An Introduction to Graph-Cut

By: Paul Scovanner

An Introduction to Graph-Cut

- Overview
  - Min-cut
  - Normalized-Cut
- Applications

An Introduction to Graph-Cut

- Graph-cut is an algorithm that finds a globally optimal segmentation solution.
- Also know as Min-cut.
- Equivalent to Max-flow. [1]

[1] Wu and Leahy: An Optimal Graph Theoretic Approach to Data Clustering:…

What is a “cut”?

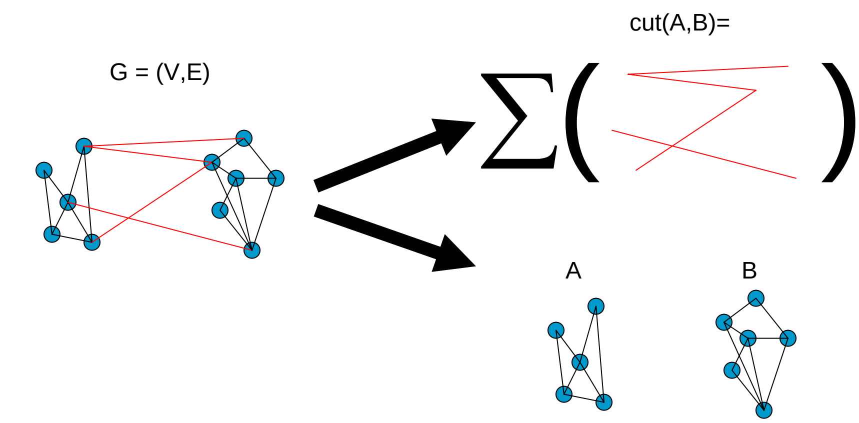
A graph G = (V,E) can be partitioned into two disjoint sets, $A, B, A \cup B = V, A \cap B = 0$
by simply removing edges connecting the two parts.

The degree of dissimilarity between these two pieces can be computed as total weight of the edges that have been removed. In graph theoretic language it is called the *cut*:

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$$ [2].

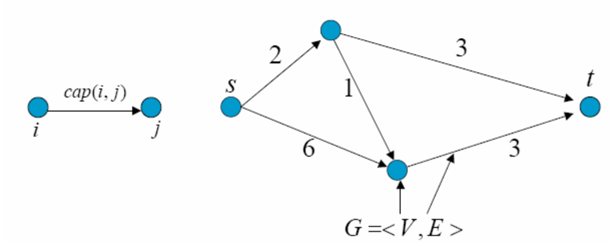
[2] Shi and Malik: Normalized cuts and image segmentation.

Example cut



Finding the Minimum-cut

- A source node and a sink node
- Directed edge (Arc) <i,j> from node i to node j
- Each arc <i,j> has a nonnegative capacity cap(i,j)
- $cap(i,j) = 0$ for non-exist arcs



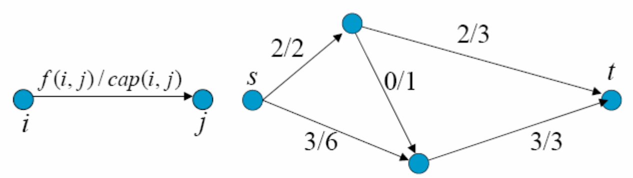
Finding the Minimum-cut

- Flow is a real value function f that assign a real value $f(i,j)$ to each arc <i,j> under :
  - Capacity constraint : $f(i,j) \le cap(i,j)$
  - Mass balance constraint:

$$\sum_{<i,j>\in E} f(i,j) - \sum_{<k,i>\in E} f(k,i) = \begin{cases} 0 & i \in V - \{s,t\} \\ |f| & i = s \\ -|f| & i = t \end{cases}$$

$|f|$ is the value of flow f

Finding the Minimum-cut

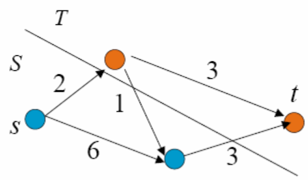


An example of flow

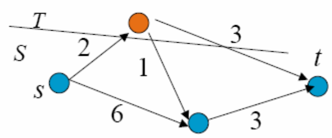
- maximum flow is the flow has maximum value among all possible flow functions

