Approximate decoding: ICM, block methods, alpha-beta swap & alpha-expansion

Julieta Martinez

University of British Columbia

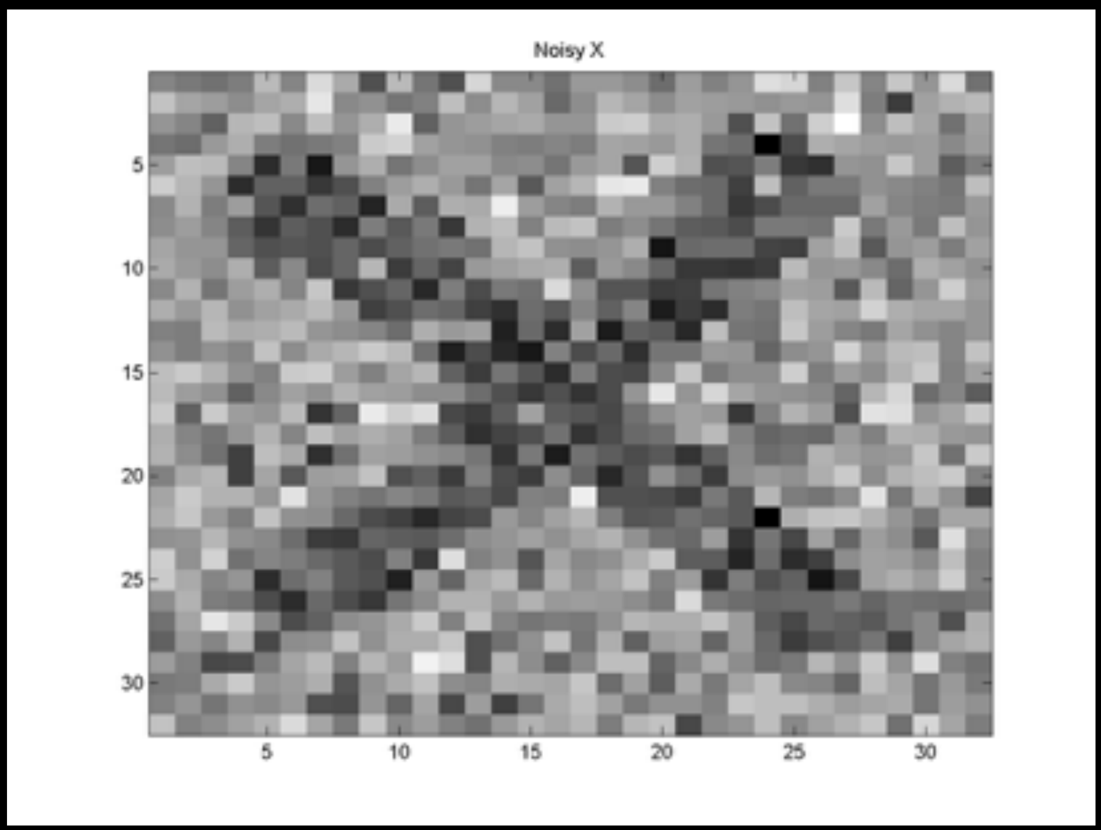
August 25, 2015

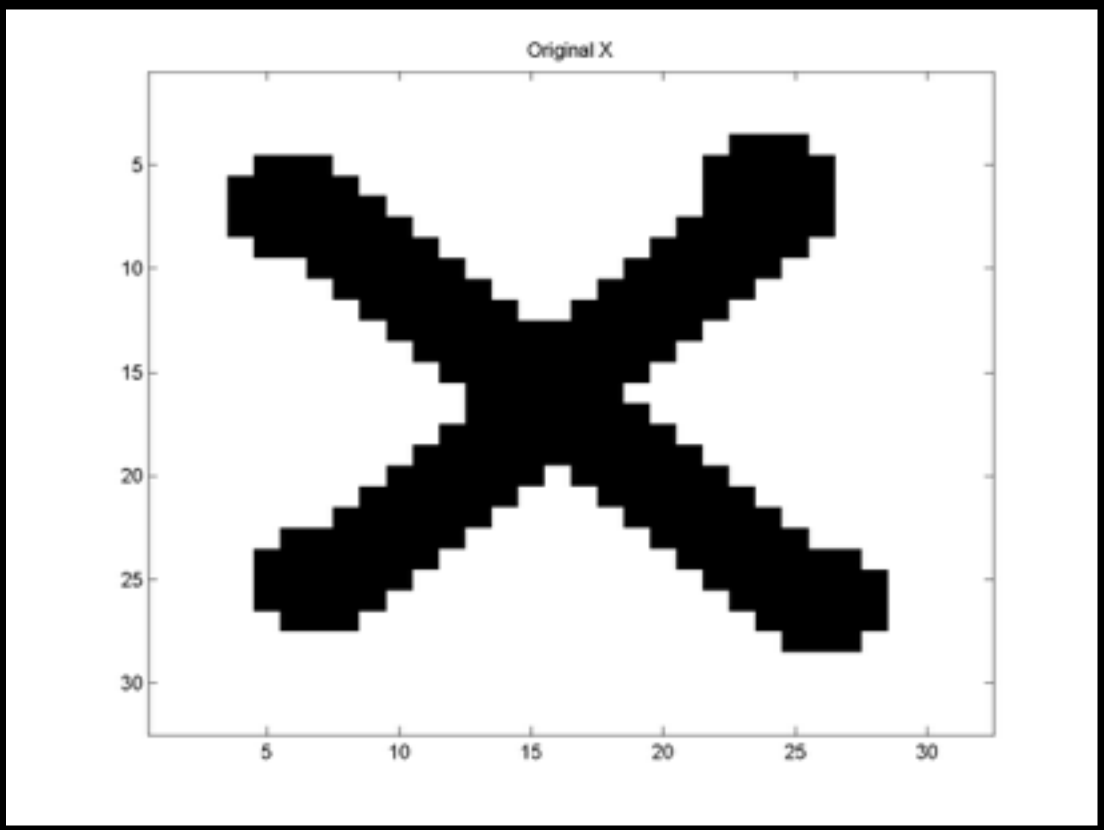
https://www.youtube.com/watch?v=kp3ik5f3-2c&t=18m36s

Outline

- Motivation
- Approximate decoding
- ICM: Iterated conditional modes
- Block methods
- Alpha-beta swap & alpha-expansion

