Approximate Minimization using Graph Cuts

Boykov, Veksler, Zabih, IEEE PAMI 23(11), pp 1222ff, 2001

Slides modified from those graciously provided by Ramin Zabih

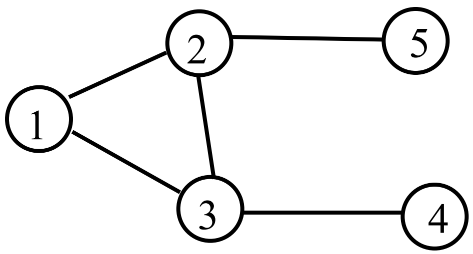
Outline (Part 1)

- Graph cuts for pixel labeling problems
  - Problem definition and motivation
  - Underlying graph algorithm (max flow)
- Global and strong local minima
  - Convex: exact global minimum
  - Non-convex: expansion move algorithm
- Theoretical and experimental properties
  - How close do we get to the global minimum?
  - What problems can graph cuts solve?

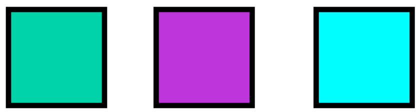
Pixel labeling problem

Given

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



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

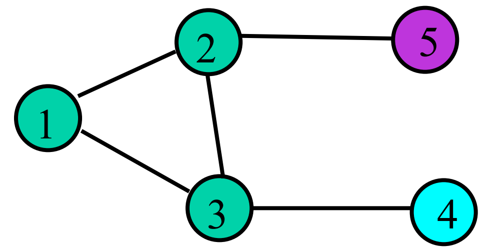


Assignment cost for giving a particular label to a particular node. Written as D.

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

Find

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



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

Solving pixel labeling problems

- We want to minimize the energy $E(f)$

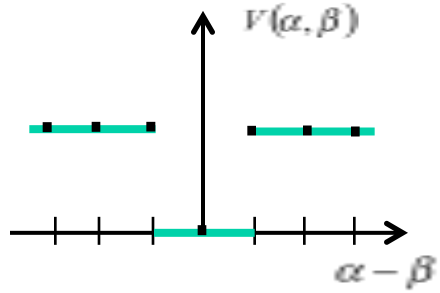
$$\arg\min_f \underbrace{\sum_{p\in\mathcal{S}} D_p(f_p)}_{\text{assignment costs}} + \underbrace{\sum_{p,q\in\mathcal{N}} V(f_p, f_q)}_{\text{separation costs}}$$

- Classical problem in vision and beyond
- Bayesian justification
  - Markov Random Fields (MRF's)