Finding the Minimum-cut

- A cut is a partition of node set V which has two subsets S and T
- A cut is a s-t cut iif $s \in S, t \in T$



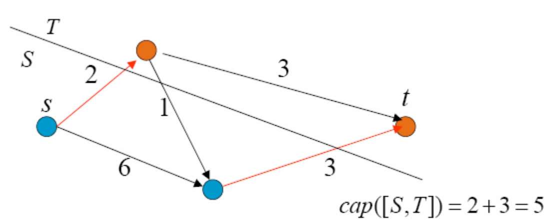
s-t cut



not a s-t cut

Finding the Minimum-cut

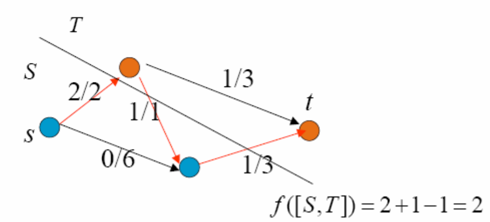
- $$cap([S,T]) = \sum_{<i,j> \in E, i \in S, j \in T} cap(i,j)$$



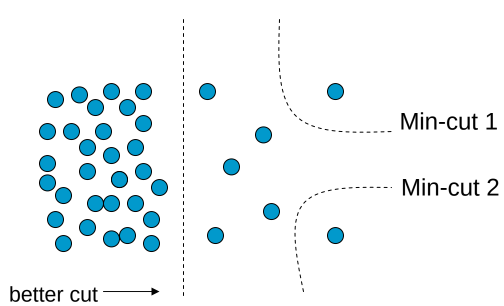
- Minimum cut is the s-t cut whose capacity is minimum among all possible s-t cuts

Finding the Minimum-cut

- $f([S,T]) = \sum_{<i,j> \in E, v_i \in S, v_j \in T} f(i,j) - \sum_{<j,i> \in E, v_i \in S, v_j \in T} f(j,i)$



The problem with Min-cut



Need to account for cluster similarity

Normalized-cut

- Instead use *normalized cut (Ncut).*

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

$$assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$$

$$Nassoc(A,B) = \left(\frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)} \right)$$

Normalized-cut

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

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

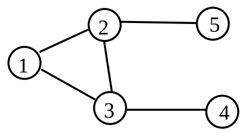
$$= 2 - \left(\frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)} \right)$$

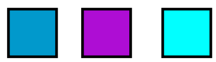
$$= 2 - Nassoc(A,B)$$

Pixel labeling problem

Given

$\mathcal{S} = \{1, \cdots, n\}$ $\mathcal{N} \subseteq \mathcal{S} \times \mathcal{S}$



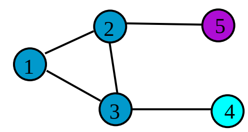
$\mathcal{L} = \{l_1, \ldots, l_m\}$ 

Assignment cost for giving a particular label to a particular node. Written as D.

Separation cost for assigning a particular pair of labels to neighboring nodes. Written as V.

Find

Labeling f = $(f_1, \ldots, f_n)$



Such that the sum of the assignment costs and separation costs (the energy E) is small

Energy Minimization

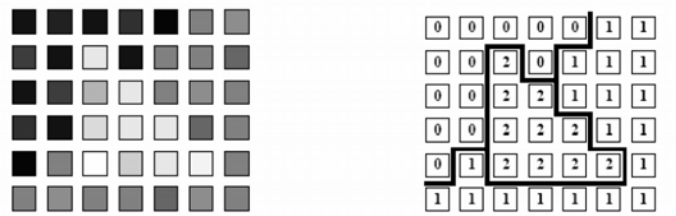
- Optimizing the labeling problem can be thought of as minimizing some energy function.

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in N} V_{p,q}(f_p, f_q)$$

$D_p(f_p)$ measure of image discrepancy

$\sum_{p,q \in N} V_{p,q}(f_p, f_q)$ measure of smoothness or other visual constraints

The Labeling Problem



(a) An image (b) A labeling

Common idea behind many Computer Vision problems

Assign labels to pixels based on noisy measurements (input images)

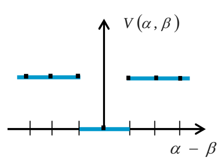
In the presence of uncertainties, find the best Labeling !