Motivation

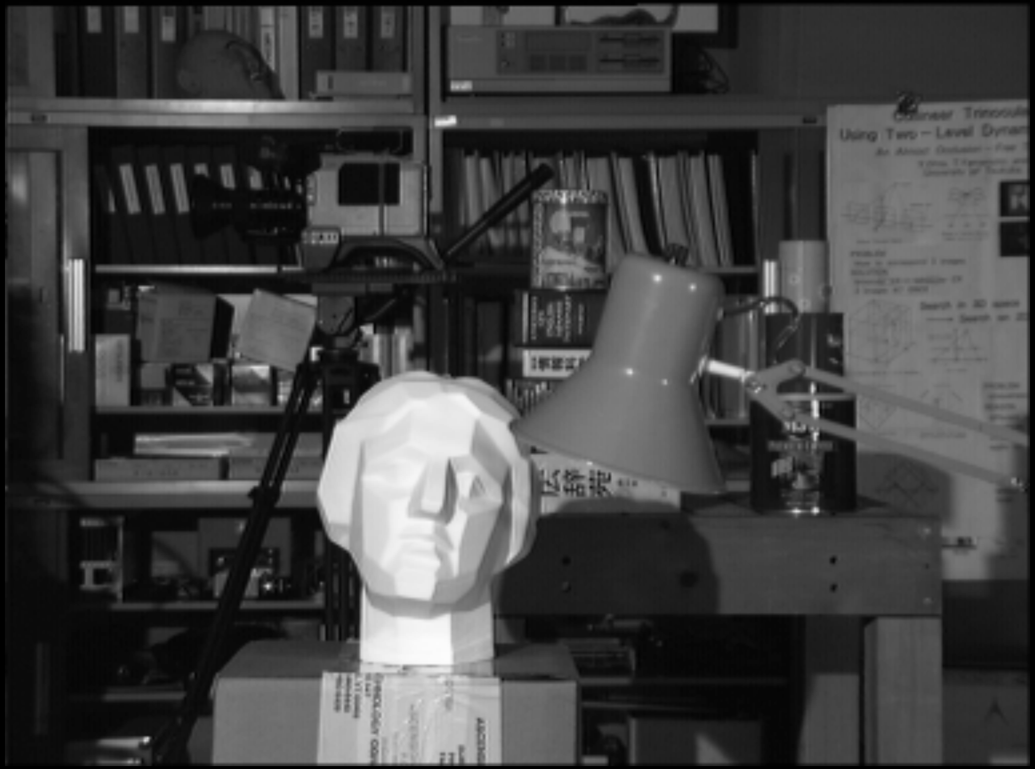

Noisy X
5
10
15
20
25
30
5
10
15
20
25
30

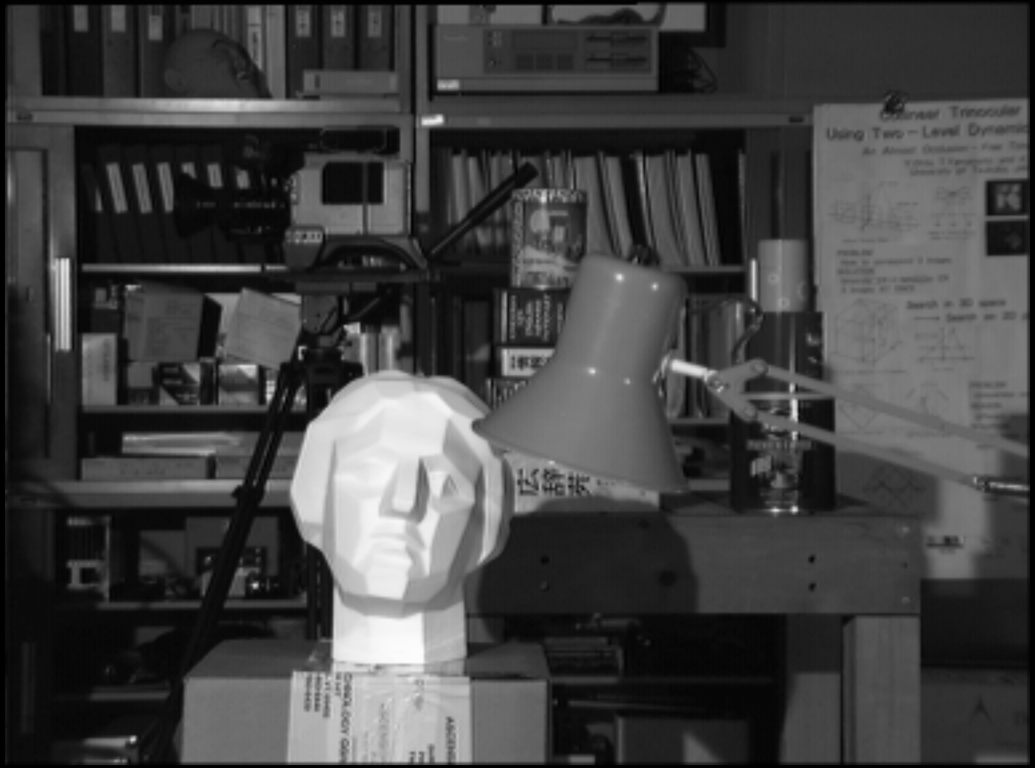

Original X
5
10
15
20
25
30
5
10
15
20
25
30

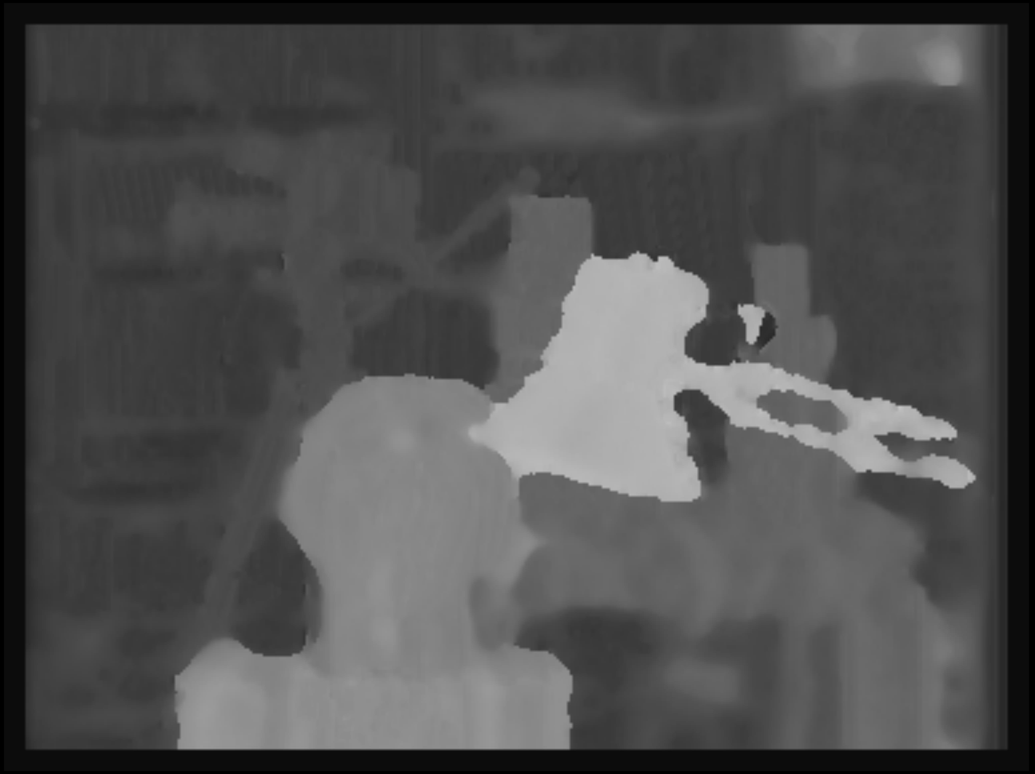
 

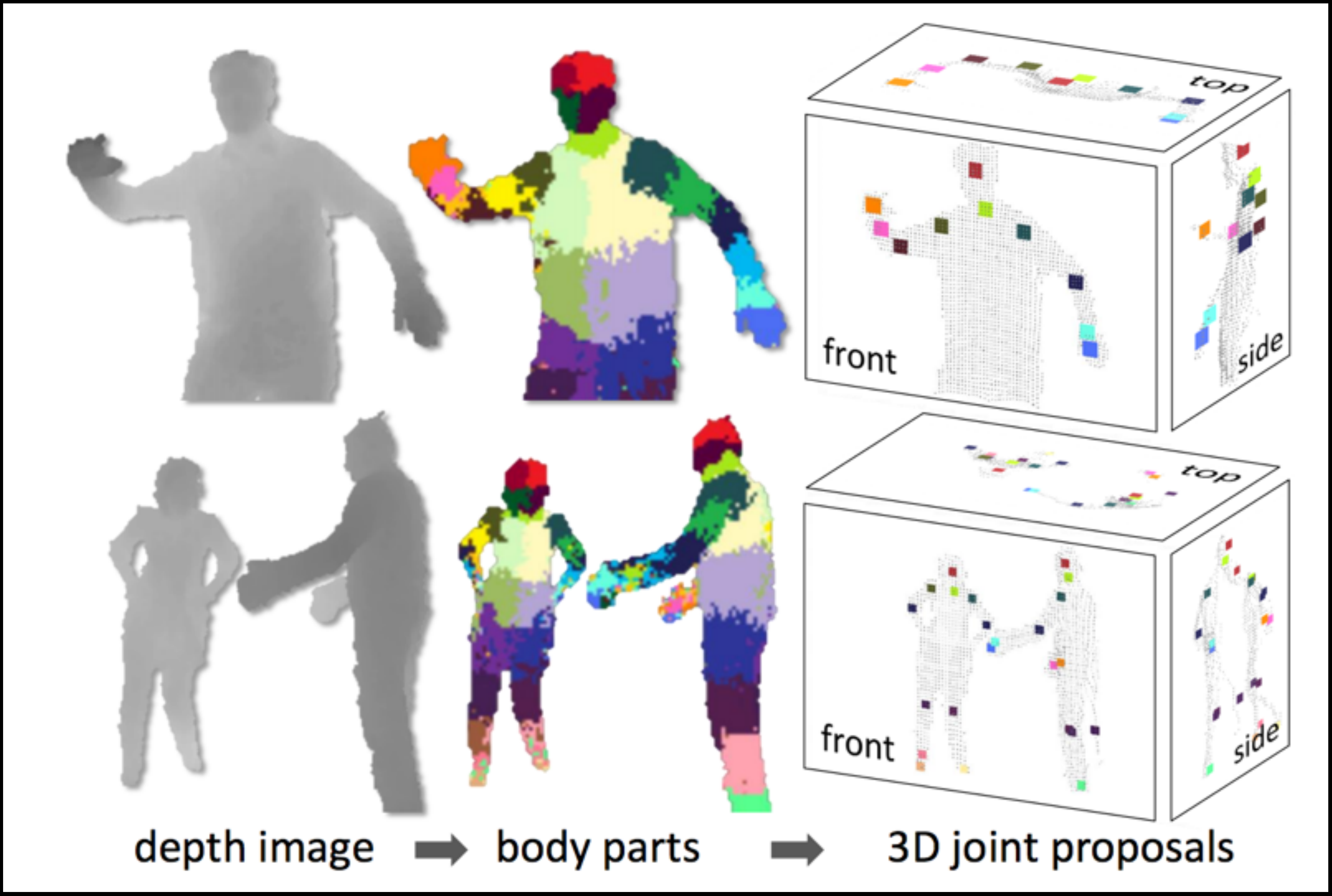
Photography by Dina Boyer: https://flic.kr/p/n9kdpQ

Tung, Frederick, and James J. Little. "Collageparsing: Nonparametric scene parsing by adaptive overlapping windows." Computer Vision–ECCV 2014. Springer International Publishing, 2014. 511-525.



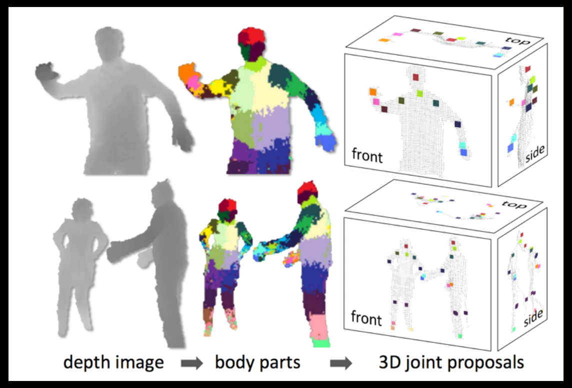




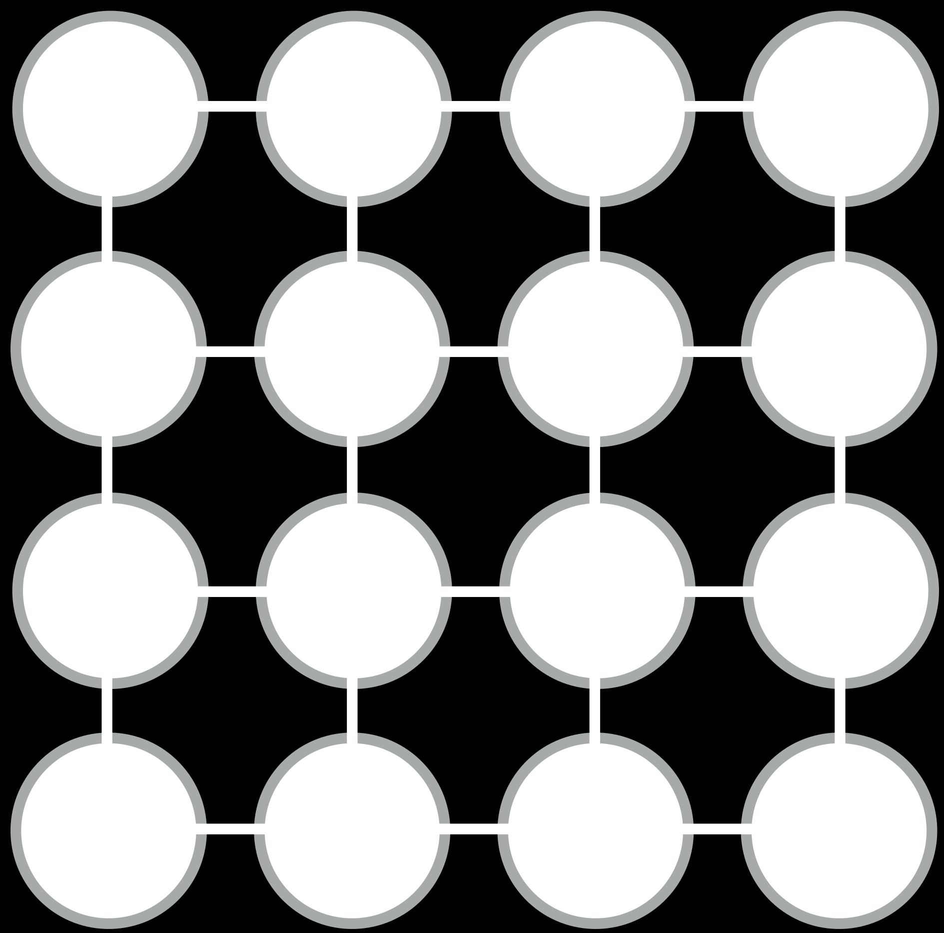

top
front
side
top
front
side
depth image
body parts
3D joint proposals

Motivation

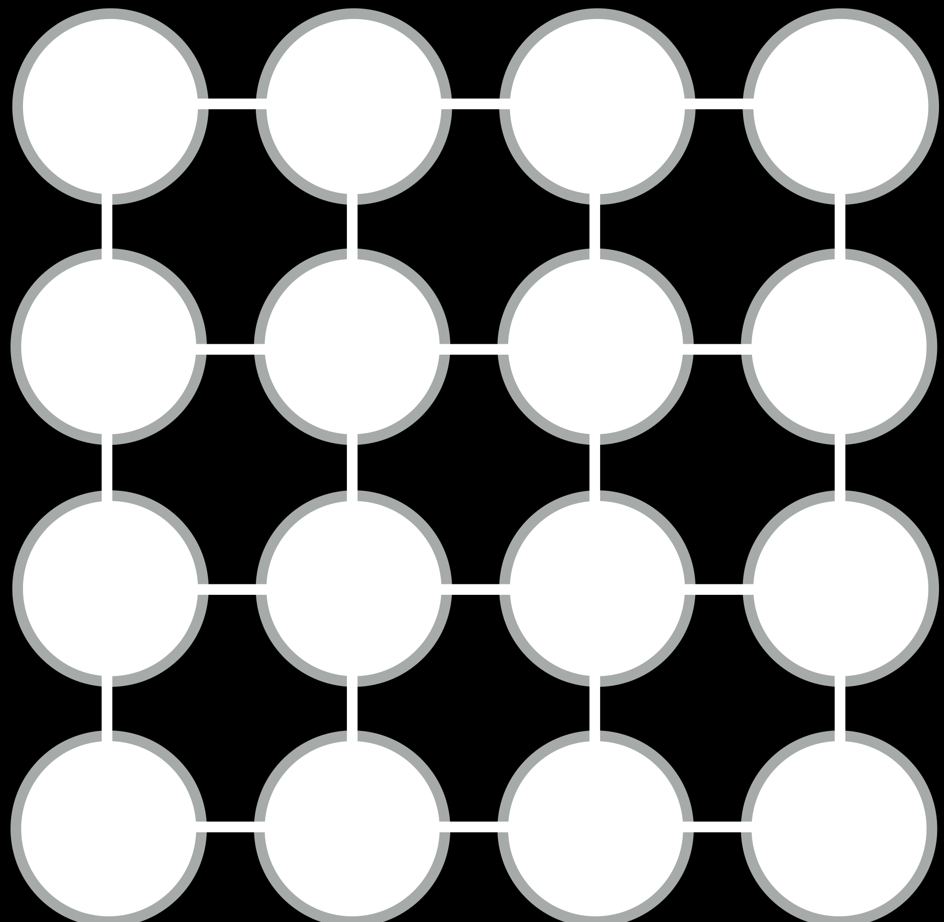
- We often have more than 2 states for each node
- Often the states are not directly comparable
- Exact decoding is NP-hard — but we still want real-time predictions