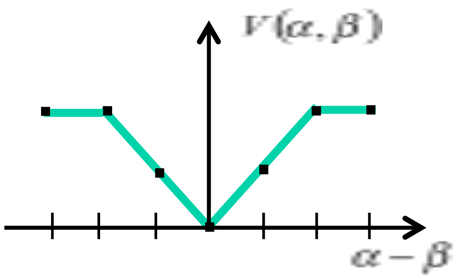
Choices of V

Robust

Potts model

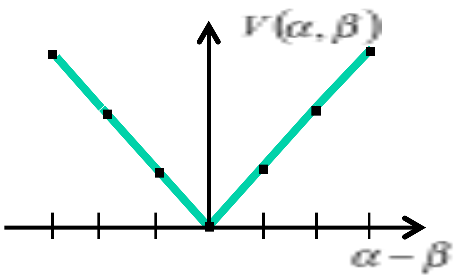


Truncated linear model

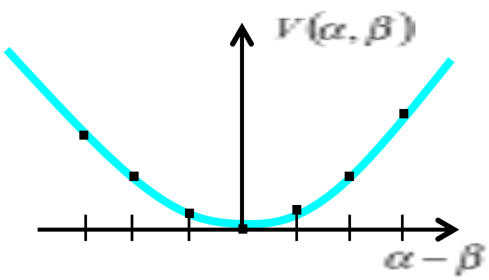


Not robust

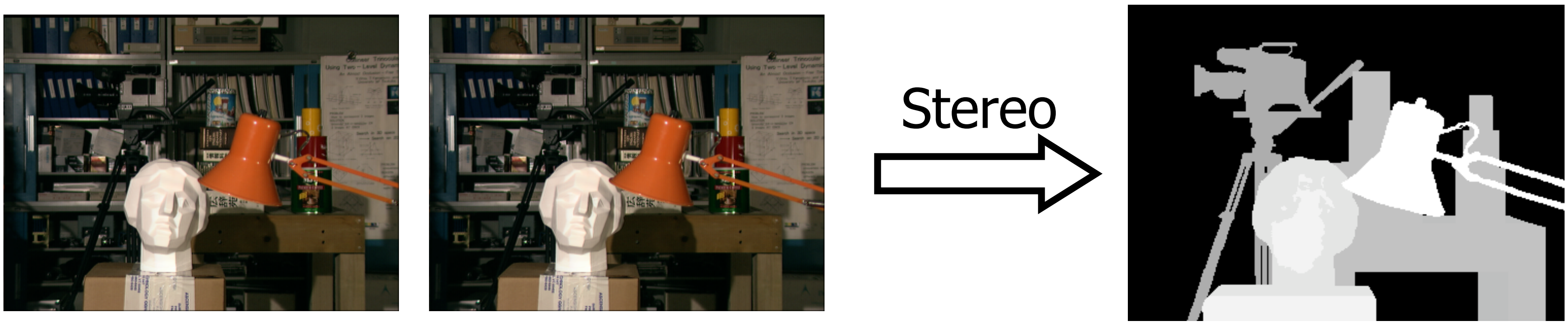
Linear model



Quadratic model



Pixel labeling for stereo



- Labels are shifts (hence depths)
- Assignment cost from intensity difference

$$D_p(\delta) = [I(p) - I'(p+\delta)]^2$$

- Neighboring pixels should be at similar depths
  - Except at the borders of objects!

How to minimize the energy?

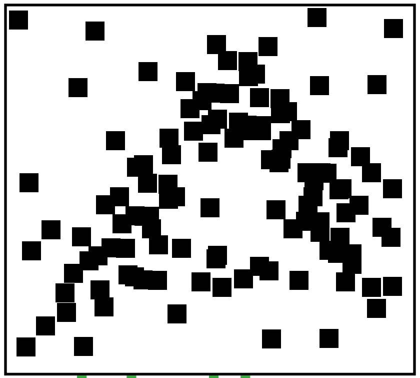
- Until late-90’s, poor solutions
  - Problem is NP-hard [K/BVZ PAMI ’01]
- In vision, we tend to focus on the deriving the “right” energy function
  - Minimize via general-purpose methods
  - Annealing, MCMC, CG, etc.
- Computer scientists disagree
  - General-purpose methods must be weak
  - Nearby energy functions can be “easy”

Graph cuts

- Reduce energy minimization problem to computing the min *s-t* cut on a graph
  - Cuts are labelings, cut costs are energy
  - Rapidly solvable by max flow
- Running times are linear in the number of pixels and labels
  - Asymptotically, low-order polynomial

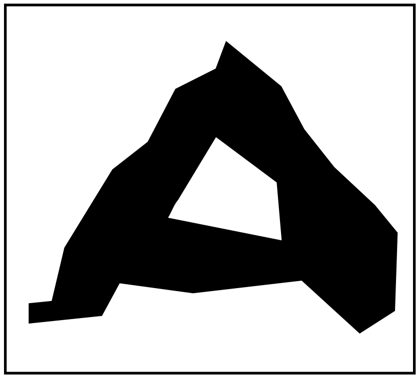
Binary image labeling problem

- Suppose we receive a noisy fax:
  - Some black pixels in the original image were flipped to white pixels, and some white pixels were flipped to black
  - We want to recover the original fax



original image

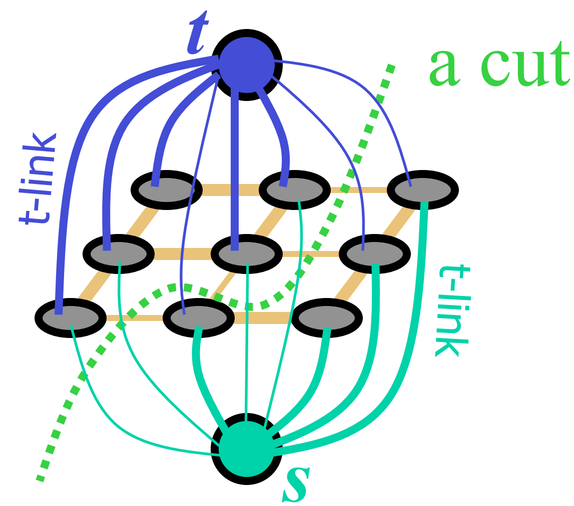
- Simple binary labeling problem
  - The sum of the assignment costs is the number of pixels that we think "flip"
  - The sum of the separation costs is the number of adjacent pixels that we think have different colors
  - Sometimes called "Ising" model