(Stereo, 3D Reconstruction, Segmentation, Image Restoration)

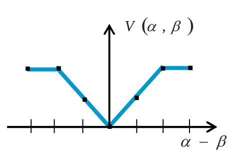
Choices of *V*

Robust

Potts model

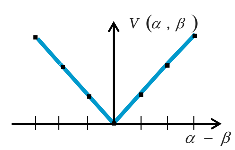


Truncated linear model

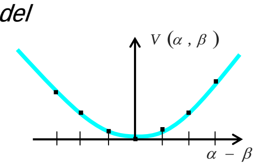


Not robust

Linear model



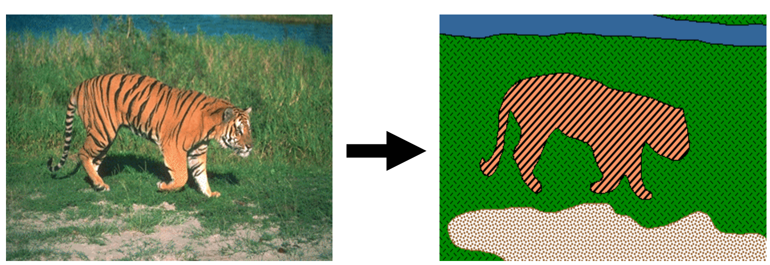
Quadratic model



What do graph cuts provide?

- For less interesting *V*, polynomial algorithm for global minimum!
- For a particularly interesting *V*, approximation algorithm
  - Proof of NP hardness
- For many choices of *V*, algorithms that find a “strong” local minimum
- Very strong experimental results

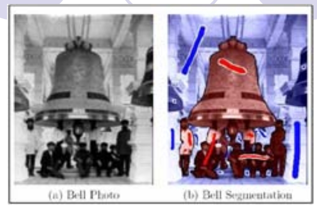
Graph Cut based Segmentation

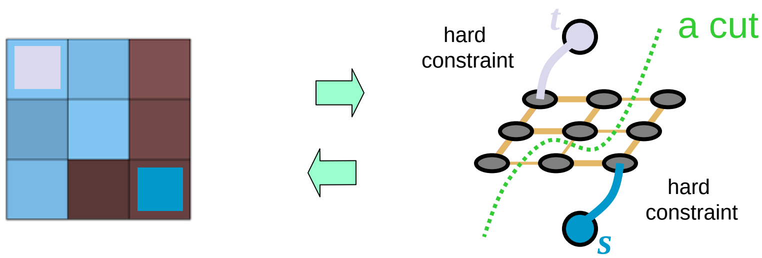


Graph Cut based Segmentation

User Guided Segmentation;

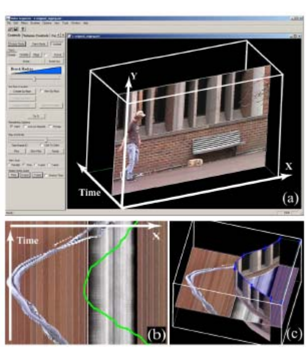
Specifies hard constraints.

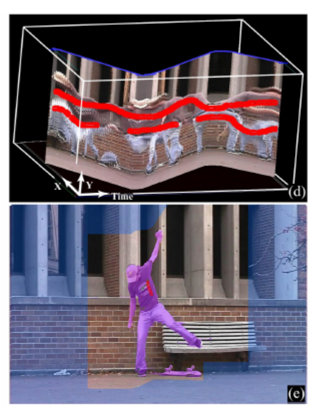




3D Applications

2D and Time





[3] Wang et al.: Interactive Video Cutout

Some Results

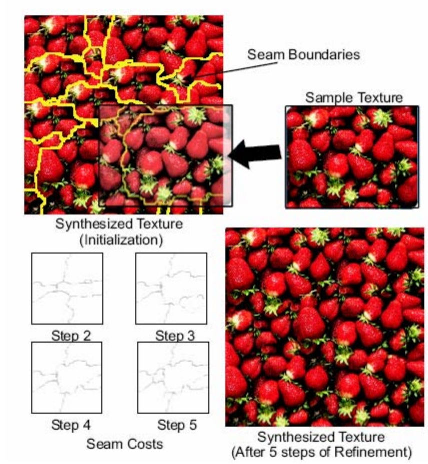
Input Image

Efros and Freeman '01
(Image Quilting)

