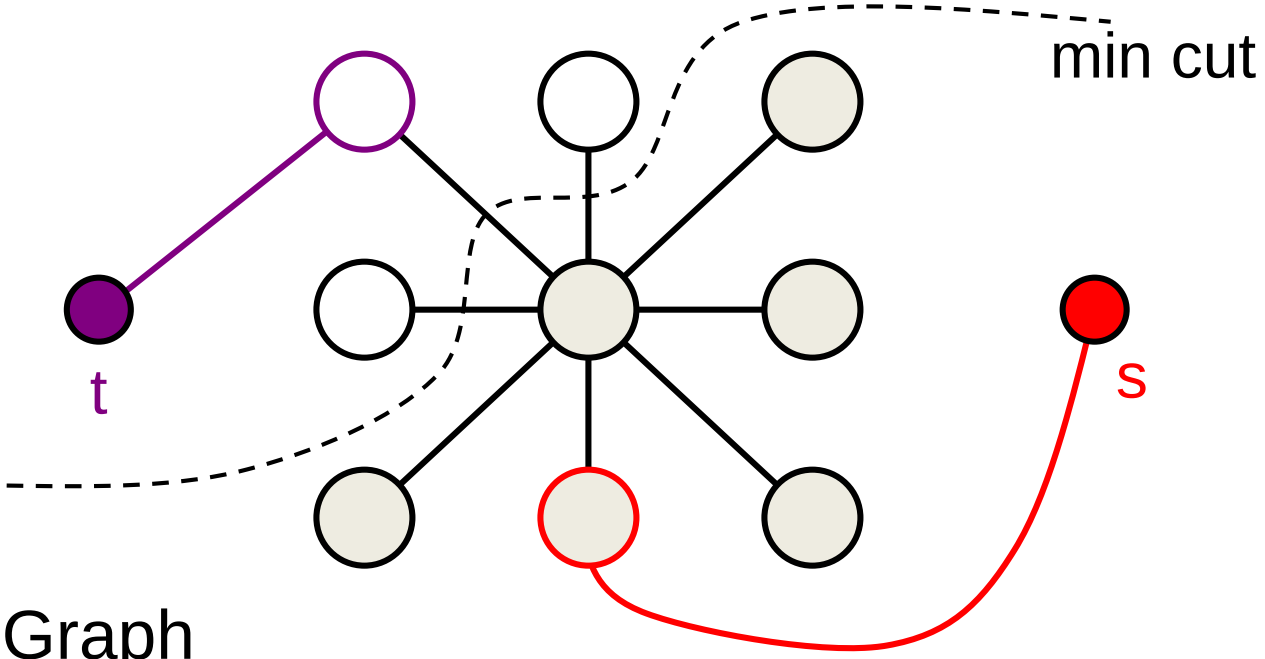
edge	weight (cost)	for
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{N}$
$\{p,S\}$	$\lambda \cdot R_p(\text{“bkg”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p,T\}$	$\lambda \cdot R_p(\text{“obj”})$	$p \in \mathcal{P},\ p \notin \mathcal{O}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

  - T-link: link to terminal (label)
    - Model for R is often determined by seeding process

$$\begin{aligned} R_p(\text{“obj”}) &= -\ln \Pr(I_p|\mathcal{O}) \\ R_p(\text{“bkg”}) &= -\ln \Pr(I_p|\mathcal{B}). \end{aligned}$$

GEORGETOWN UNIVERSITY

Segmentation by min (s-t) cut [Boykov 2001]

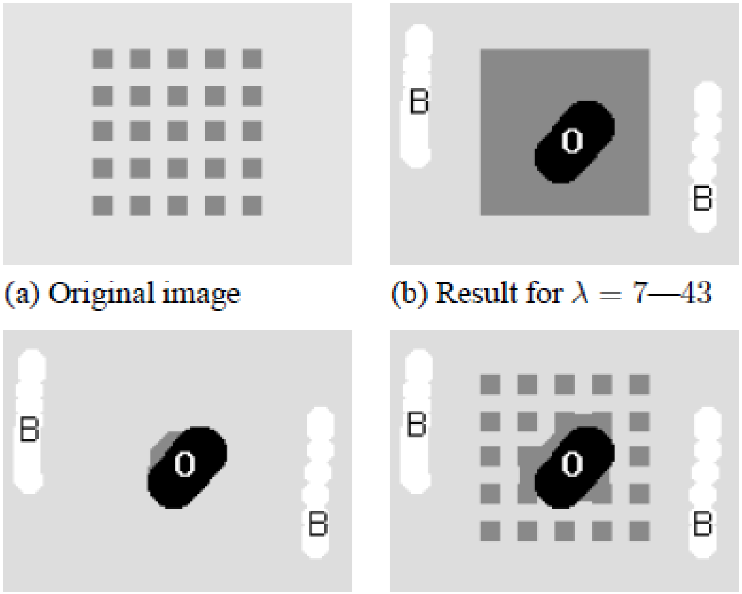


- Graph
  - node for each pixel, link between pixels
  - specify a few pixels as foreground and background
    - create an infinite cost link from each bg pixel to the “t” node
    - create an infinite cost link from each fg pixel to the “s” node
  - compute min cut that separates s from t
  - how to define link cost between neighboring pixels?

GEORGETOWN UNIVERSITY

Example Results [Boykov]

- Results
  - Dependent upon seeding process
  - Dependent on choice of \lambda



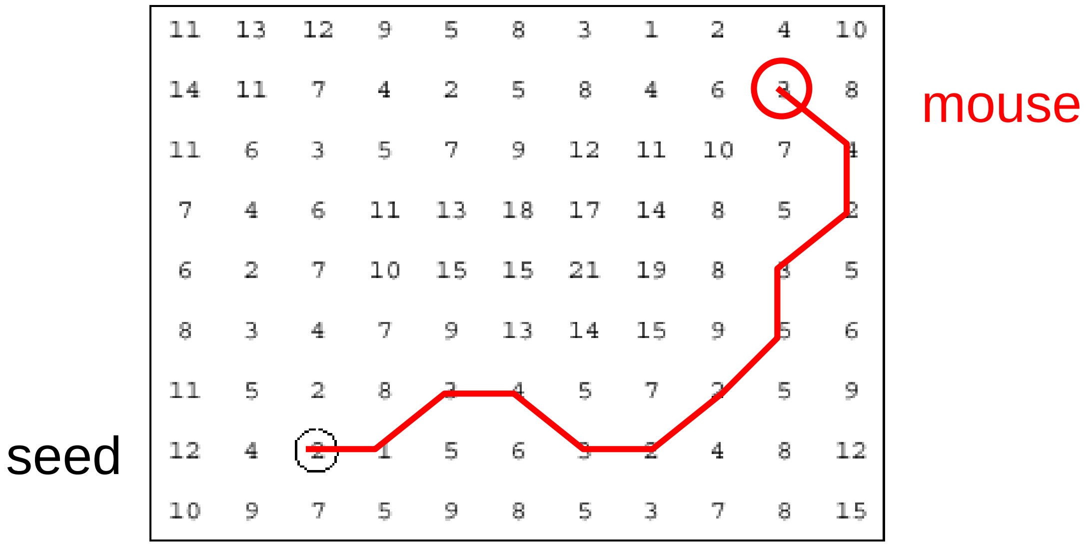
(a) Original image (b) Result for $\lambda = 7$—43

(c) Result for $\lambda = 0$ (d) Result for $\lambda = 60$

Figure 2. Synthetic Gestalt example. The segmentation results in (b-d) are shown for various levels λ of relative importance of "region" versus "boundary" in (1). Note that the result in (b) corresponds to a wide range of λ.

