CS664 Lecture #11:
Graph cuts for more than two labels

Some material taken from:

- **Yuri Boykov, University of Western Ontario**
  http://www.csd.uwo.ca/faculty/yuri/
- **Aseem Agarwala, University of Washington**
  http://www.cs.washington.edu/homes/aseem

Announcements

- PS 1 due in a week
- Guest lecture on Tuesday by Ashish Raj
  - Some vision problems in MR reconstruction

Recap

- Linear potentials can be solved exactly
- Potts model can be turned into the multiway cut problem

Local improvement method

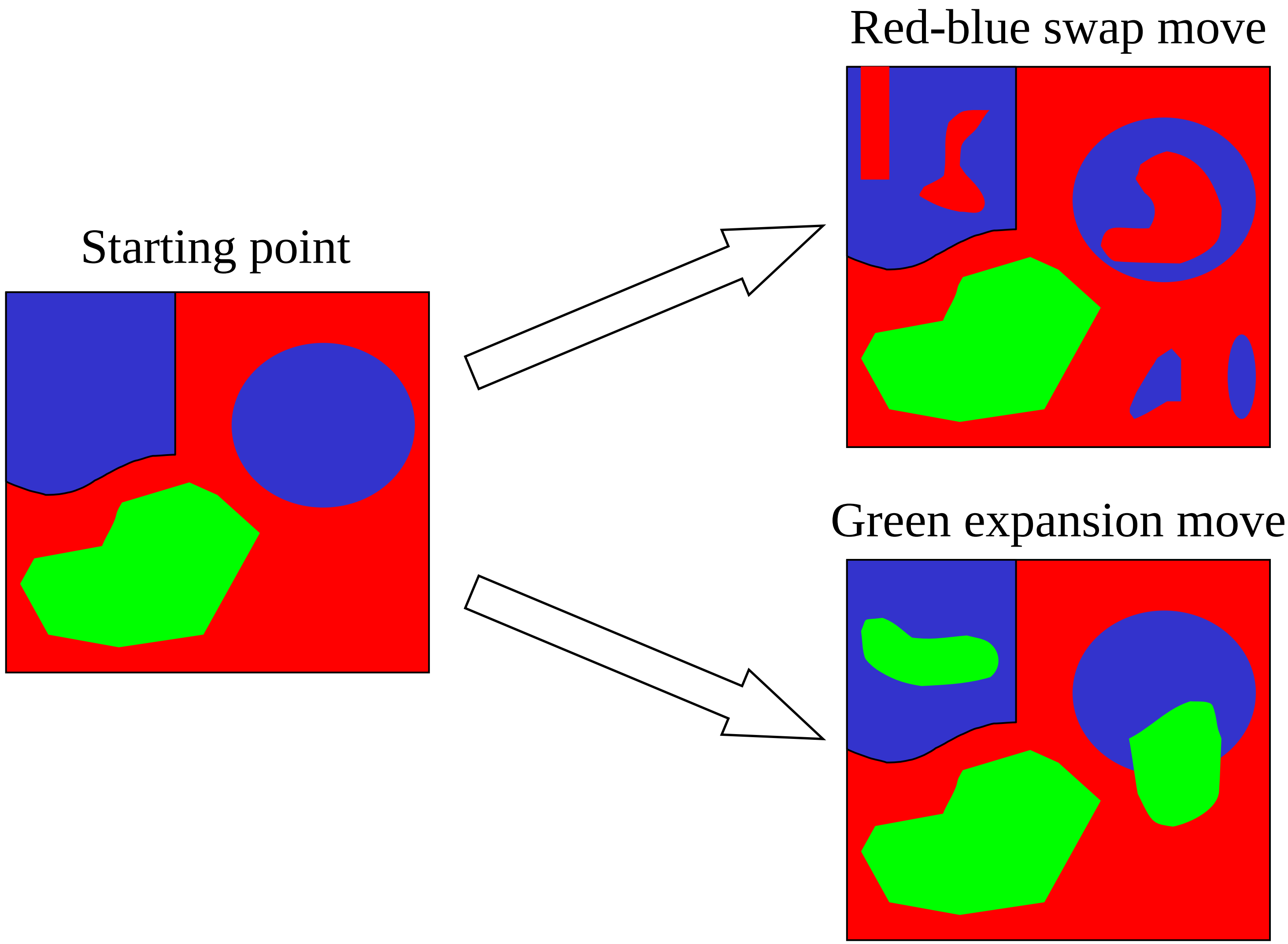
$$x_1 := \arg\min_x E(x, x_2, x_3, \ldots, x_n)$$
$$x_2 := \arg\min_x E(x_1, x, x_3, \ldots, x_n)$$
$$\vdots$$
$$x_n := \arg\min_x E(x_1, x_2, x_3, \ldots, x)$$

- Subproblem: pick a pixel, find the label that minimizes E, repeat
  - Minimize restricted version of E (line search)
  - Computes a local minimum

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
  - 2-approximation for the Potts model
  - Within 0.15% of global min!