Graph-cut

[3] Kwatra et al.: Graphcut Textures: Image and Video Synthesis Using Graph Cuts

Some Results

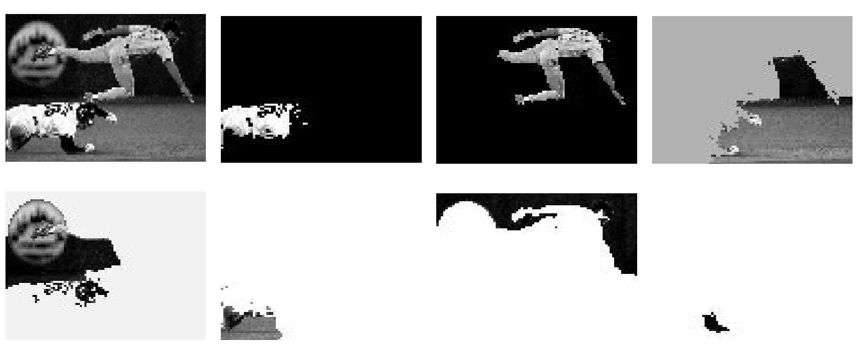


[4] Kwatra et al.: Graphcut Textures: Image and Video Synthesis Using Graph Cuts

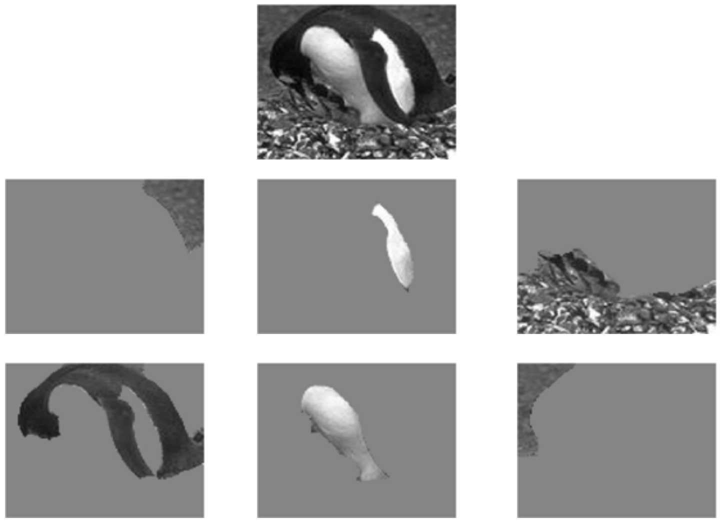
Some Results

[5] Rother, Kolmogorov and Blake:
“GrabCut“ - Interactive Foreground Extraction using Iterated Graph Cuts

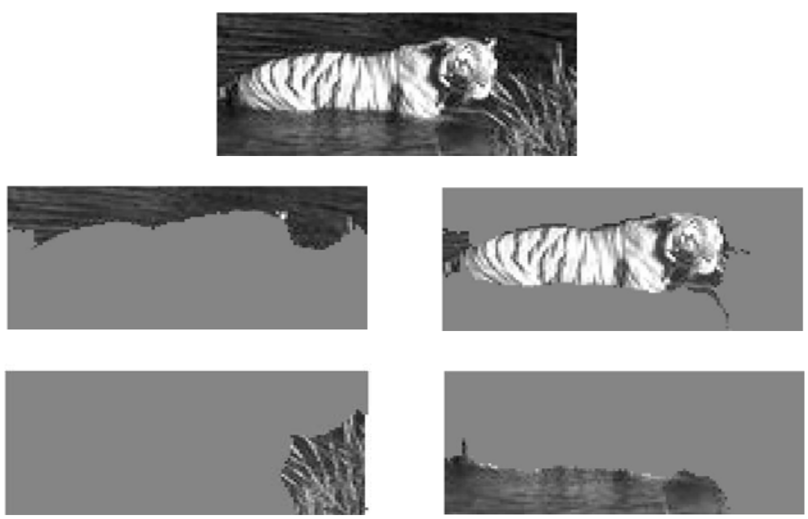
Some Results



Some Results



Some Results



- So far weight measurement has been distance, but could also use appearance, texture, or other information to calculate similarity measure.