Intelligent Scissors (interactive, contour-based, graph-model)

- Basic Idea
  - Define edge score for each pixel
    - edge pixels have low cost
  - Find lowest cost path from seed to mouse



Questions

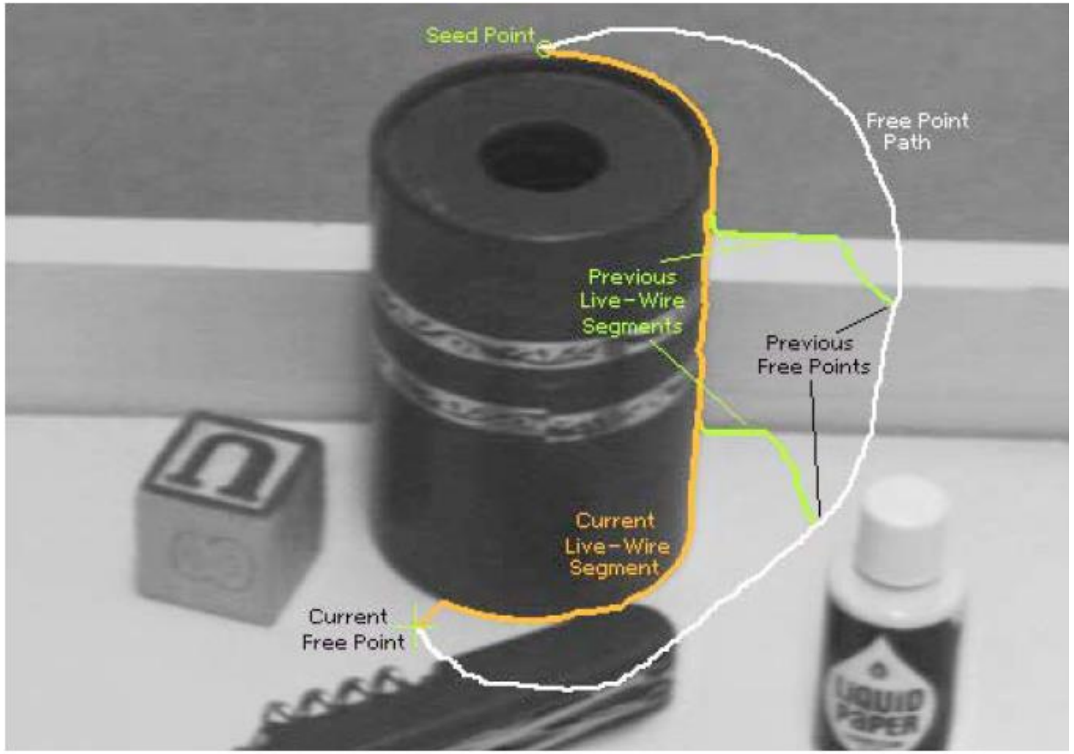
- How to define costs?
- How to find the path?

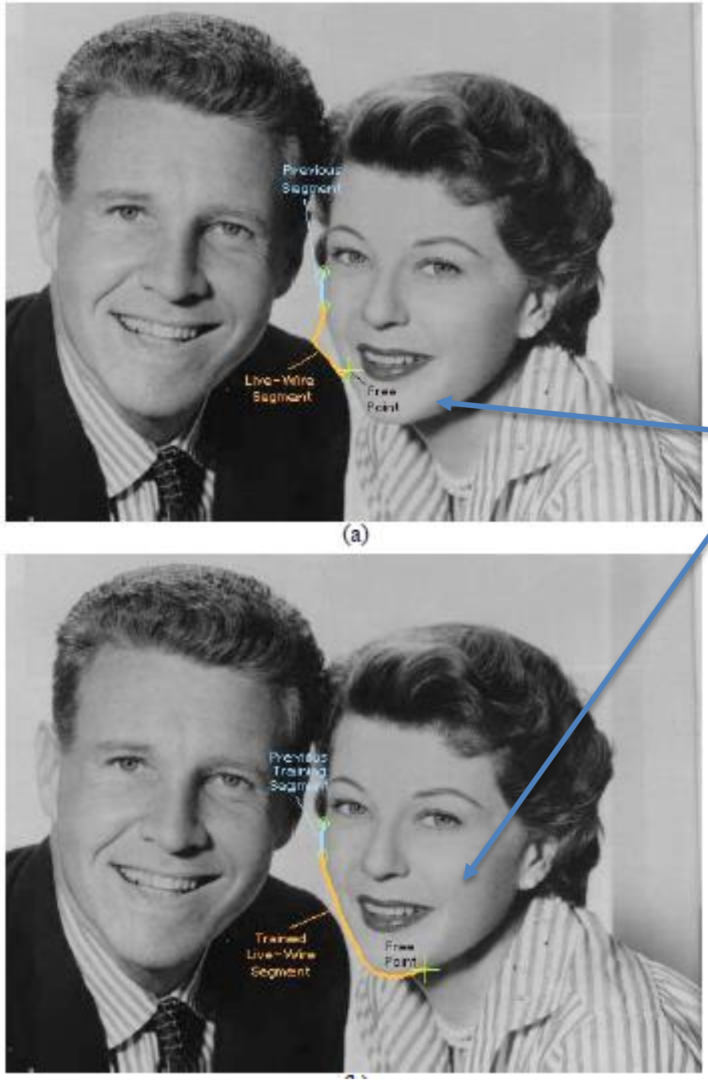
Intelligent Scissors (demo)

https://youtu.be/X_dZ_7xAcIM

Intelligent Scissors [Mortensen 95]

- Approach answers a basic question
  - Q: how to find a path from seed to mouse that follows object boundary as closely as possible?



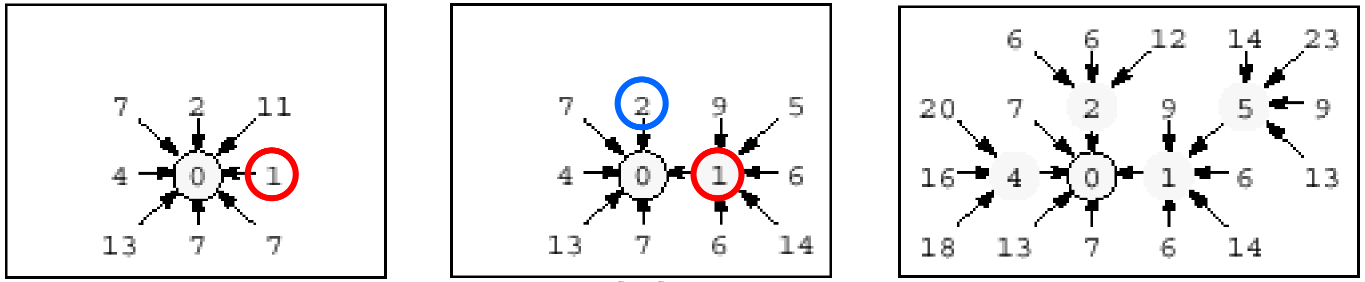


Can escape local optima as cursor moves and more information about the boundary is learned.