restored image

Solution via graph cuts

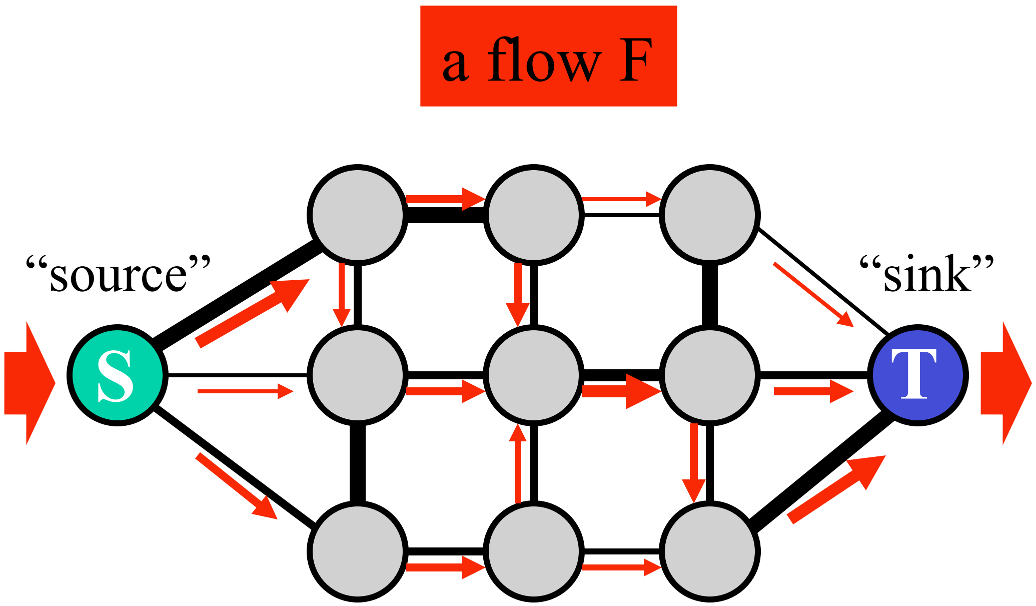
- Build the appropriate graph
  - Image pixels are nodes in the graph
    - Nearby pixels (nodes) connected by an edge, which we call an n-link
    - Terminal ***s*** is identified with label 0, and connected by edge we call a t-link with every image pixel
    - Terminal ***t*** is identified with label 1 and connected by t-link with every image pixel



  - A cut separates ***t*** from ***s***
    - Each pixel stays connected to either ***t*** or ***s*** (label 1 or 0)
    - Cuts correspond to labelings, and with right edge weights cost is same

Minimum cut gives the minimum energy labeling

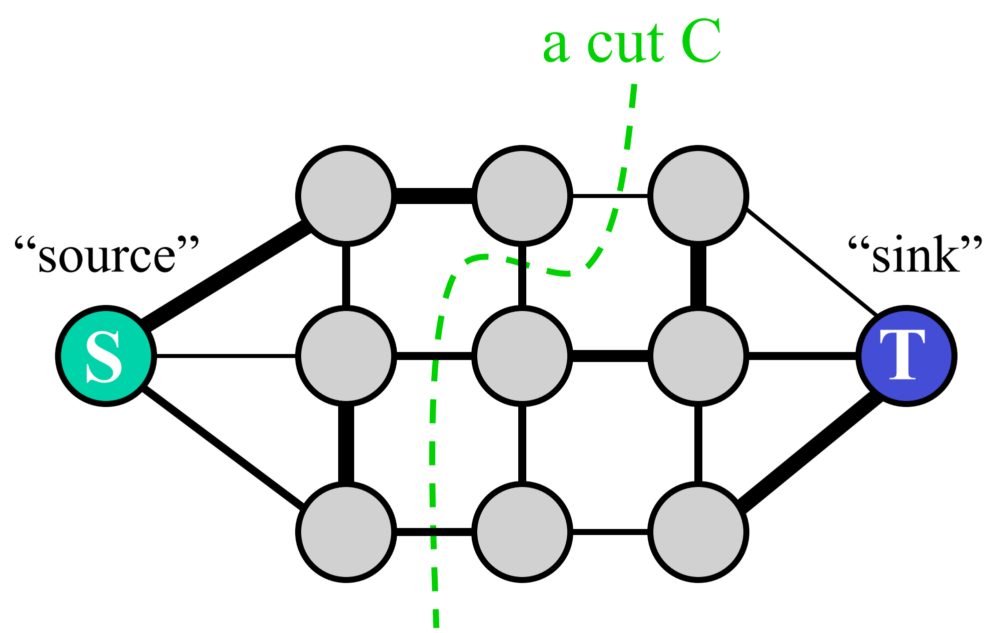
Maximum flow problem



A graph with two terminals

- Max flow problem:
  - Each edge is a "pipe"
  - Find the largest flow F of "water" that can be sent from the "source" to the "sink" along the pipes
  - Source output = sink input = flow value
  - Edge weights give the pipe's capacity