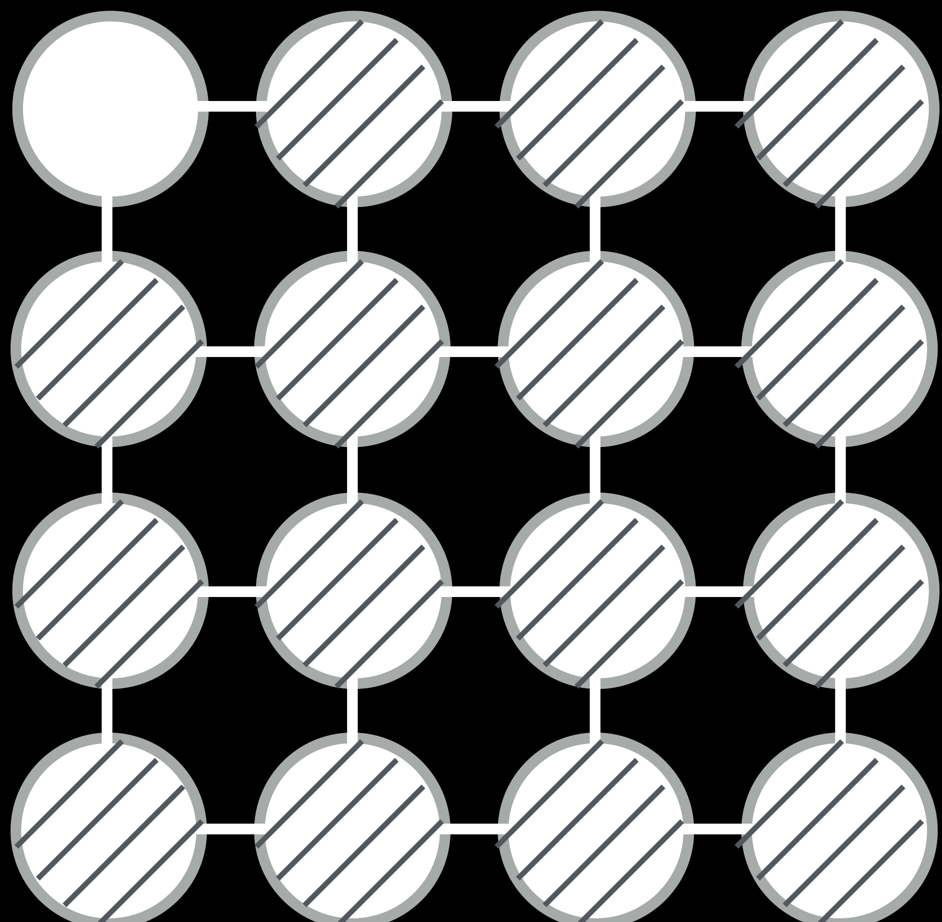
ICM: Iterated Conditional Modes



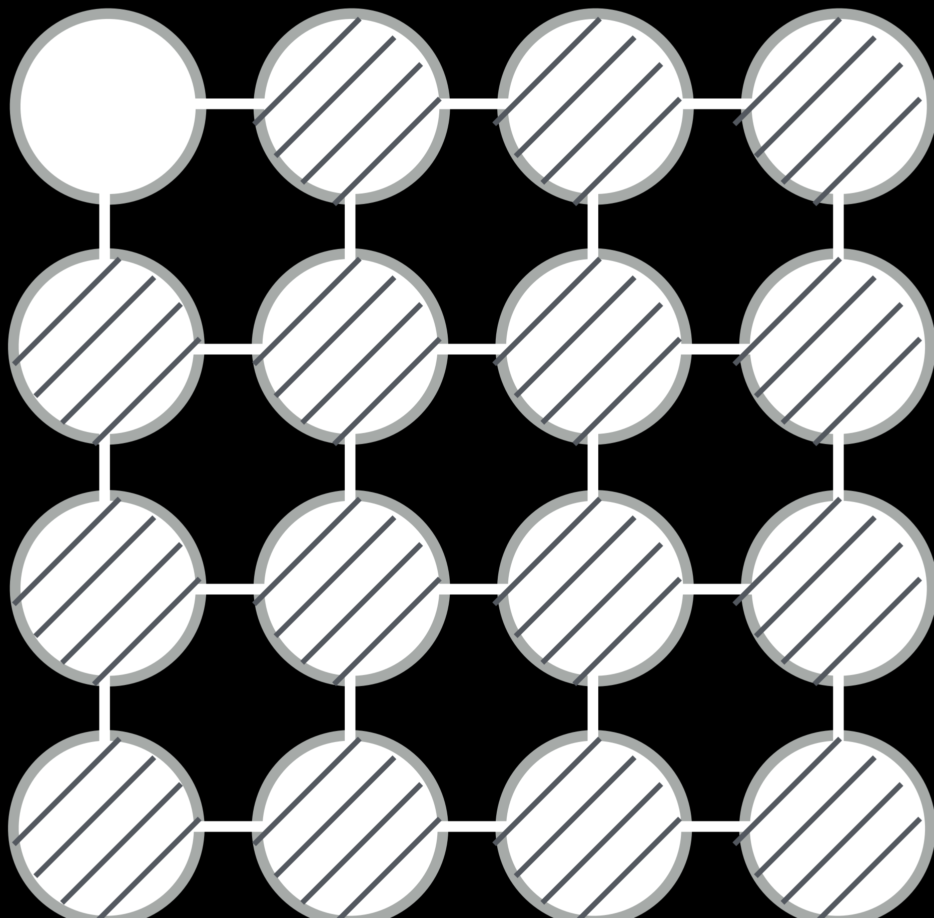
- Until convergence



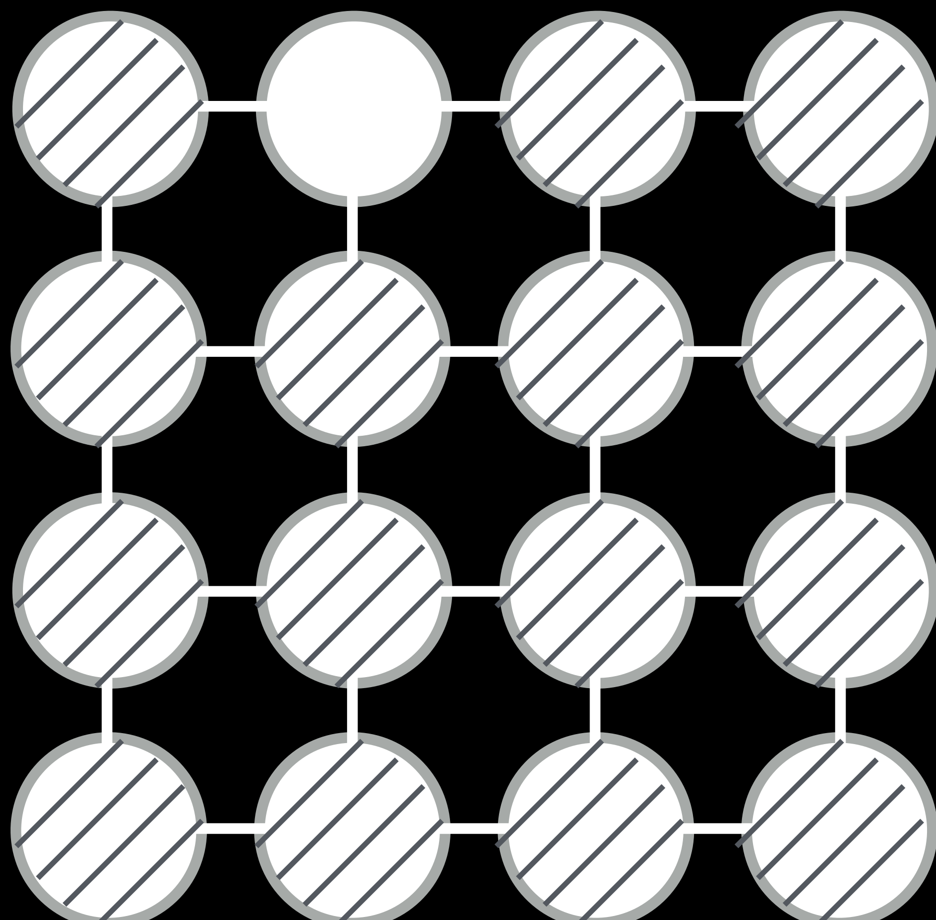
- Until convergence
  - For $x_i \in \mathbf{X}$



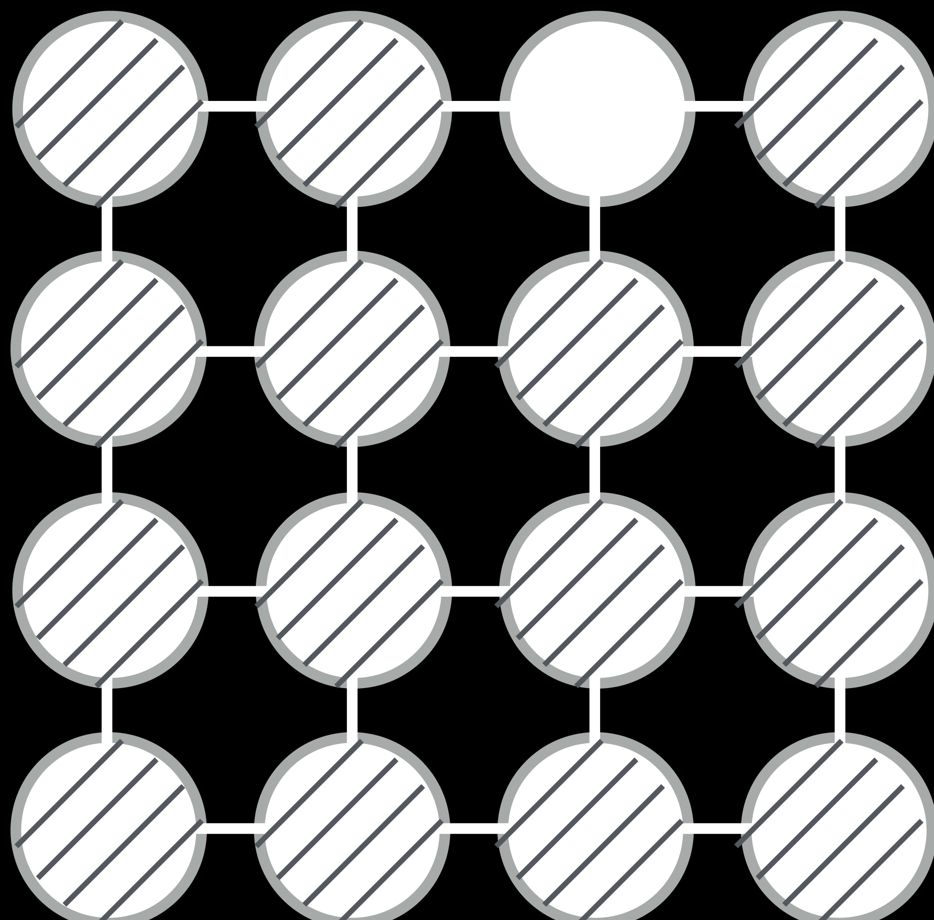
- Until convergence
  - For $x_i \in \mathbf{X}$
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed



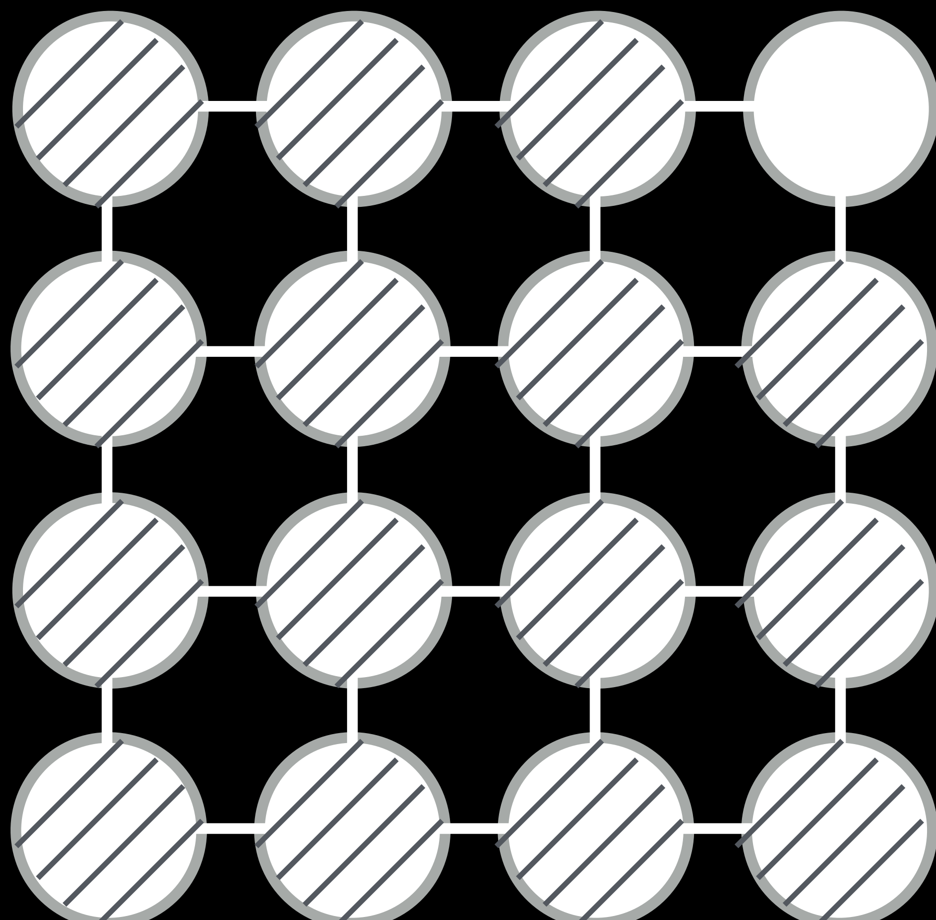
- Until convergence
  - For $x_i \in \mathbf{X}$
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed
    - Solve for x_i



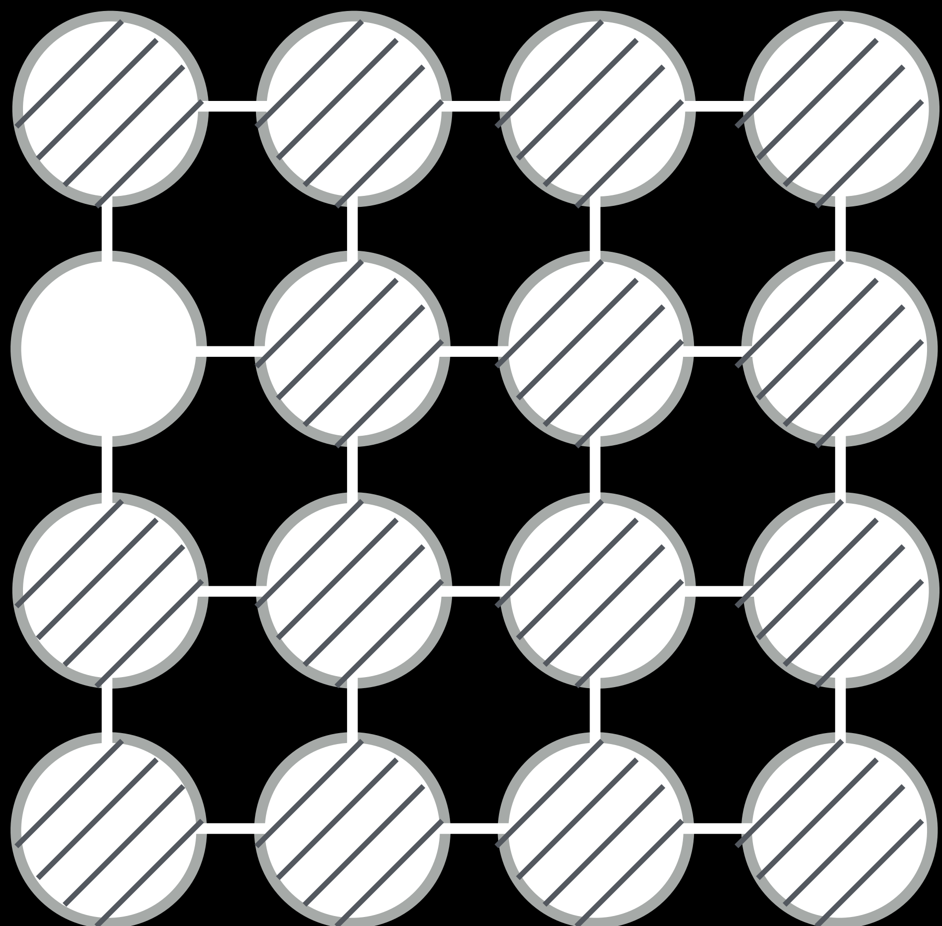
- Until convergence
  - For $x_i \in \mathbf{X}$
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed
    - Solve for x_i



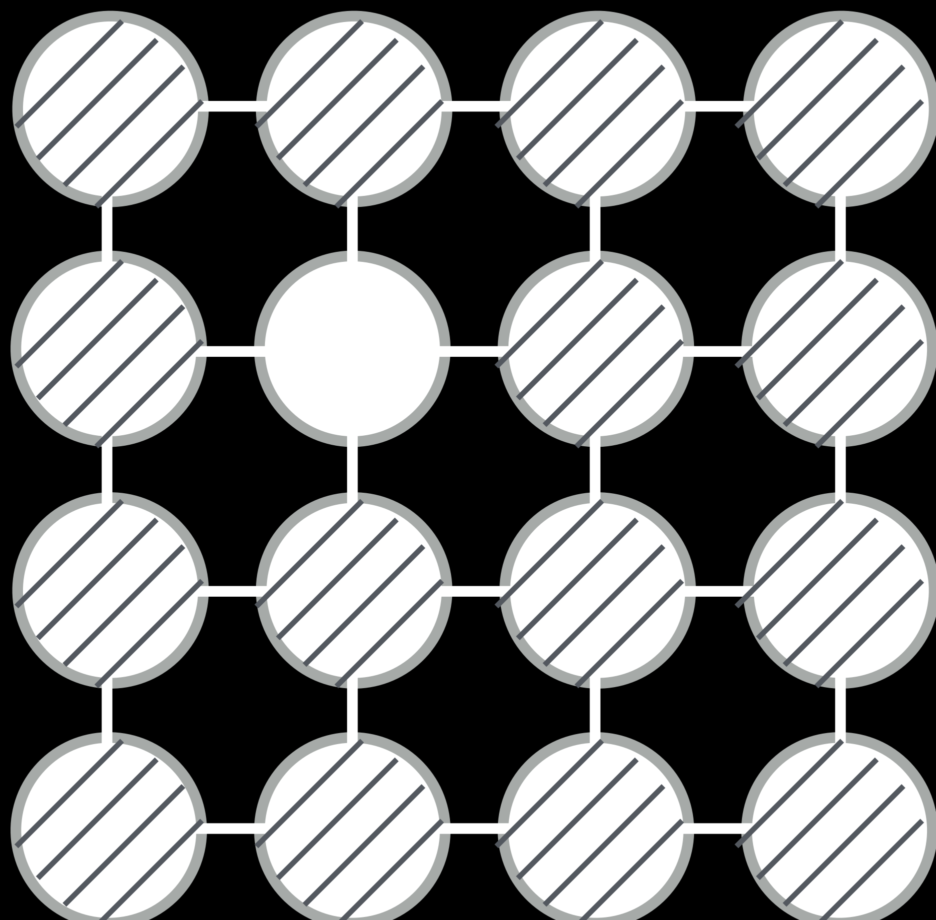
- Until convergence
  - For $x_i \in \mathbf{X}$
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed
    - Solve for x_i



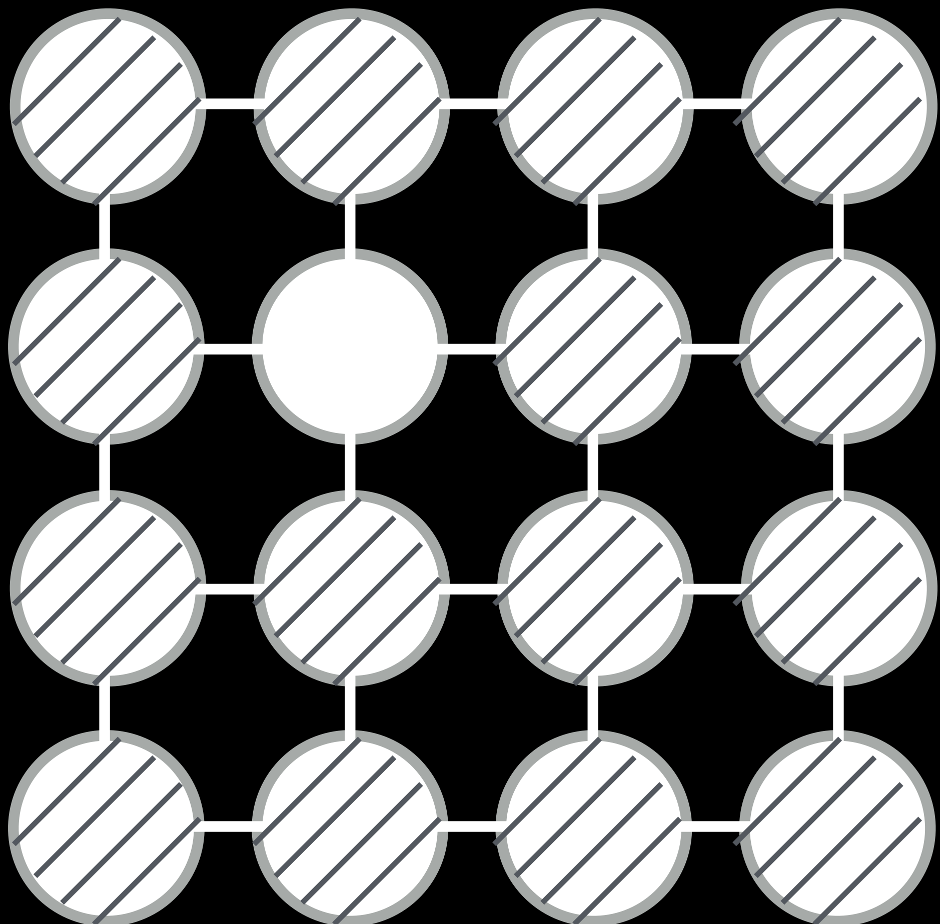
- Until convergence
  - For $x_i \in \mathbf{X}$
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed
    - Solve for x_i



- Until convergence
  - For $x_i \in \mathbf{X}$
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed
    - Solve for x_i



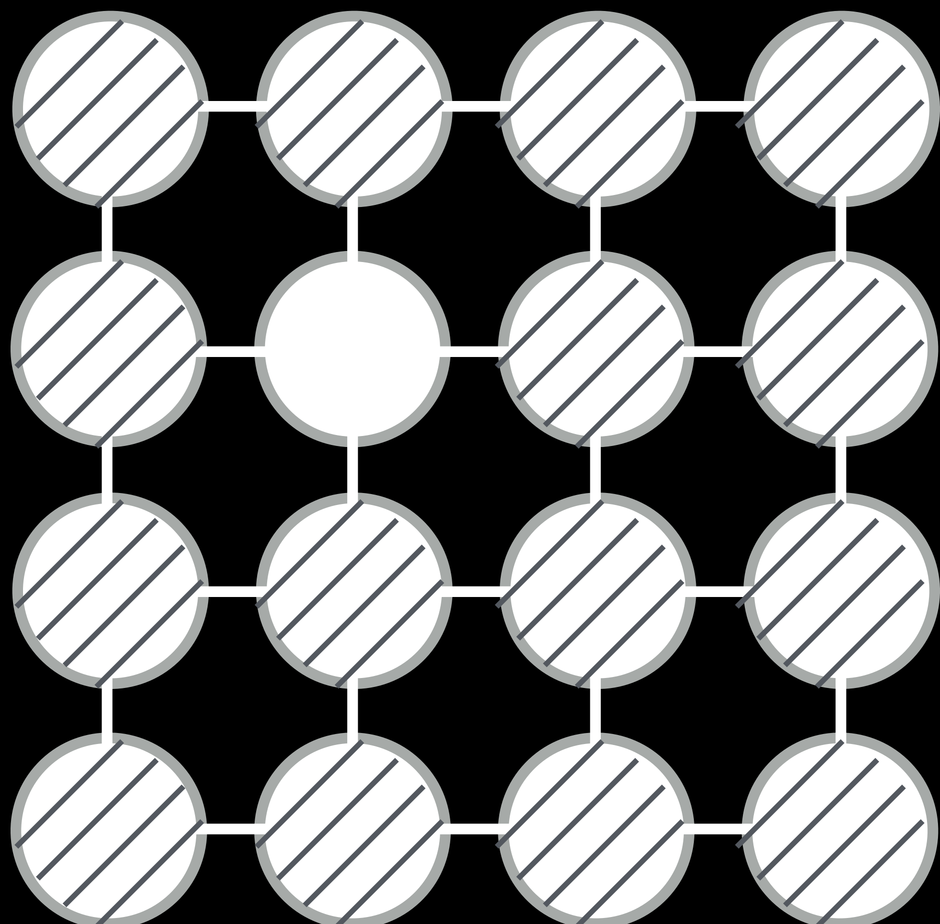
- Until convergence
  - For $x_i \in \mathbf{X}$
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed
    - Solve for x_i