Path Search (basic idea)

- Graph Search Algorithm
  - Computes minimum cost path from seed to *all other pixels*

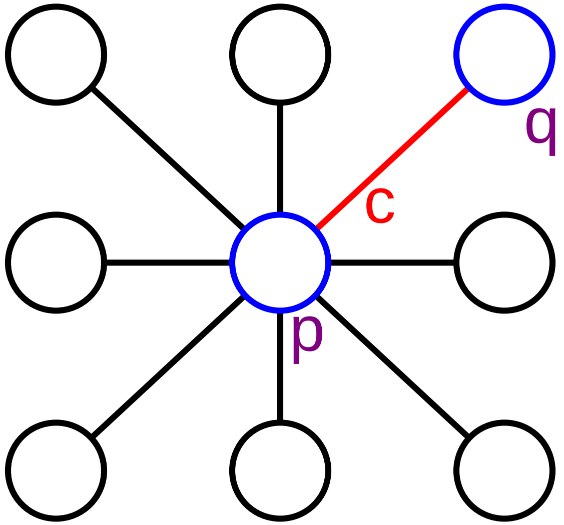
11	13	12	9	5	8	3	1	2	4	10
14	11	7	4	2	5	8	4	6	3	8
11	6	3	5	7	9	12	11	10	7	4
7	4	6	11	13	18	17	14	8	5	2
6	2	7	10	15	15	21	19	8	3	5
8	3	4	7	9	13	14	15	9	5	6
11	5	2	8	3	4	5	7	2	5	9
12	4	2	1	5	6	3	2	4	8	12
10	9	7	5	9	8	5	3	7	8	15



GEORGETOWN UNIVERSITY

How does this really work?

- Treat the image as a graph



Graph

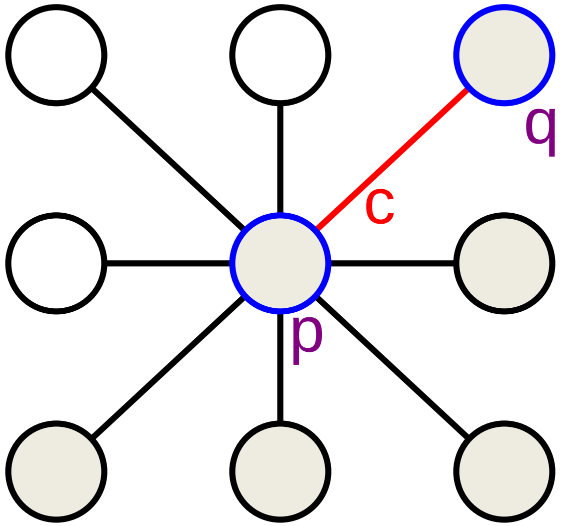
- node for every pixel **p**
- link between every adjacent pair of pixels, **p**,**q**
- cost **c** for each link

Note: each *link* has a cost

GEORGETOWN UNIVERSITY

Defining the costs

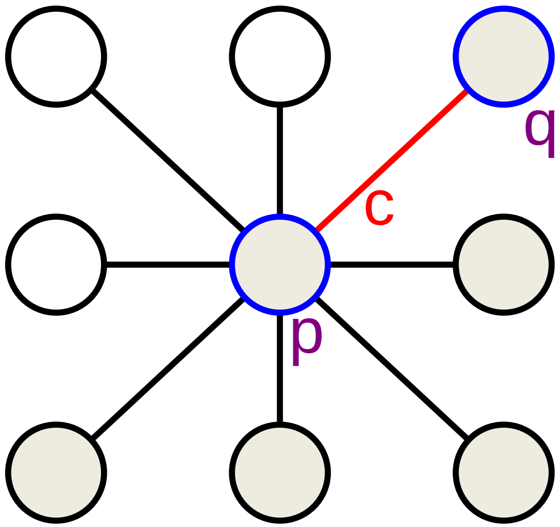
- Treat the image as a graph

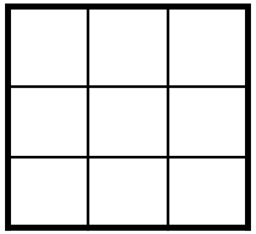


Want to “hug” image edges: how to define cost of a link?

- the link should follow the intensity edge
  - want intensity to change rapidly $\perp$ to the link
- c $\approx$ - |difference of intensity $\perp$ to link |