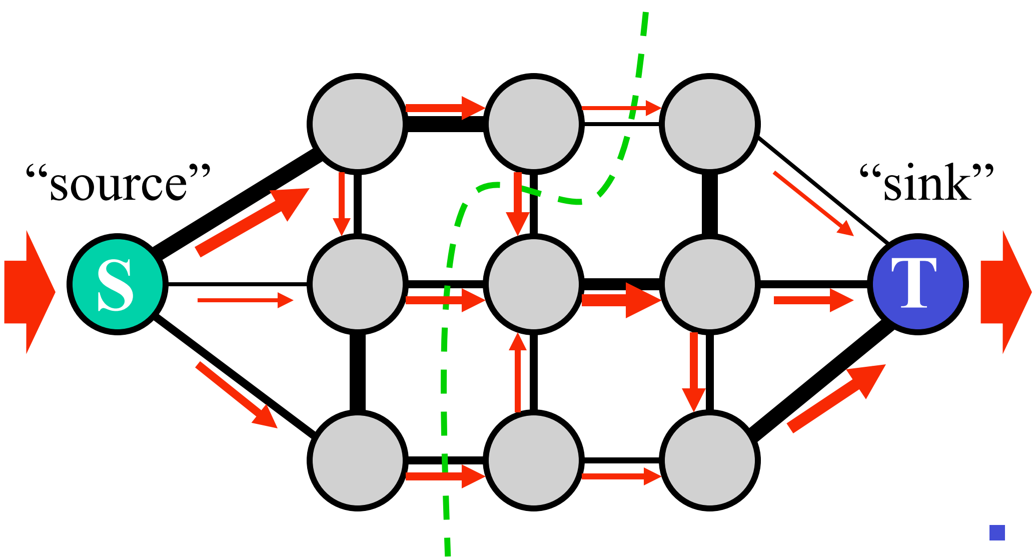
Minimum cut problem



A graph with two terminals

- Min cut problem:
  - Find the cheapest way to cut the edges so that the "source" is separated from the "sink"
  - Cut edges going from source side to sink side
  - Edge weights now represent cutting "costs"

Max flow/Min cut theorem



A graph with two terminals

- Max Flow = Min Cut:
  - Proof sketch: value of a flow is value over any cut
  - Maximum flow saturates the edges along the minimum cut
    - Ford and Fulkerson, 1962
    - Problem reduction!
- Ford and Fulkerson gave first polynomial time algorithm for globally optimal solution