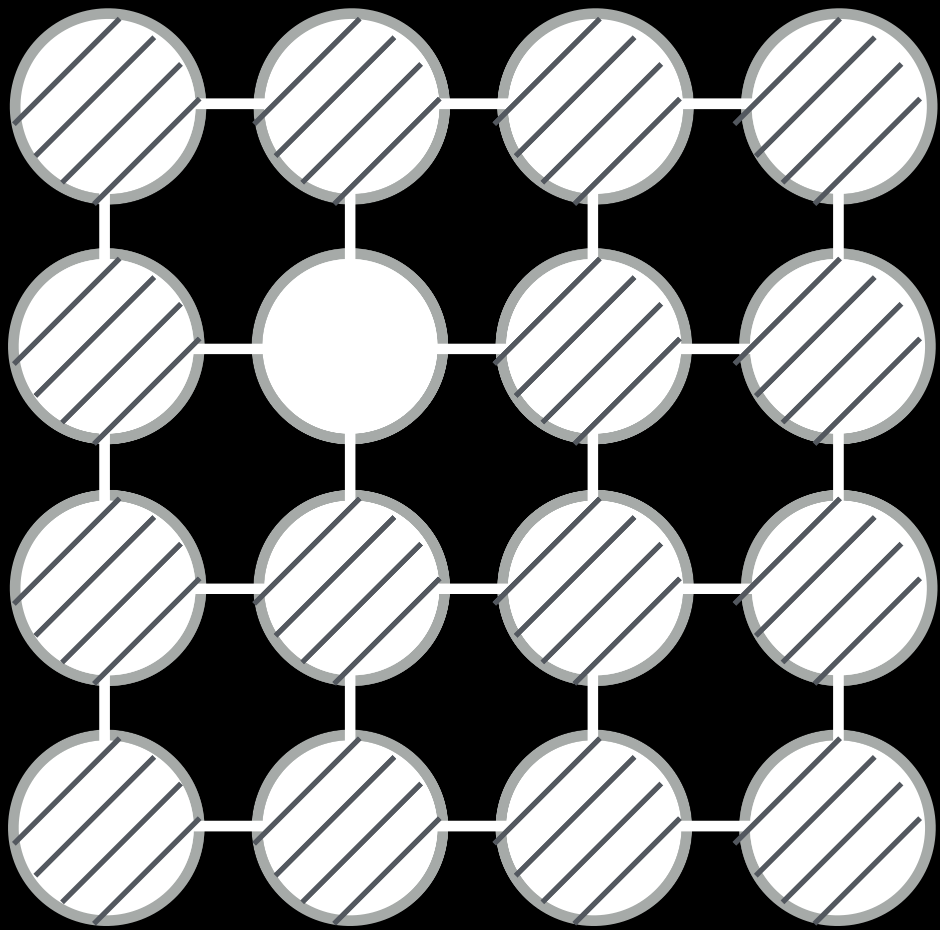
...and so on

- Until convergence
  - For $x_i \in \mathbf{X}$
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed
    - Solve for x_i

- Until convergence
  - For $x_i \in \mathbf{X}$ iterated
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed
    - Solve for x_i

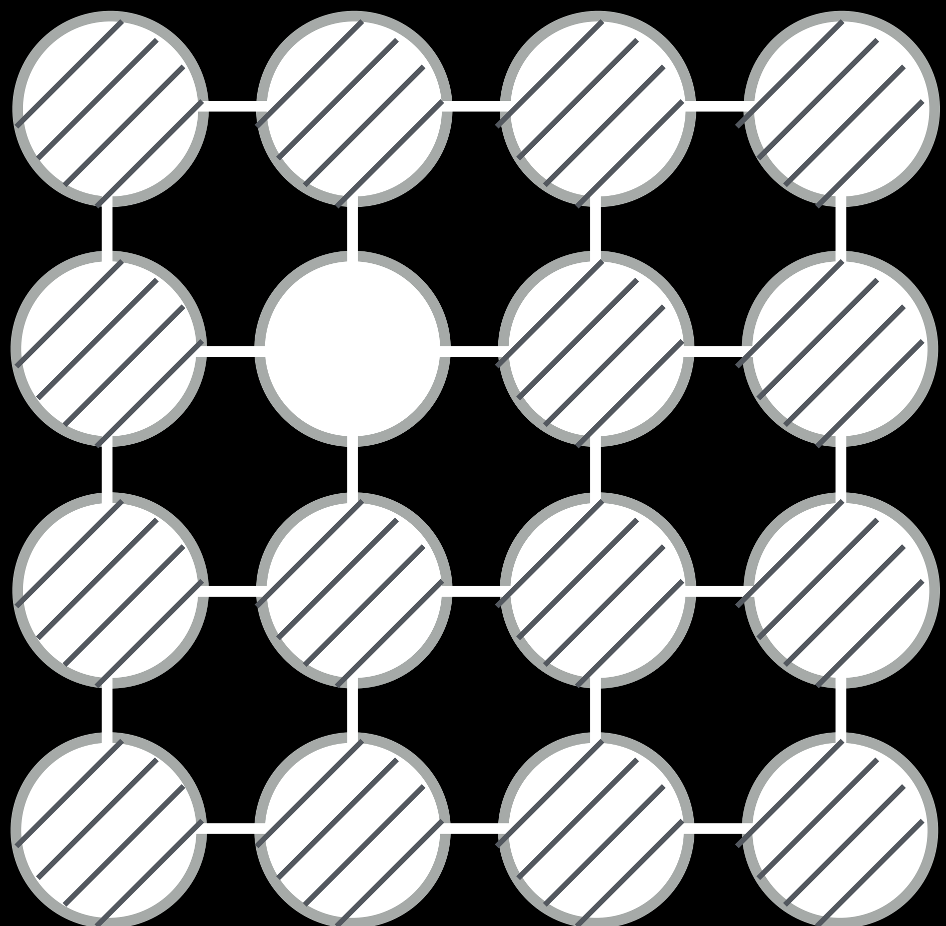


...and so on



...and so on

- Until convergence
  - For $x_i \in \mathbf{X}$ iterated
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed conditional
    - Solve for x_i



...and so on

- Until convergence
  - For $x_i \in \mathbf{X}$ iterated
    - Assume that all other nodes $x_j \in \mathbf{X}, j \neq i$ are fixed conditional
    - Solve for x_i modes

Pros and cons of ICM

- Pros
  - Super fast
  - Super easy to implement
- Cons
  - Greedy — will get stuck on local minima
  - In practice, local minima will not be very good

Improving ICM

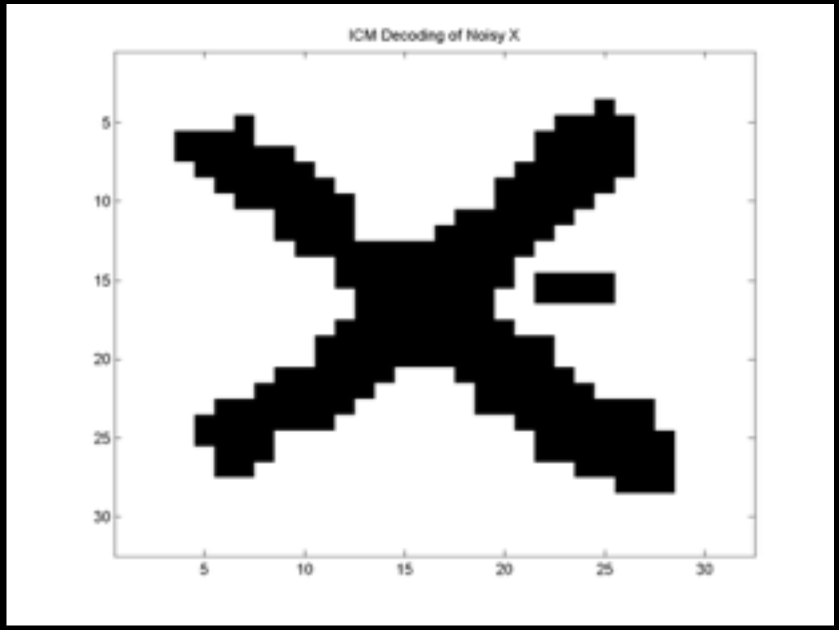
Guaranteed to find global minimum if done enough times

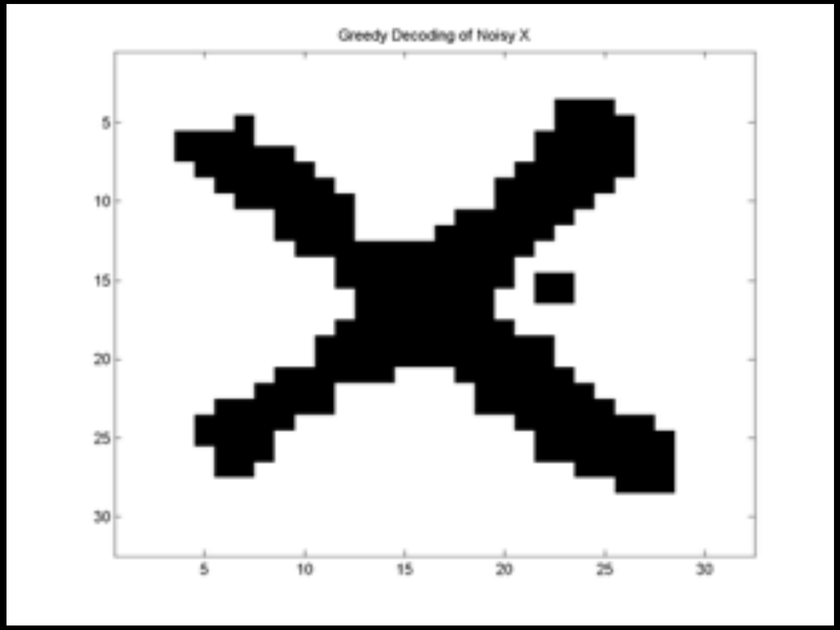
- *Restart* with different initializations

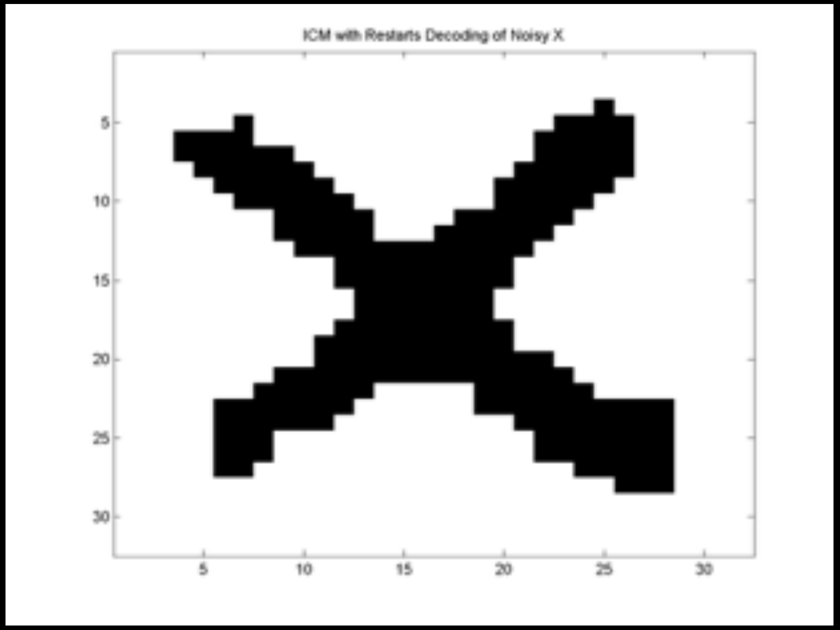
- Look at all the nodes, and update only the one that gives the *best improvement*