Defining the costs

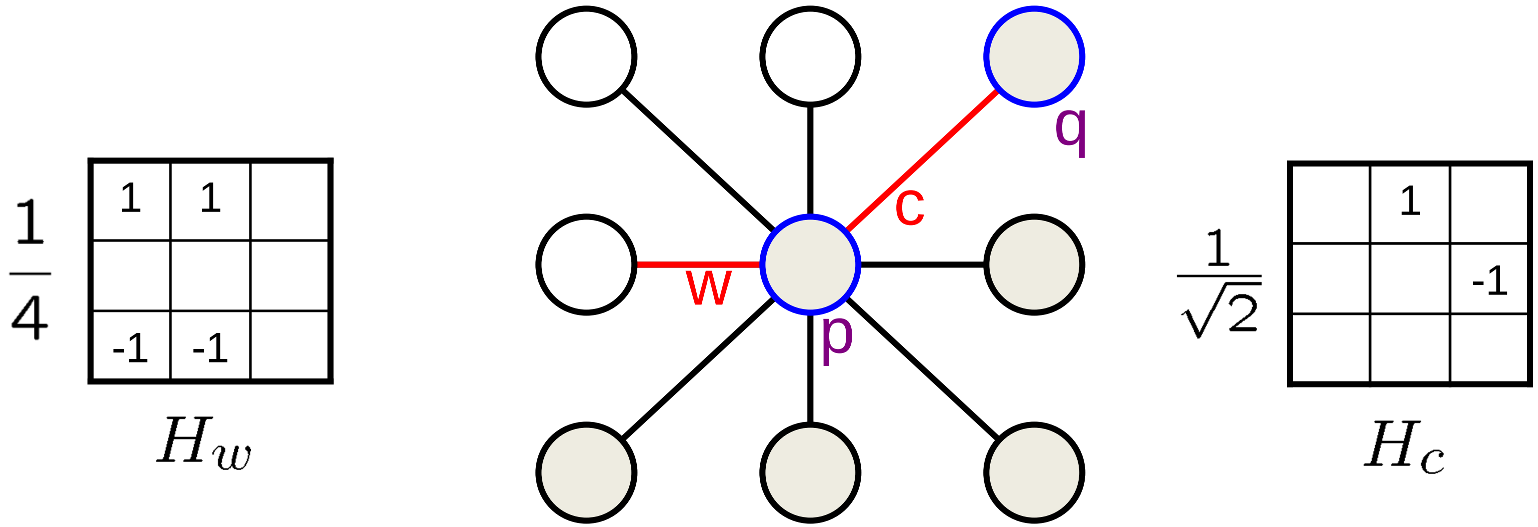




H_c

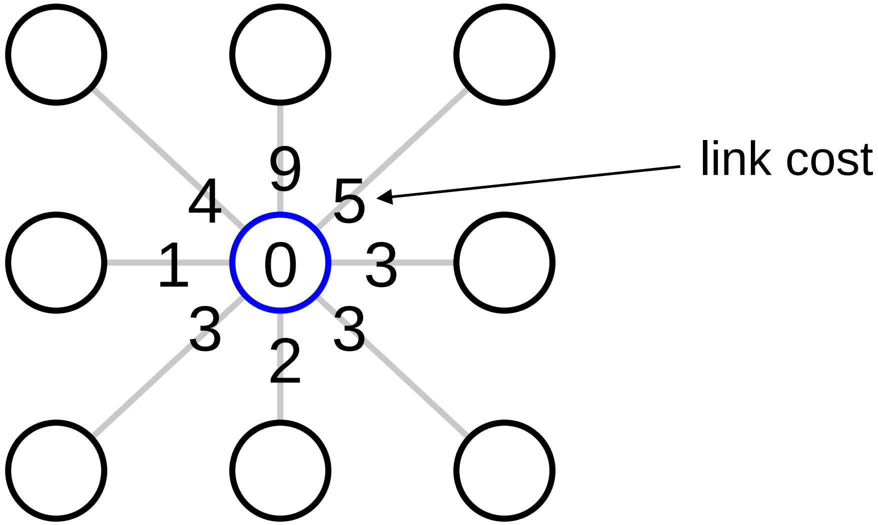
- c can be computed using a cross-correlation filter
  - assume it is centered at p
- Also typically scale c by its length
  - set c = (max-|filter response|)
    - where max = maximum |filter response| over all pixels in the image

Defining the costs



- c can be computed using a cross-correlation filter
  - assume it is centered at p
- Also typically scale c by its length
  - set c = (max-|filter response|)
    - where max = maximum |filter response| over all pixels in the image

Dijkstra's shortest path algorithm



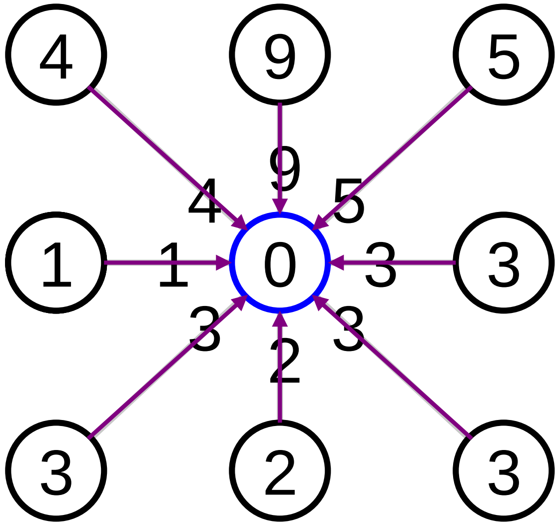
Algorithm

1. init node costs to ∞, set p = seed point, cost(p) = 0
2. expand p as follows:

   for each of p's neighbors q that are not expanded

   » set cost(q) = min(cost(p) + c_{pq}, cost(q))

Dijkstra's shortest path algorithm



Algorithm

1. init node costs to ∞, set p = seed point, cost(p) = 0
2. expand p as follows:

   for each of p's neighbors q that are not expanded

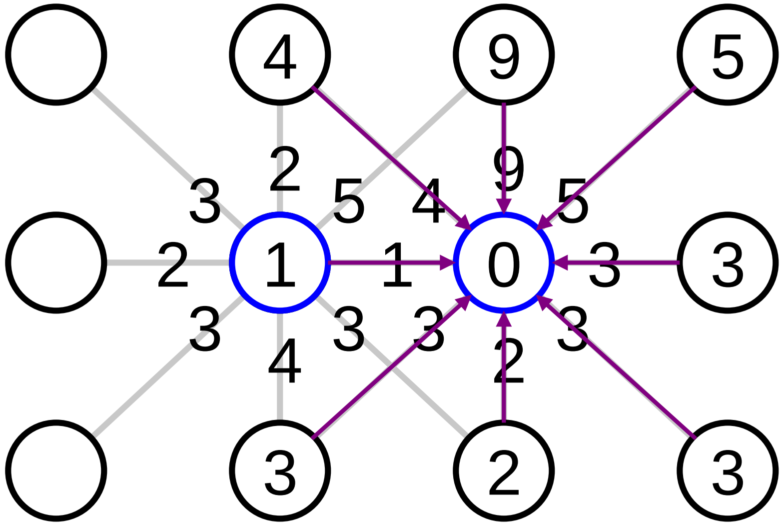
   » set cost(q) = min(cost(p) + c_{pq}, cost(q))

      » if q's cost changed, make q point back to p

   » put q on the ACTIVE list (if not already there)

GEORGETOWN UNIVERSITY

Dijkstra's shortest path algorithm



Algorithm

1. init node costs to ∞, set p = seed point, cost(p) = 0
2. expand p as follows:

   for each of p's neighbors q that are not expanded

   » set cost(q) = min(cost(p) + c_{pq}, cost(q))

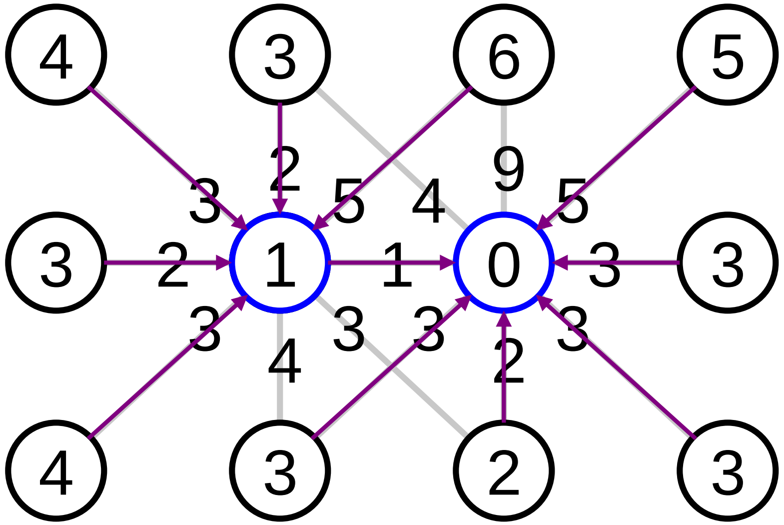
      » if q's cost changed, make q point back to p

   » put q on the ACTIVE list (if not already there)

3. set r = node with minimum cost on the ACTIVE list
4. repeat Step 2 for p = r

GEORGETOWN UNIVERSITY

Dijkstra's shortest path algorithm



Algorithm

1. init node costs to ∞, set p = seed point, cost(p) = 0
2. expand p as follows:

   for each of p's neighbors q that are not expanded

   » set cost(q) = min(cost(p) + c_{pq}, cost(q))

   » if q's cost changed, make q point back to p

   » put q on the ACTIVE list (if not already there)
3. set r = node with minimum cost on the ACTIVE list
4. repeat Step 2 for p = r

GEORGETOWN UNIVERSITY

Dijkstra's shortest path algorithm

- Properties
  - It computes the minimum cost path from the seed to every node in the graph. This set of minimum paths is represented as a *tree*
  - Running time, with N pixels:
    - $O(N^2)$ time if you use an active list
    - O(N log N) if you use an active priority queue (heap)
    - takes fraction of a second for a typical (640x480) image
  - Once this tree is computed once, we can extract the optimal path from any point to the seed in O(N) time.
    - it runs in real time as the mouse moves
  - What happens when the user specifies a new seed?