Move examples



The swap move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label pair (α,β)
   2.1 Find the lowest *E* labeling within a single $\alpha\beta$-swap
   2.2 Go there if it's lower *E* than the current labeling
3. If *E* did not decrease in the cycle, done
   Otherwise, go to step 2

The expansion move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label α
   2.1 Find the lowest *E* labeling within a single α-expansion
   2.2 Go there if it's lower *E* than the current labeling
3. If *E* did not decrease in the cycle, done

Otherwise, go to step 2

Algorithm properties

- Graph cuts (only) used in key step 2.1
  - On a *binary* sub-problem, as we'll see
- In a cycle the energy doesn't increase
  - Convergence in *O(n)* cycles
    - In practice, termination occurs in a few cycles
- When the algorithms converge, the resulting labeling is a local minimum
  - Even when allowing an arbitrary swap move or expansion move

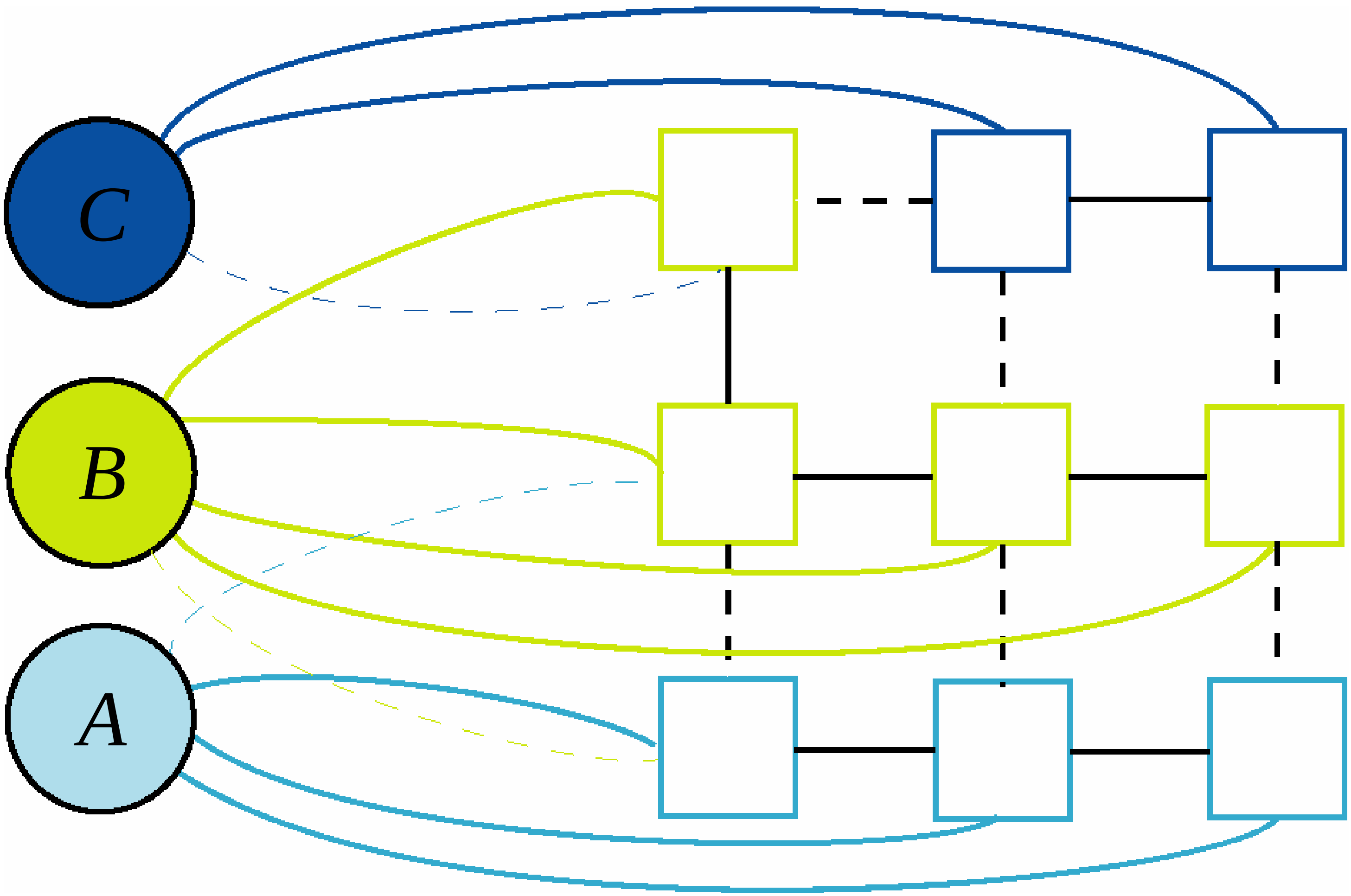
Very large neighborhood search

- A local minimum with respect to these moves is the best answer in a very large neighborhood
  - For example, there are $O(k\ 2^n)$ labelings within a single expansion move
  - Starting at an arbitrary labeling, you can get to the global minimum in *k* expansion moves