Improving ICM

ICM with restarts



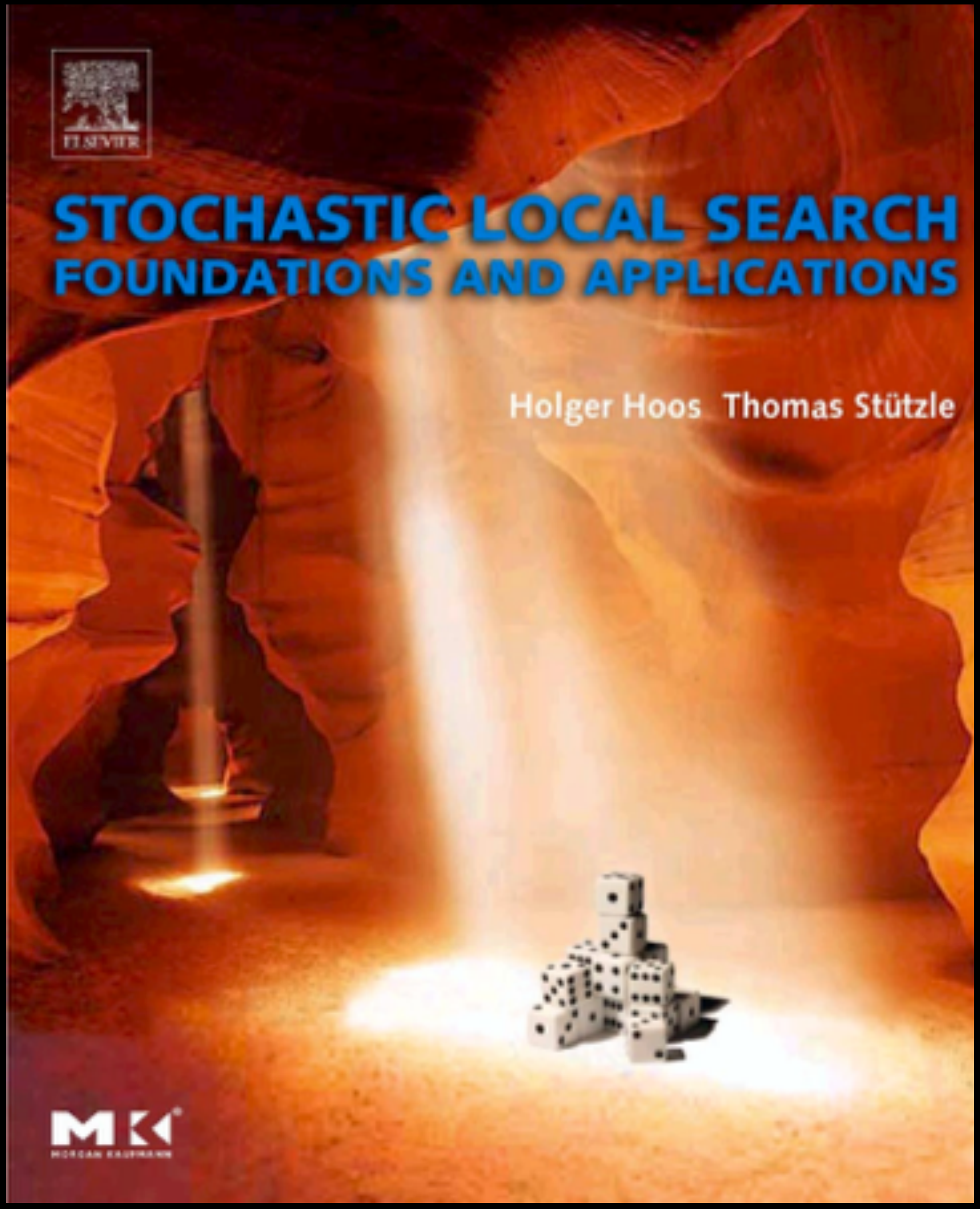




Vanilla ICM

ICM with best improvement

Other variants of local search



- Local search is an area of research on its own
- Usually a top performer on SAT, TSP, scheduling, and other NP-hard problems
- Read the book of my supervisor ☺

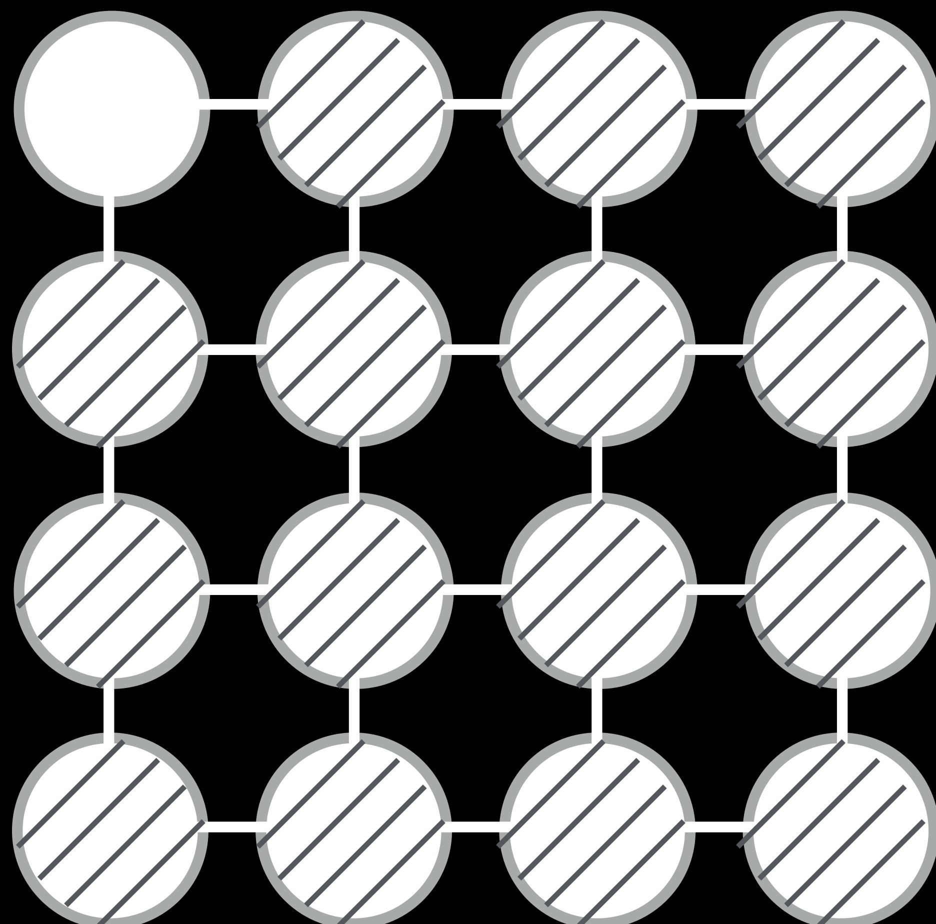
Block methods

Block ICM

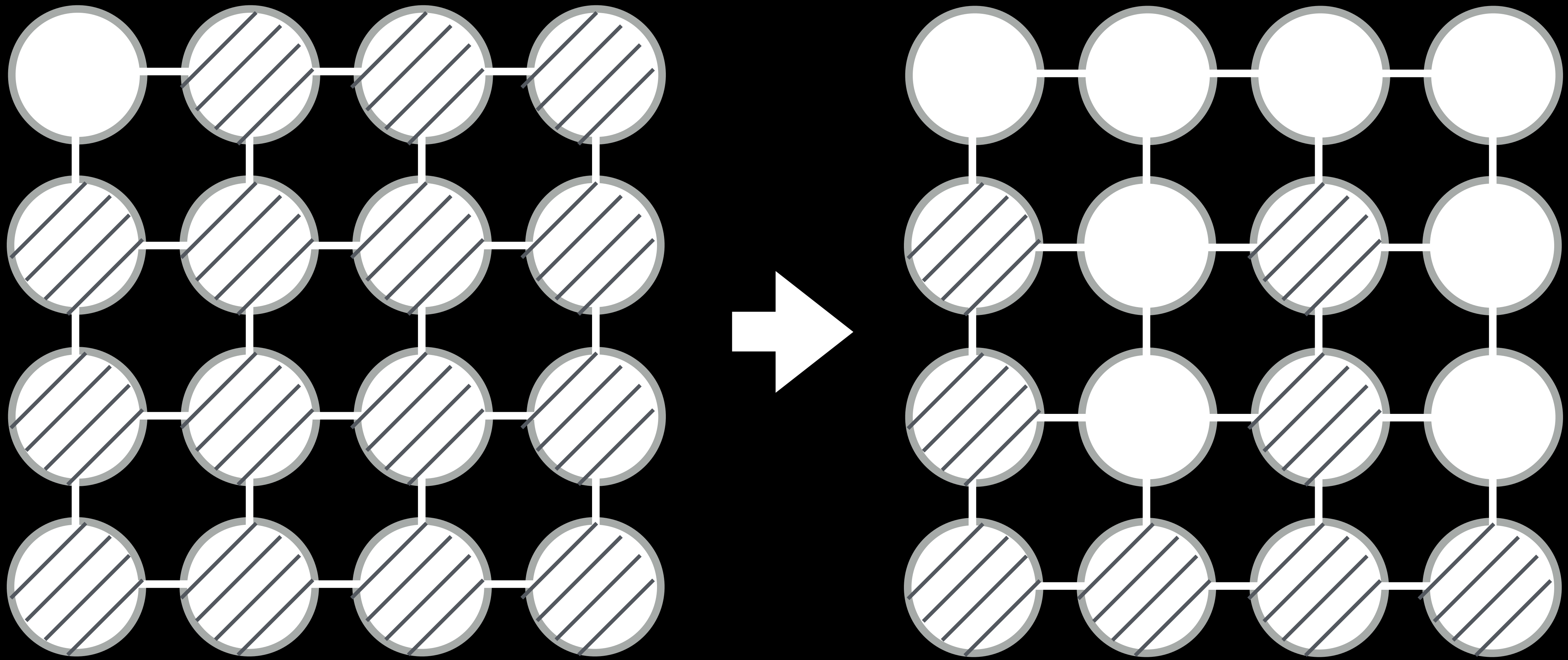
- In ICM we updated one node at a time
- Simple extension: update more than one node at a time
- In fact, we now have a toolbox to do exact decoding in large graphs

as long as they have a nice graph structure

Block ICM

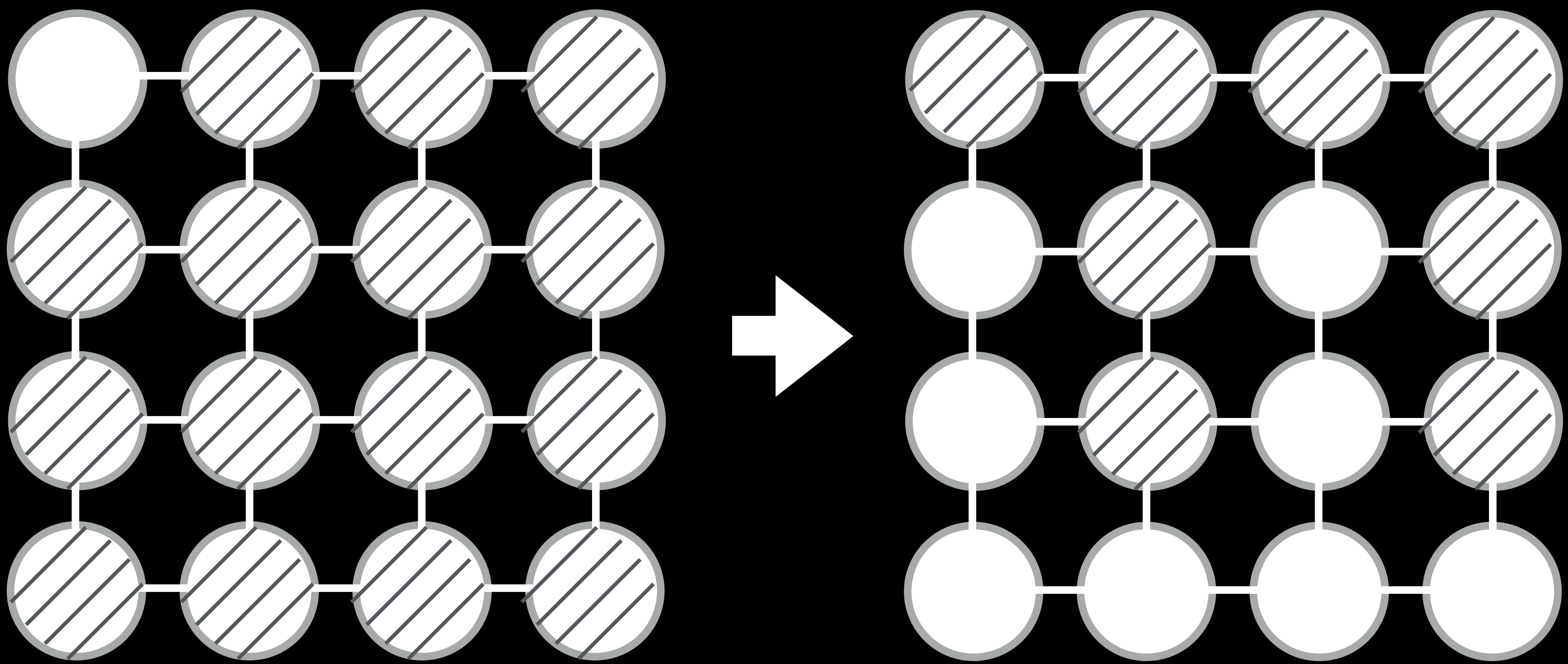


Block ICM



decode optimally

Block ICM



decode optimally too

Other block methods

- Many methods have this 1-variable, >1-variables generalization

Mark will cover this next Monday

- **Inference**: mean field updates the marginal of 1 variable at a time — we can update the marginals of >1 variables at a time