Grabcut [Rother et al., SIGGRAPH 2004]

https://youtu.be/ufiTlDp4Iqc

GrabCut: Iterated Graph Cuts similar to Boykov et al

User Initialisation

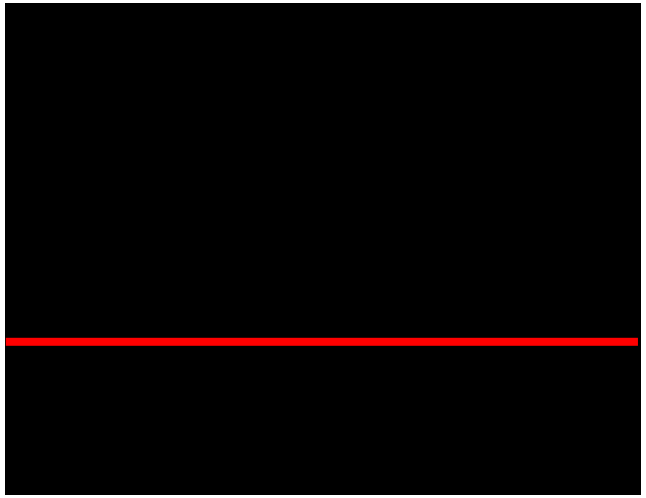
Learn foreground color model

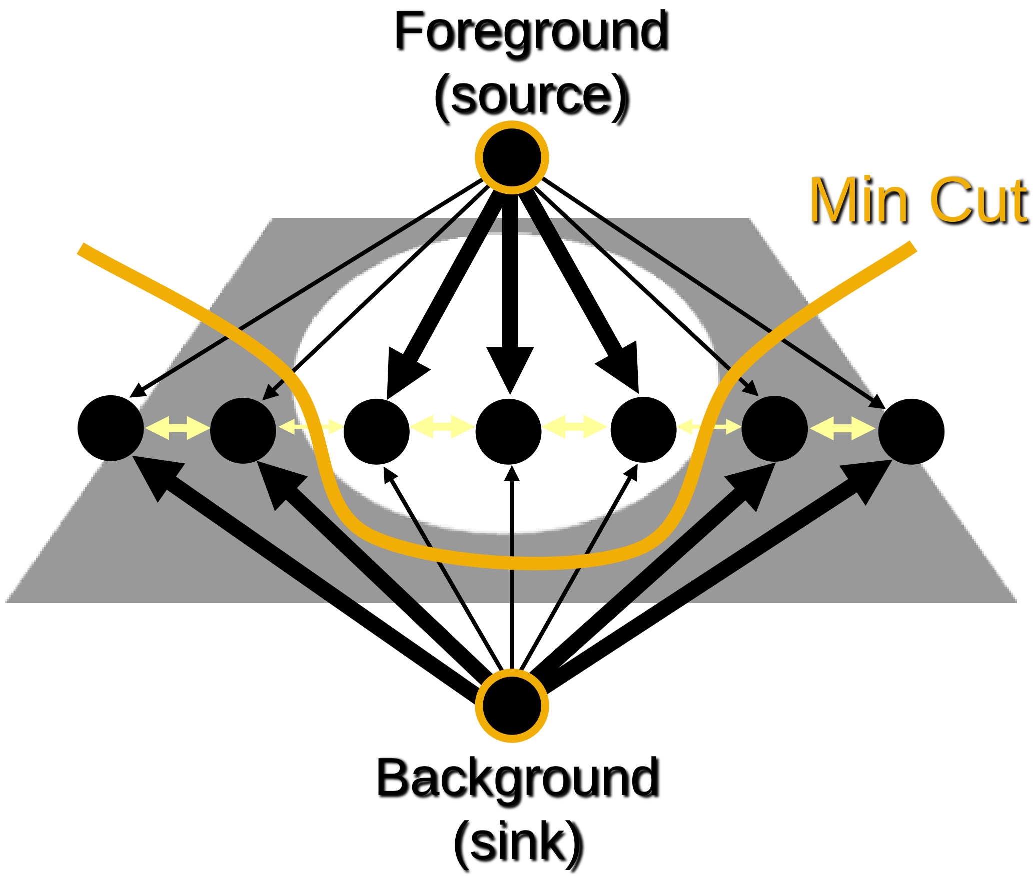
Graph cuts to infer the foreground

GEORGETOWN UNIVERSITY

Graph Cuts modelling in images

Image



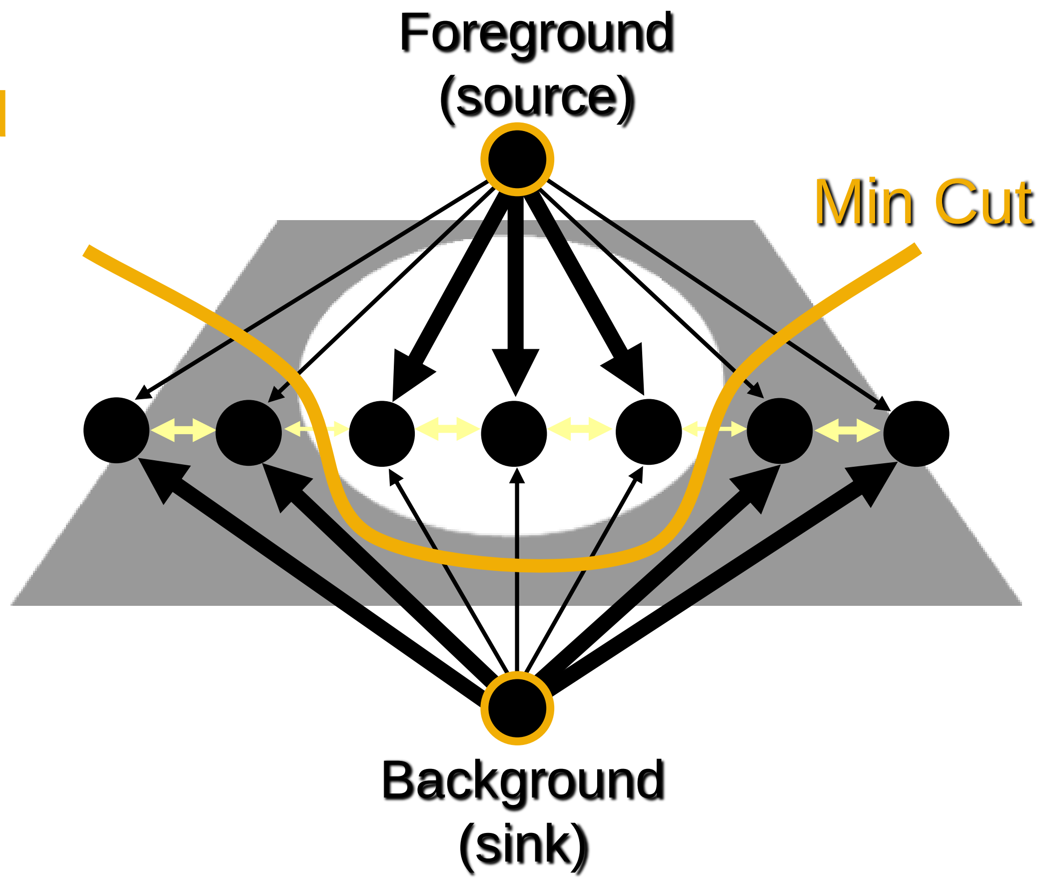


Cut: separating source and sink; Energy: collection of edges

Min Cut: Global minimal enegry in polynomial time

Graph Cuts for foreground extraction