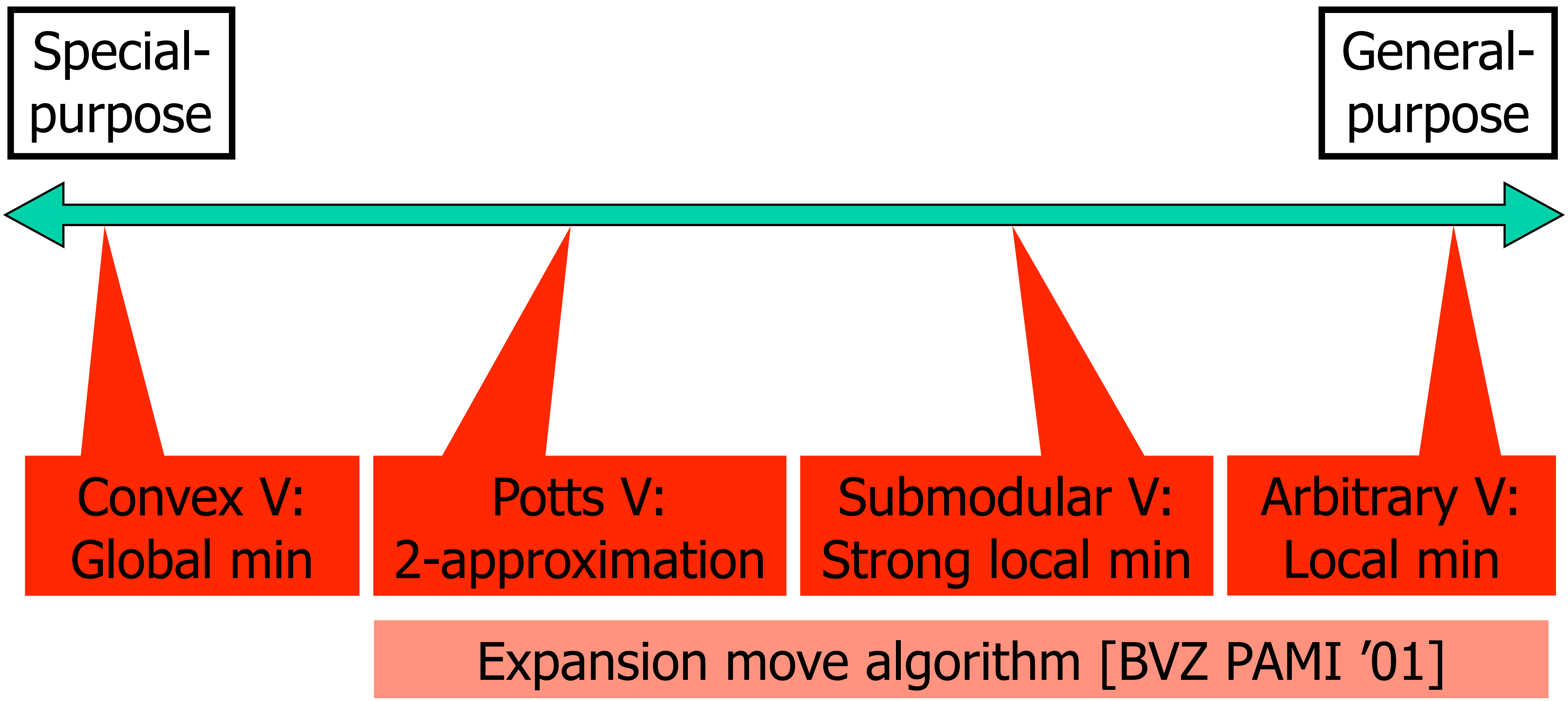
Fast algorithms for min cut

- Max flow problem can be solved fast
  - Many algorithms, such as augmenting paths
    - Find a path from S to T that does not go through any saturated edge
    - Push more flow through that path
- Most graph problems are intractable
  - Variants of min cut are NP-hard
- Example: multiway cut problem
  - More than 2 terminals
  - Find lowest cost edges separating them all

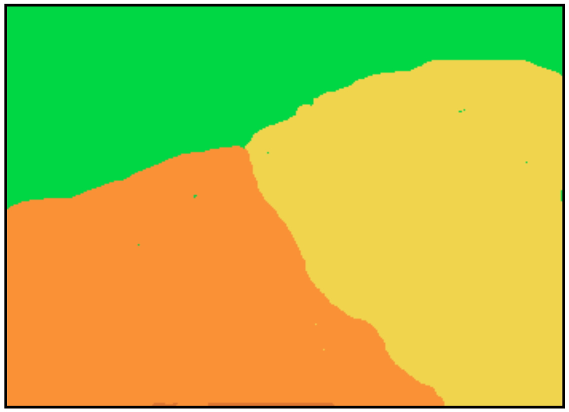
What do graph cuts provide?

- For less robust V, polynomial algorithm for global minimum!
  - Discrete version of TV, but with non-convex D
- For a particularly robust V, an approximation algorithm
  - Proof of NP hardness
- For many choices of V, algorithms that find a "strong" local minimum
- High quality experimental results
  - Within 1% of the global minimum on a wide range of benchmarks