- **Sampling**: Gibbs sampling samples 1 variable at a time — we can sample a block of >1 variables at a time

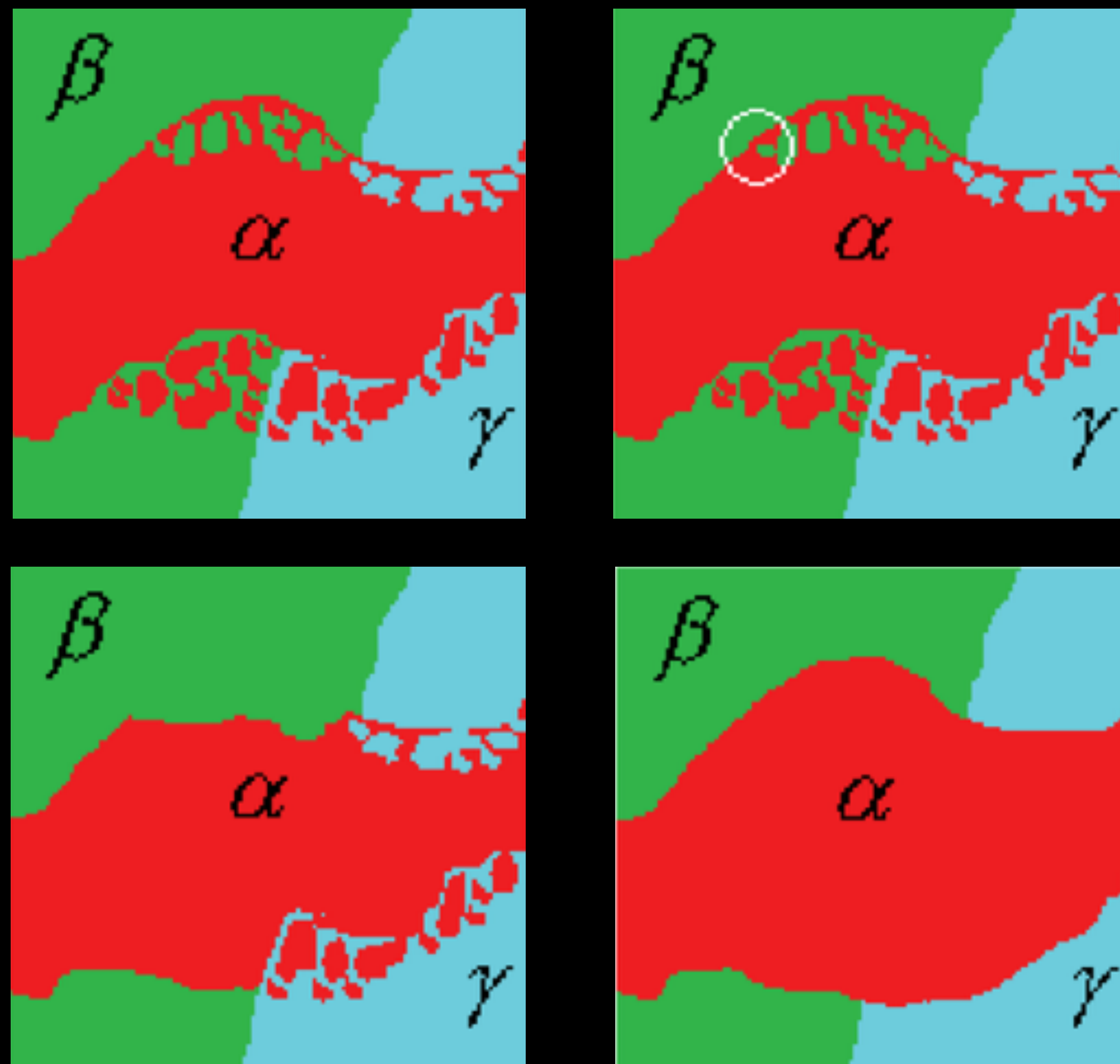
Jason will cover this tomorrow

Alpha-beta swap & alpha-expansion

An overview

original labelling

ICM



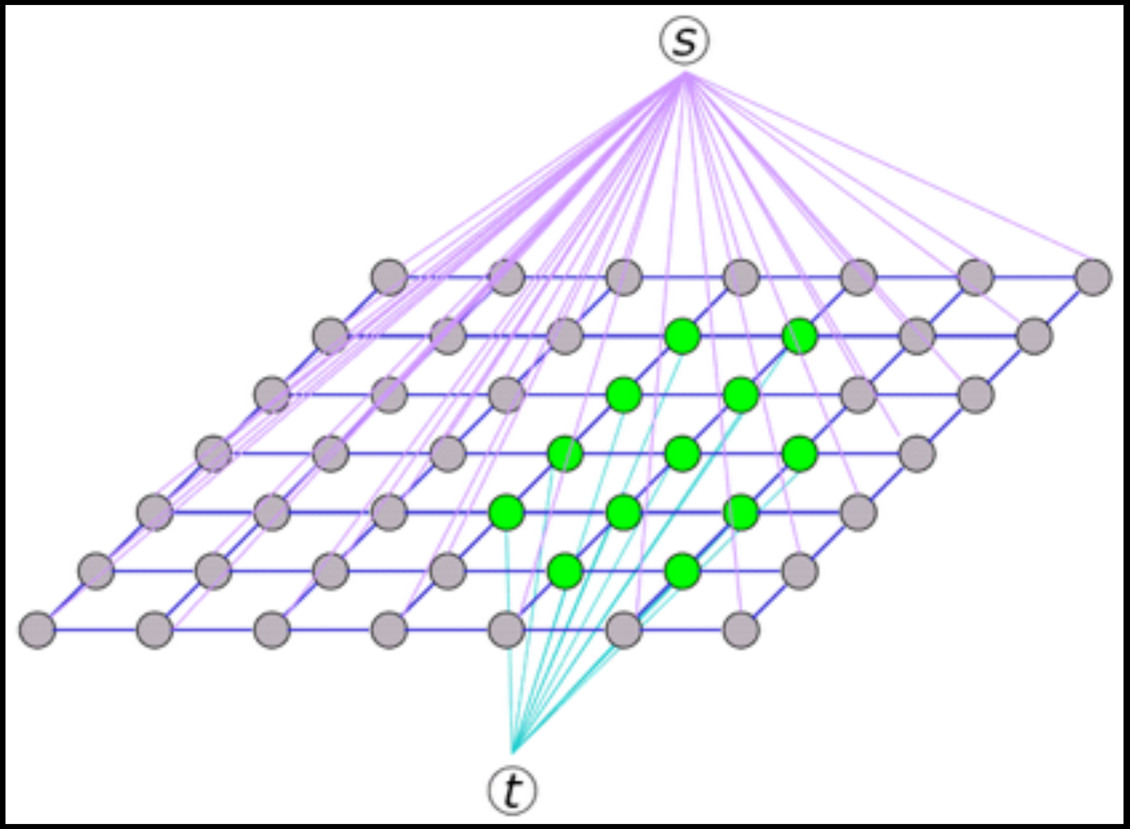
alpha-beta swap

alpha-expansion

Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.11 (2001): 1222-1239.

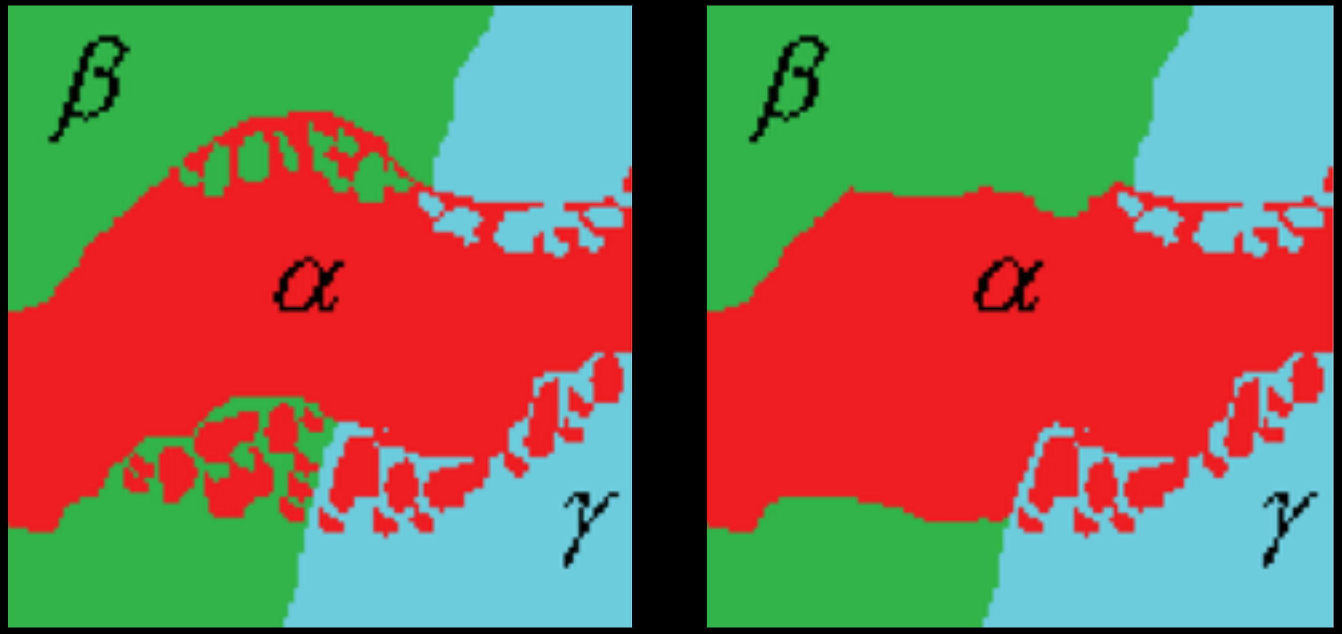
Remember graph cuts

- Assume a binary label set (e.g. black and white)
- Connect nodes to “source” and “sink”
- If the submodular property is satisfied ($\theta_{01} + \theta_{10} > \theta_{00} + \theta_{11}$) then we can decode this exactly solving a max-flow problem
- Problem: most images are not binary