Assume we know foreground is **white** and background is **black**

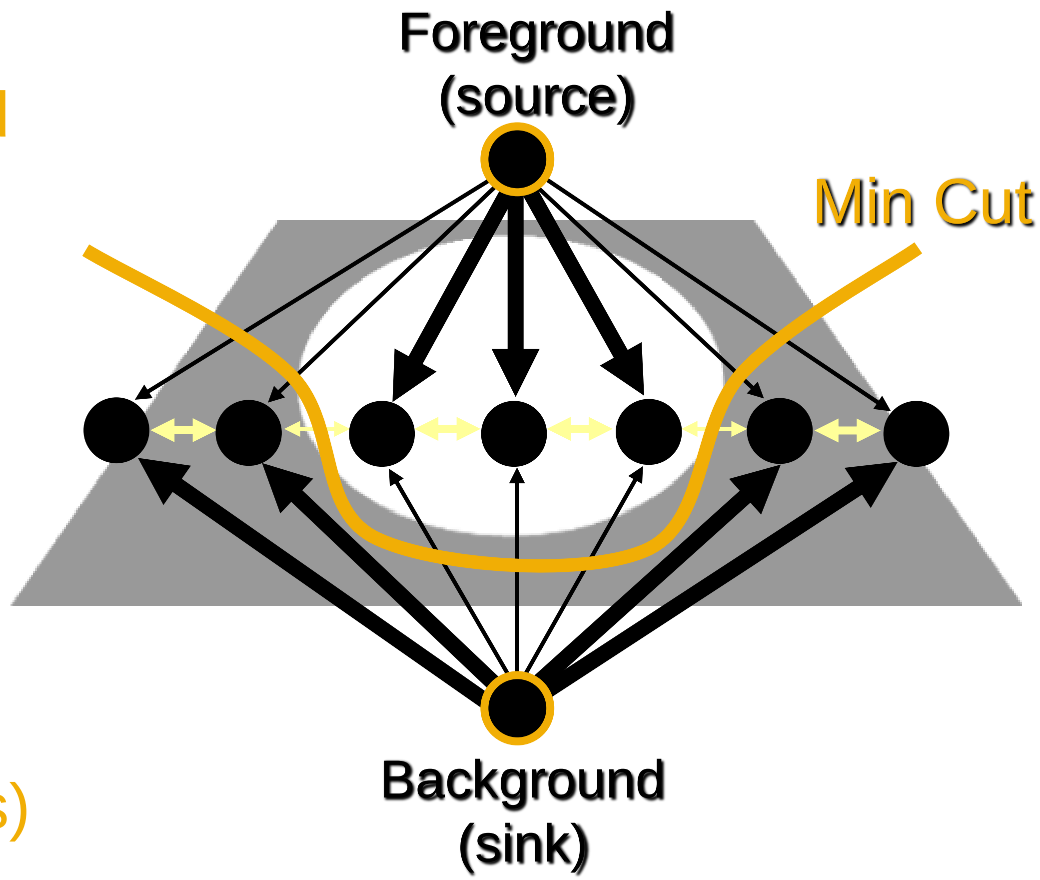


Graph Cuts for foreground extraction

Assume we know foreground is **white** and background is **black**

Data term = (cost of assigning label)

Regularization = (cost of separating neighbors)



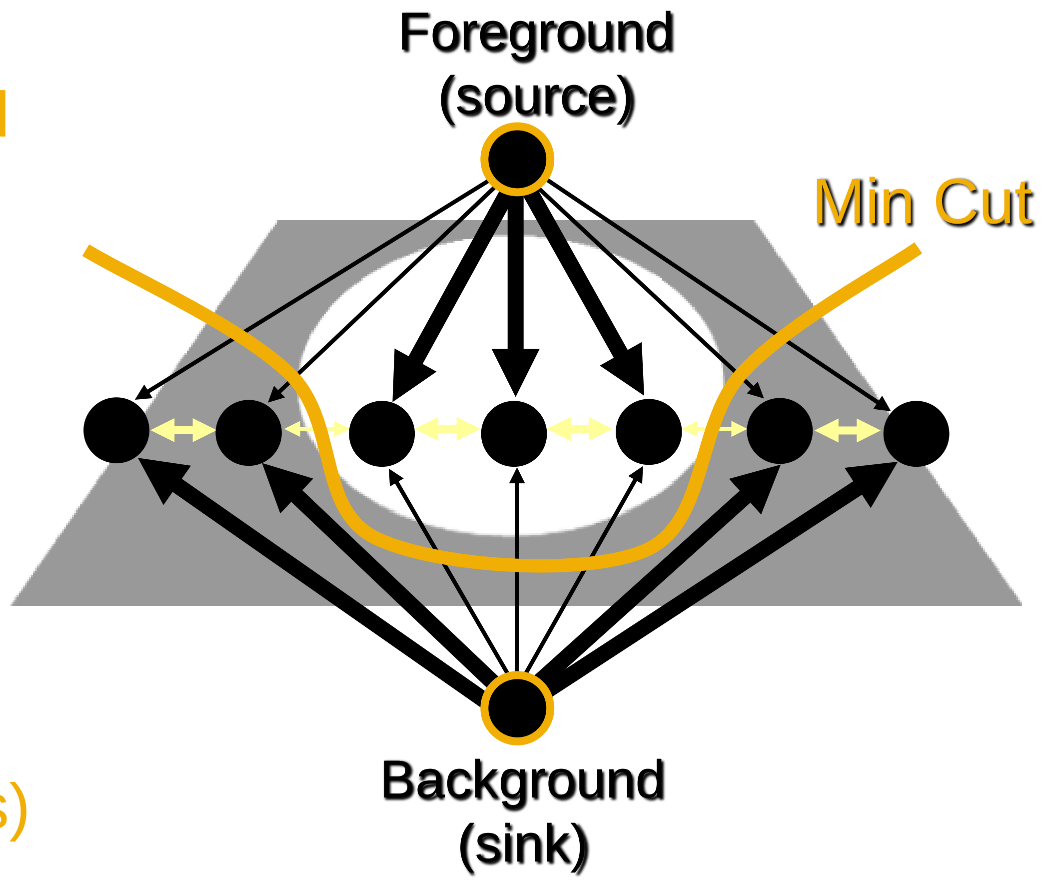
GEORGETOWN UNIVERSITY

Graph Cuts for foreground extraction

Assume we know foreground is **white** and background is **black**

Data term = **whiteness** (cost of assigning label)