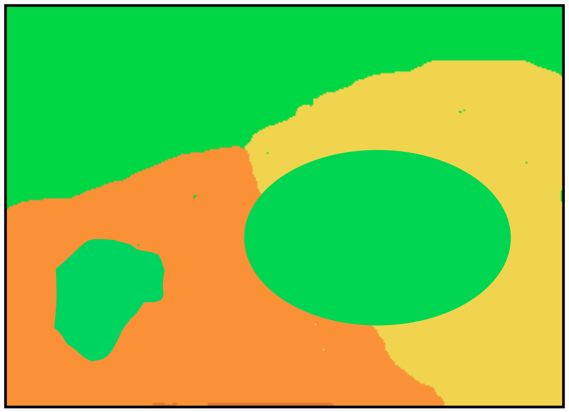
Spectrum of results



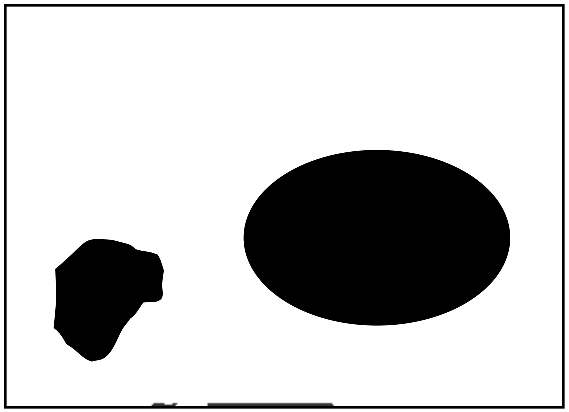
Binary sub-problem



Input labeling

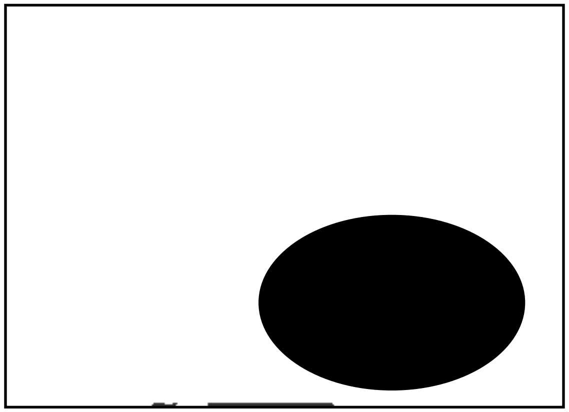


Expansion move



Binary image

Expansion move energy

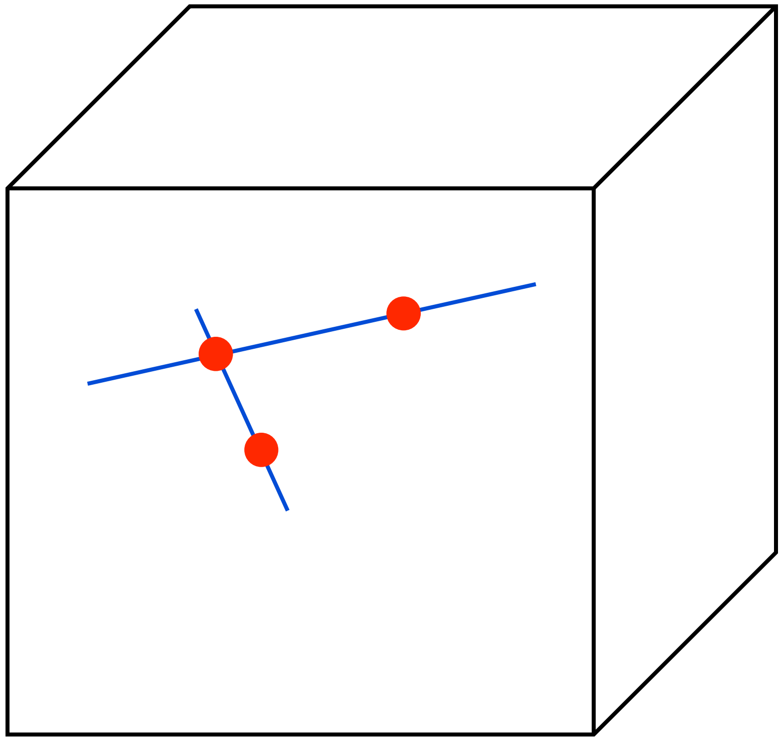


Goal: find the binary image with lowest energy

Binary image energy is a restricted version of original E

Depends on f, alpha

Local improvement methods

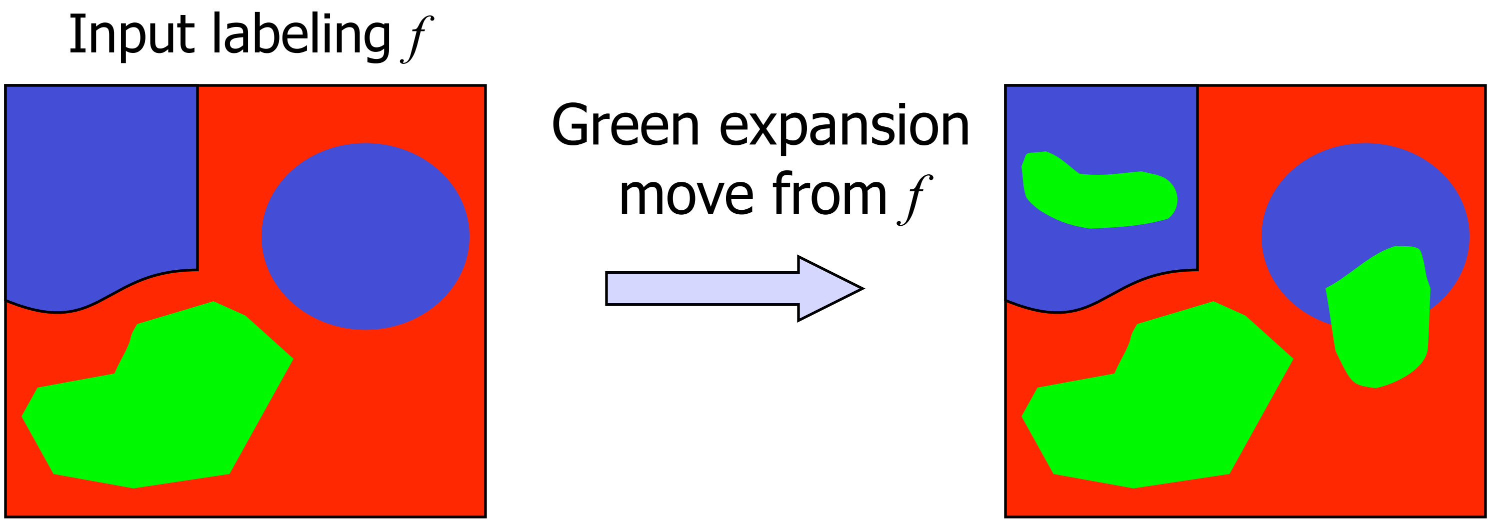


- Subproblem: locally minimize restricted version of E
- Ultimately computes a minimum w.r.t. any line

Local improvement vs. Graph cuts

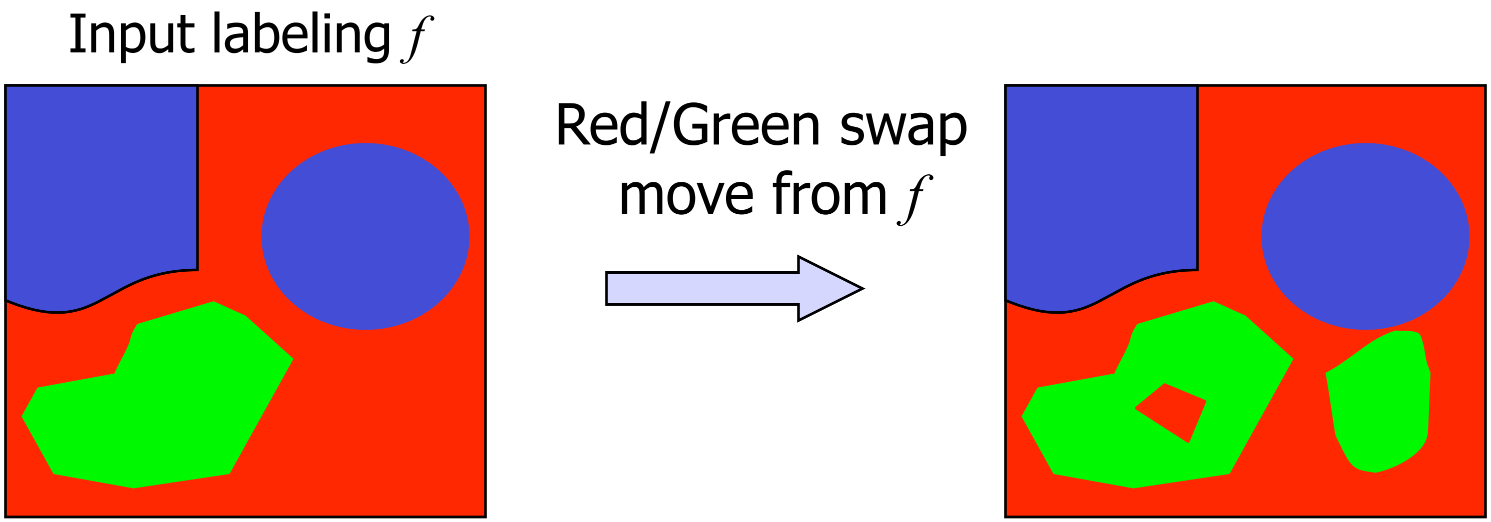
- Continuous vs. discrete
  - No floating point with graph cuts
- Local min in line search vs. global min
- Minimize over a line vs. hypersurface
  - Containing $O(2^n)$ candidates
- Local minimum: weak vs. strong
  - Theoretical guarantees concerning distance from global minimum
    - 2-approximation for a common choice of E
  - Within 1% of global min on benchmarks!

Expansion move algorithm



- Find green expansion move that most decreases E
  - Move there, then find the best blue expansion move, etc
  - Done when no alpha-expansion move decreases the energy, for any label alpha

Swap algorithm



- Find a swap of pixel labels that most decreases energy
  - Move there, then find the next best swap move, etc
  - Done when no swap move decreases the energy, for any pair of labels

Swap Optimization Alg.