http://www.caam.rice.edu/~wy1/ParaMaxFlow/2007/06/binary-code.html

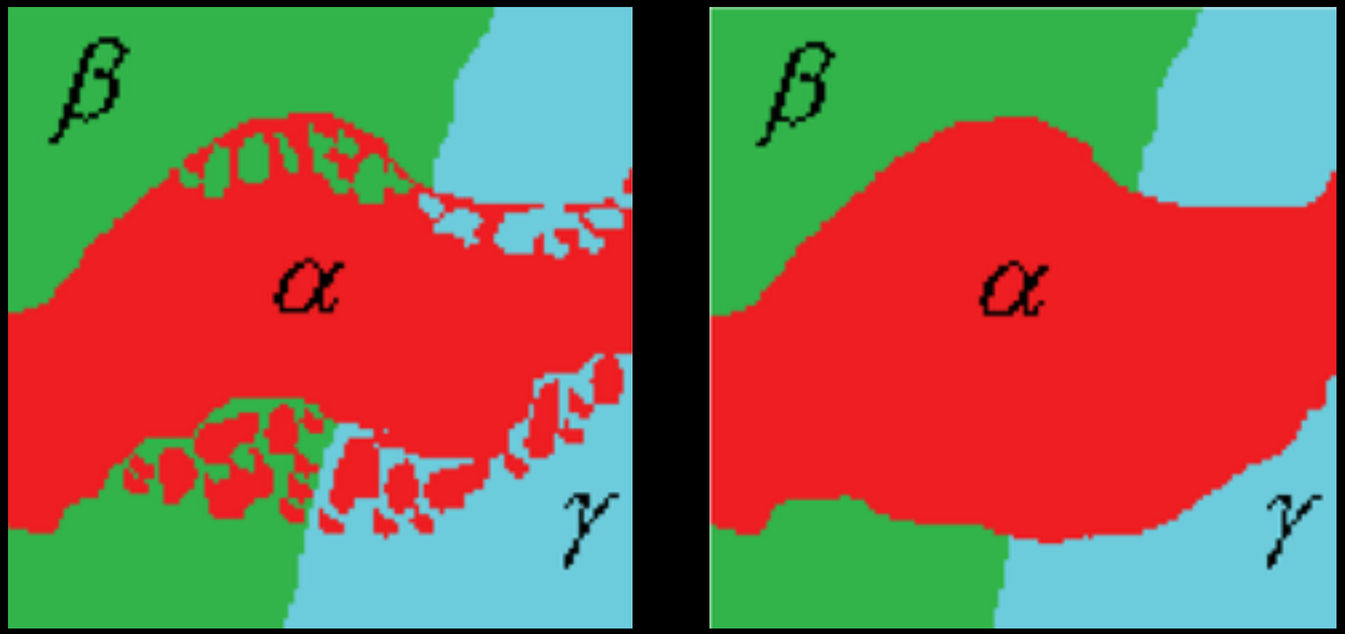
Alpha-beta swap



- All pixels not labelled alpha or beta stay fixed
- Do a graph cut on alpha and beta
- Some alpha nodes become beta and vice-versa

Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.11 (2001): 1222-1239.

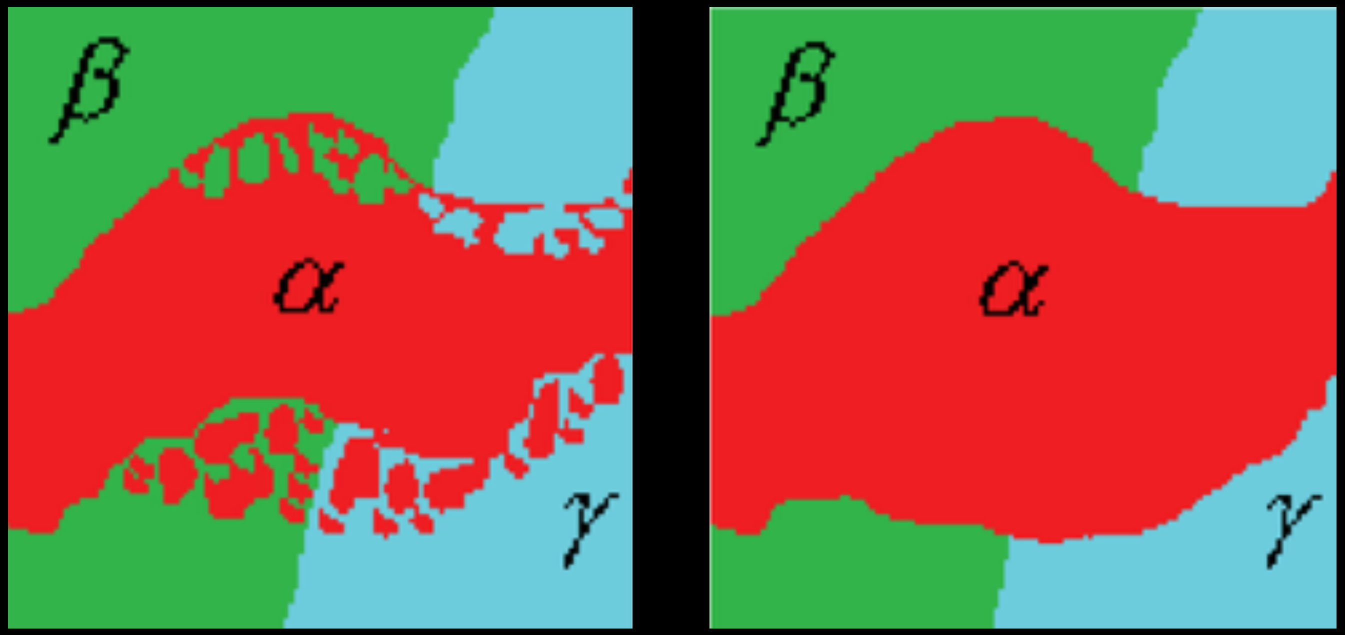
Alpha-expansion



- All pixels not alpha may change to alpha
- Some pixels become alpha

Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.11 (2001): 1222-1239.

Alpha-expansion



- The required constraints are stronger than those for beta-swap: $\theta(\alpha,\alpha)+\theta(\beta,\gamma)<\theta(\alpha,\gamma)+\theta(\beta,\alpha)$ for any alpha, beta and gamma triplet

Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.11 (2001): 1222-1239.

Performance on the Tsukuba image

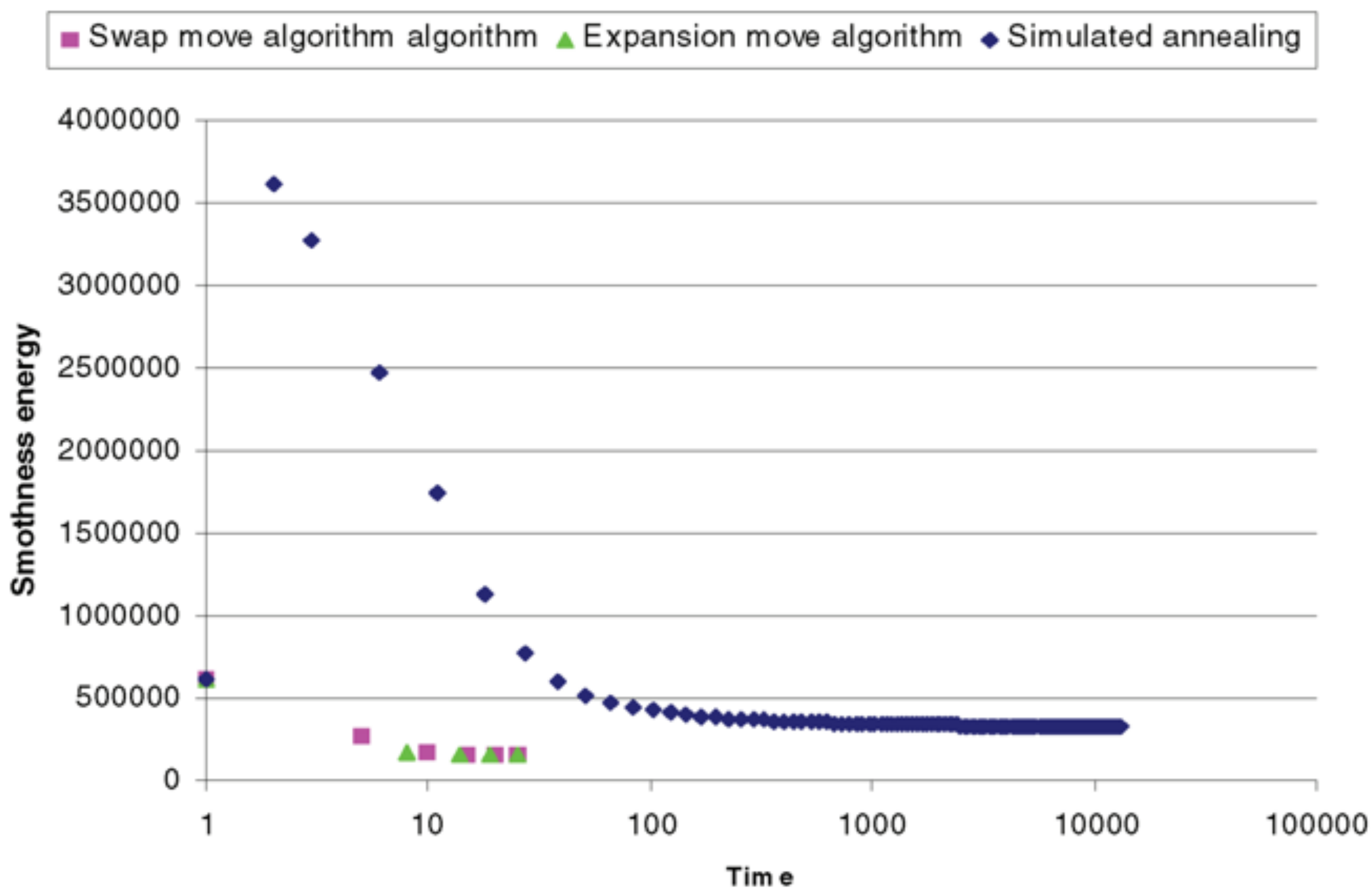


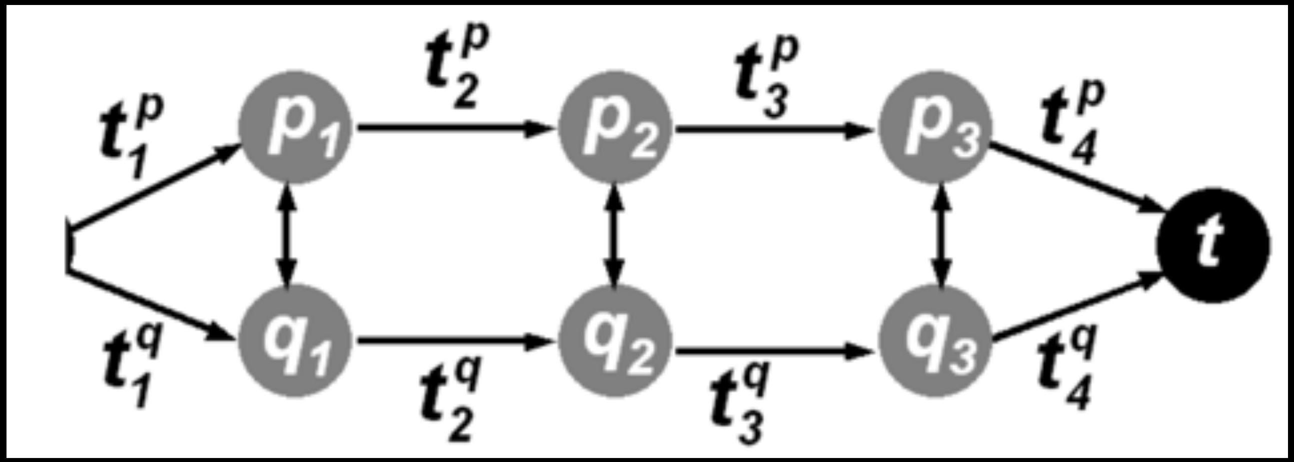
Fig. 11. Energy versus time (in seconds) of expansion, swap, and simulated annealing algorithms for the problem in Fig. 10a. The starting energy is the same for all algorithms.

Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.11 (2001): 1222-1239.

Exact multi-label optimization

- There is *one* case when multi-label optimization is possible
- We have to assume that the energy is

$$E(f) = \sum_{p \in \mathcal{P}} D_p f_p + \sum_{(p,q) \in \mathcal{N}} \lambda_{pq} |f_p - f_q|$$



- If the minimum cut severs edge t^p_i, assign label i to p

Boykov, Yuri, and Olga Veksler. "Graph cuts in vision and graphics: Theories and applications." Handbook of mathematical models in computer vision. Springer US, 2006. 79-96.

Cool applications

- Graphcut textures (SIGGRAPH 03): https://www.youtube.com/watch?v=Ya6BshBH6G4
- Grabcut demo (SIGGRAPH 04) using OpenCV: https://youtu.be/kAwxLTDDAwU?t=19s