Regularization = (cost of separating neighbors)

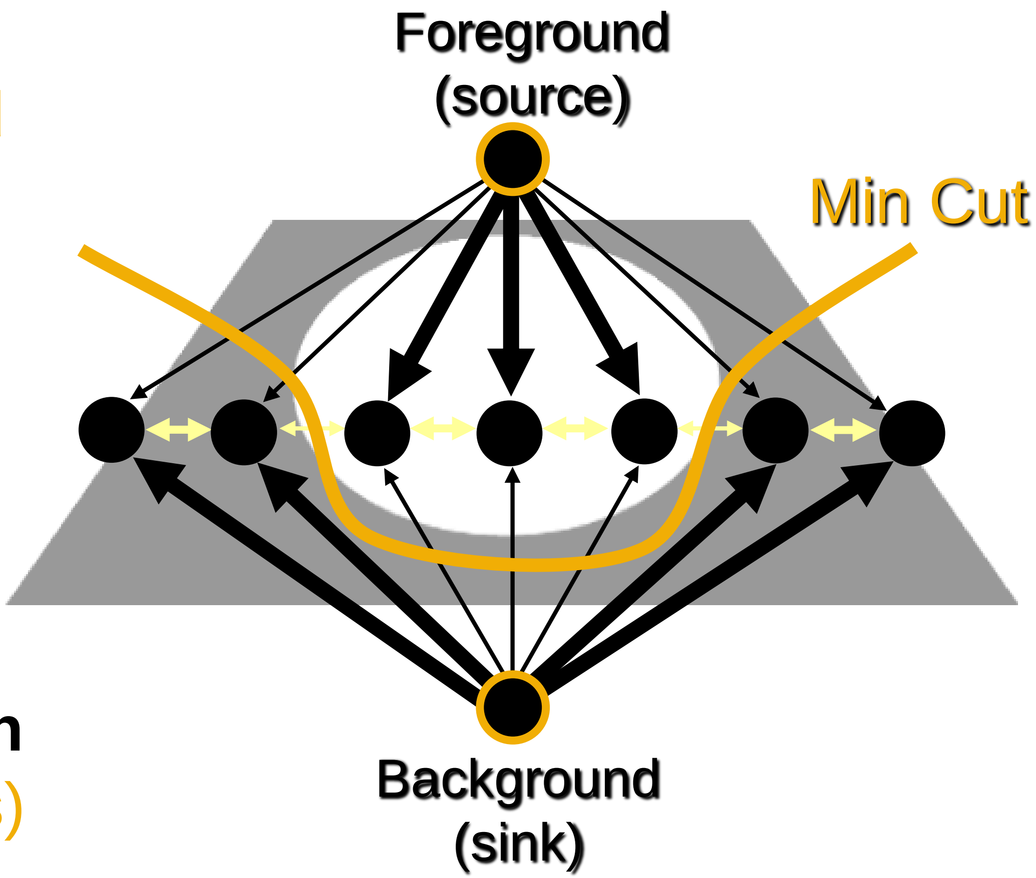


Graph Cuts for foreground extraction

Assume we know foreground is **white** and background is **black**

Data term = **whiteness**
(cost of assigning label)

Regularization = **color match**
(cost of separating neighbors)



GEORGETOWN UNIVERSITY

We are all set now !