1. Start with an arbitrary labeling f
2. Success = 0
3. For each label (resp. pair)
    1. Find f* = arg min E*(f’) within one alpha -expansion (resp. alpha-beta swap) of f
    2. If (E(f*) > E(f)) set f = f* and success = 1
4. If success = 1, repeat from 2
5. Return f

Finding the Optimal Swap

- Construct a subgraph just on the pair of labels
- Compute special energies for the t nodes
  - t-alpha = data(alpha) + regularization over nodes on in graph
  - t-beta = data(beta) + regularization over nodes not in graph
  - Pixel connection = regularization V(alpha,beta)
- Compute a cut
- Assign new labels to pixels in subgraph

Sample results

Right answers

Dynamic programming
Correlation
Graph cuts

Expansion moves in action

initial solution

-expansion

-expansion

-expansion

-expansion

-expansion

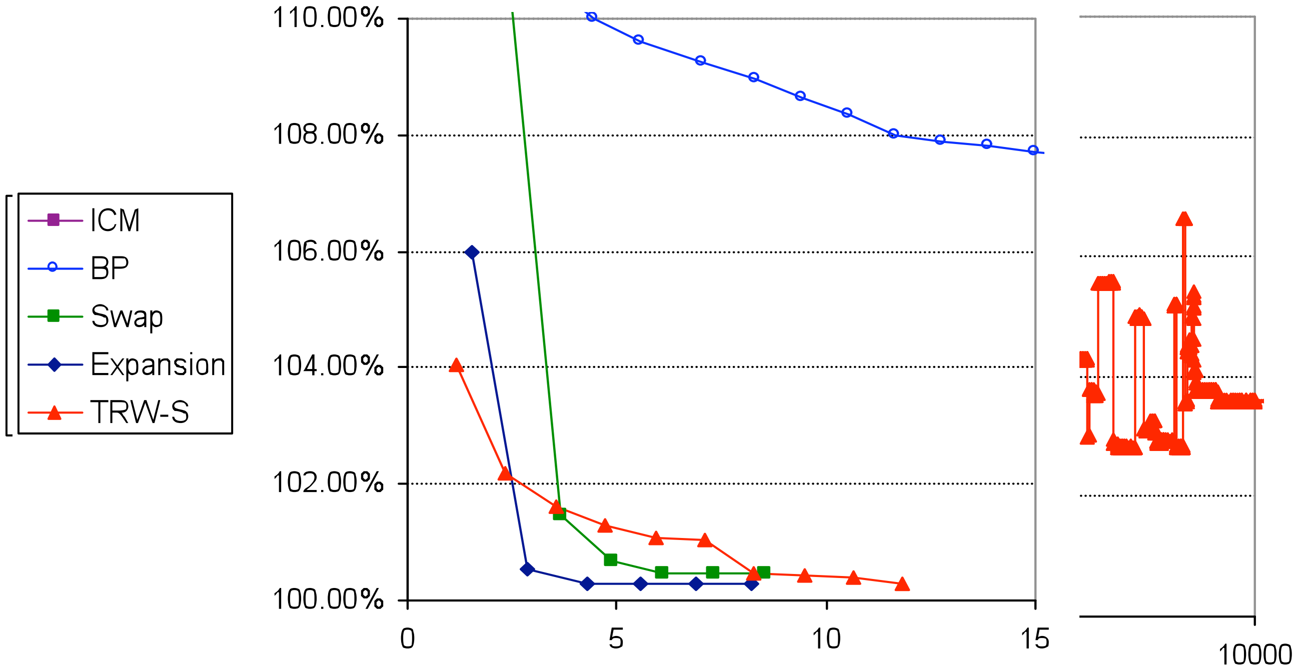
-expansion

-expansion

Must choose expansion that gives the largest decrease in energy:
binary energy minimization subproblem

Theoretical and experimental properties of the expansion move algorithm

Experimental performance



Easy problem (Tsukuba)

Hard problem (Photomontage)

Summary

- Discrete optimization methods like graph cuts can be very powerful
- High quality solutions for non-convex optimization problems in thousand of dimensions
- Strong experimental results
- Ties to many branches of applied math

Acknowledgements

- Major ideas
  - Basic construction: Hammer '65
  - Binary application: Greig, Porteus & Seheult '86
  - Convex application: Ishikawa '03
  - Expansion moves: Boykov, Veksler & Zabih '01
  - Regularity: Kolmogorov & Zabih '04
- Slides from:
  - Aseem Agarwala, Yuri Boykov, Vladimir Kolmogorov, Carsten Rother, Olga Veksler