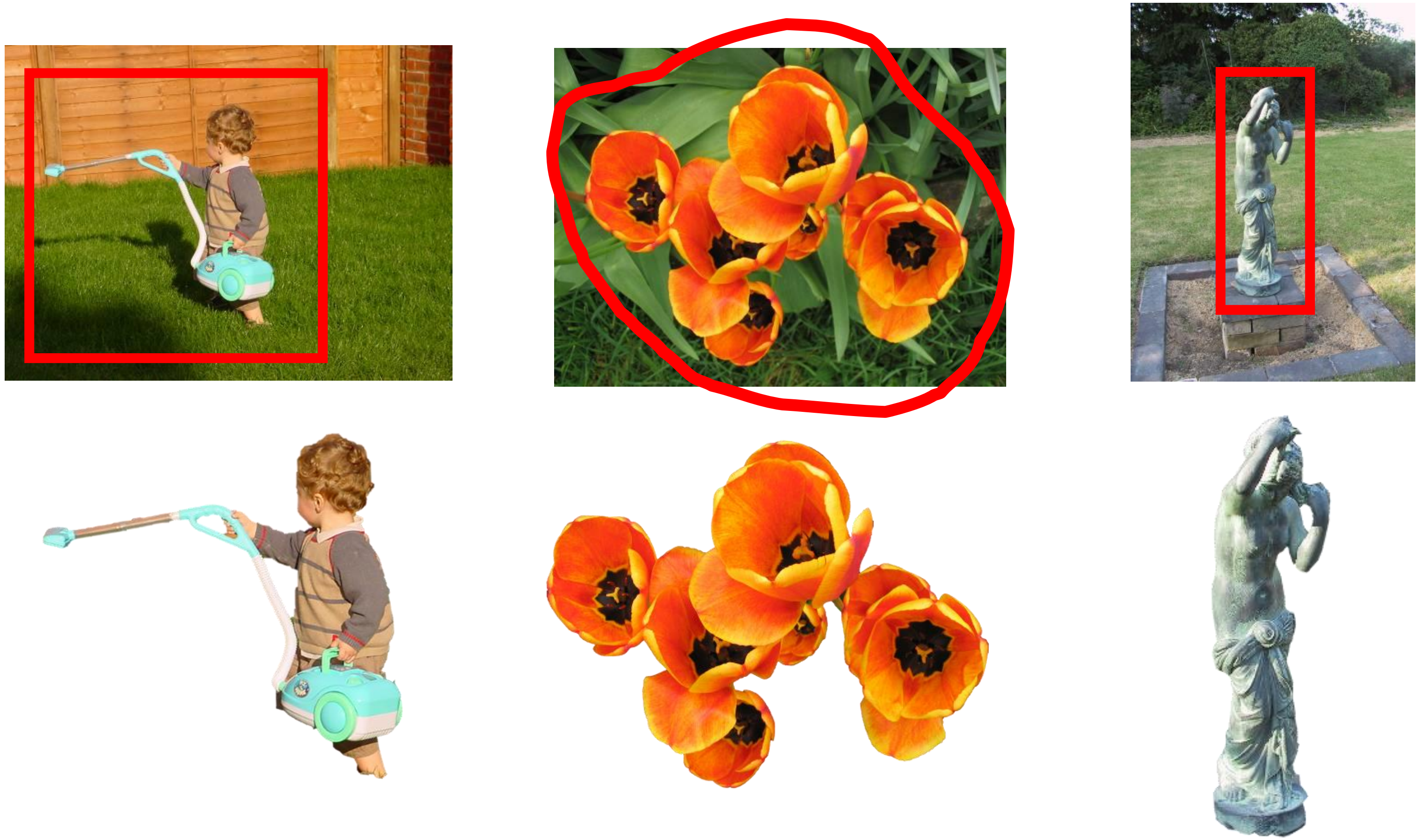
User Initialisation

Learn foreground color model

Graph cuts to infer the foreground

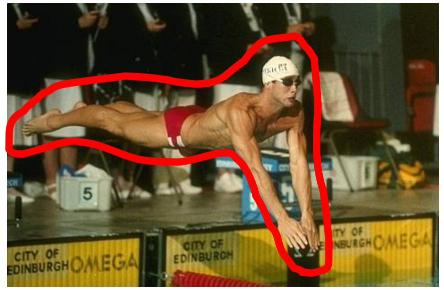
GEORGETOWN UNIVERSITY

Moderately straightforward examples



… GrabCut completes automatically

Difficult Examples

	Camouflage & Low Contrast	Fine structure	No telepathy
Initial Rectangle			
Initial Result			

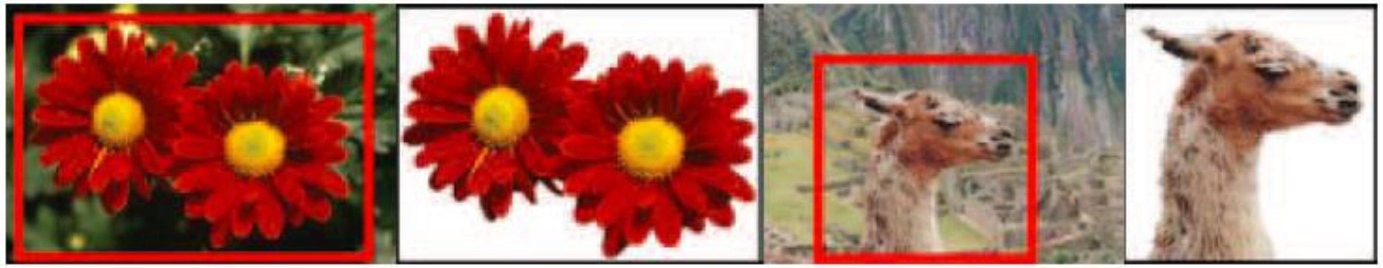


FIGURE 9.12: In a grabcut interface for interactive segmentation, a user marks a box around the object of interest; foreground and background models are then inferred by a clustering method, and the object is segmented. If this segmentation isn't satisfactory, the user has the option of painting foreground and background strokes on pixels to help guide the model. *This figure was originally published as Figure 1 of "GrabCut Interactive Foreground Extraction using Iterated Graph Cuts" by C. Rother, V. Kolmogorov, and A. Blake, Proc. ACM SIGGRAPH, 2004 © ACM, 2004.*

GEORGETOWN UNIVERSITY

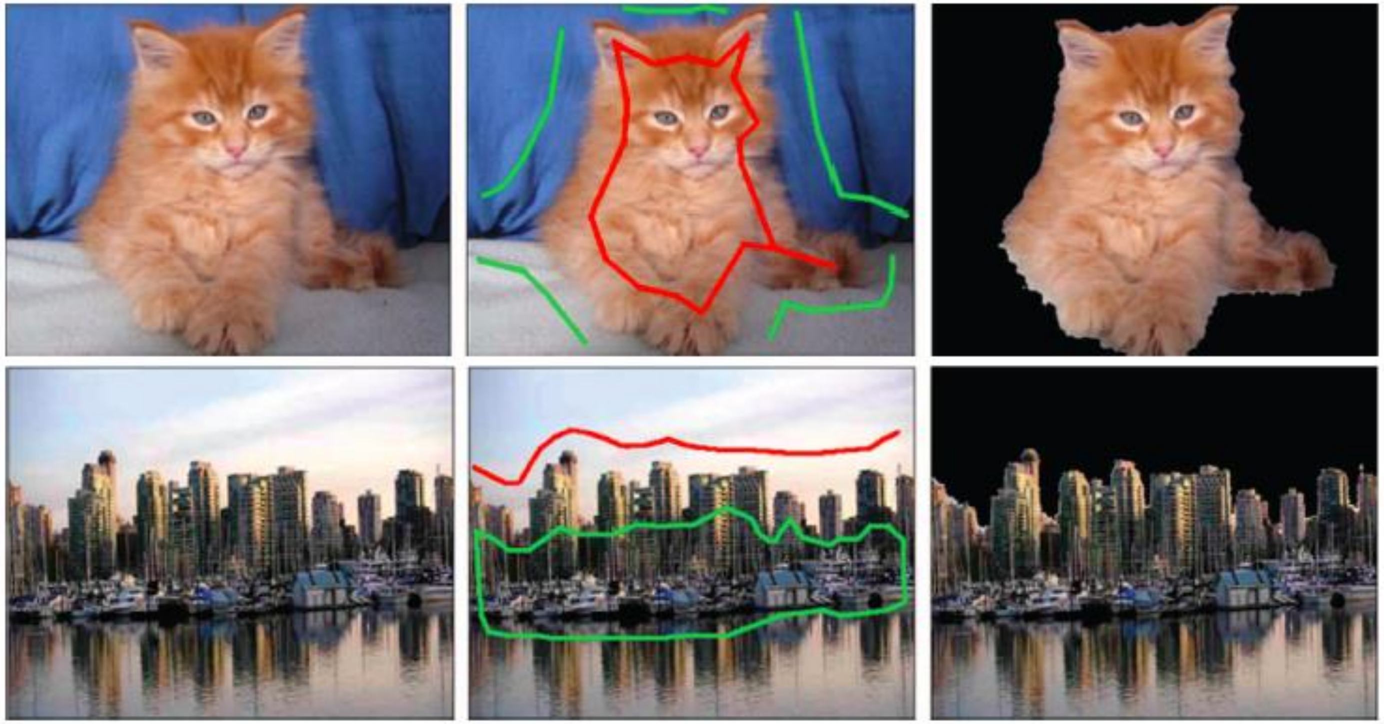


FIGURE 9.11: A user who wants to cut an object out of an image (**left**) could mark some foreground pixels and some background pixels (**center**), then use an interactive segmentation method to get the cut out components on the **right**. The method produces a model of foreground and background pixel appearance from the marked pixels, then uses this information to decide a figure ground segmentation. *This figure was originally published as Figure 9 of "Interactive Image Segmentation via Adaptive Weighted Distances," by Protiere and Sapiro, IEEE Transactions on Image Processing, 2007 © IEEE, 2007.*

GEORGETOWN UNIVERSITY

Notes: Is user-input required?

- Fully-Automated vs semi-automated
  - Automatic methods are possible, but are often very application specific
    - classical image segmentation methods are automatic
  - Argument for user-directed methods?
    - Pros: only user knows desired scale/object of interest
    - Cons: Requires a User
  - Many approaches and variants exist

GEORGETOWN UNIVERSITY