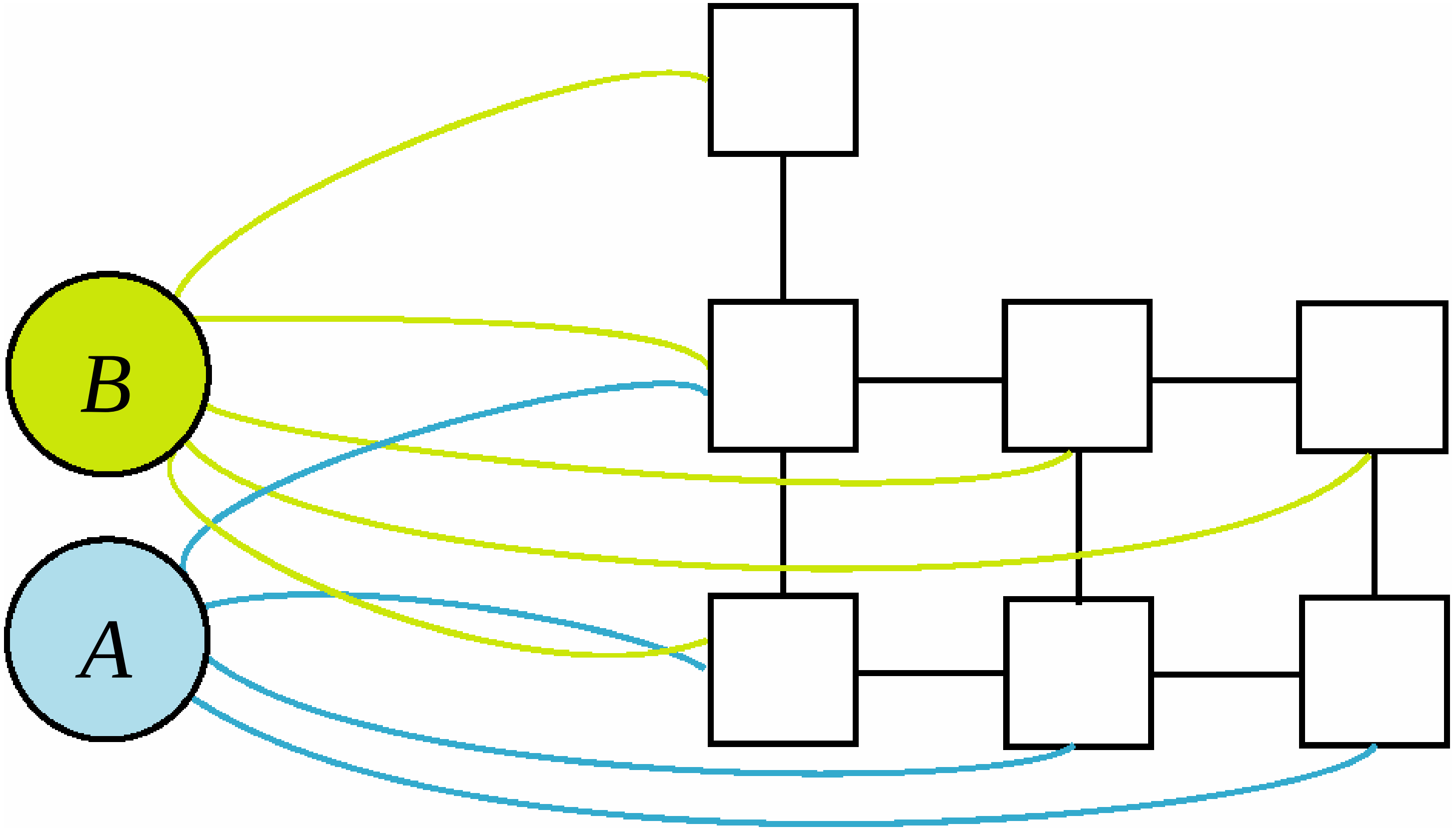
Swap move algorithm

- Let's look at how we compute the best swap move
  - It's basically just the two-label case we already know about
- Quite simple for the Potts model

Start with an arbitrary labeling



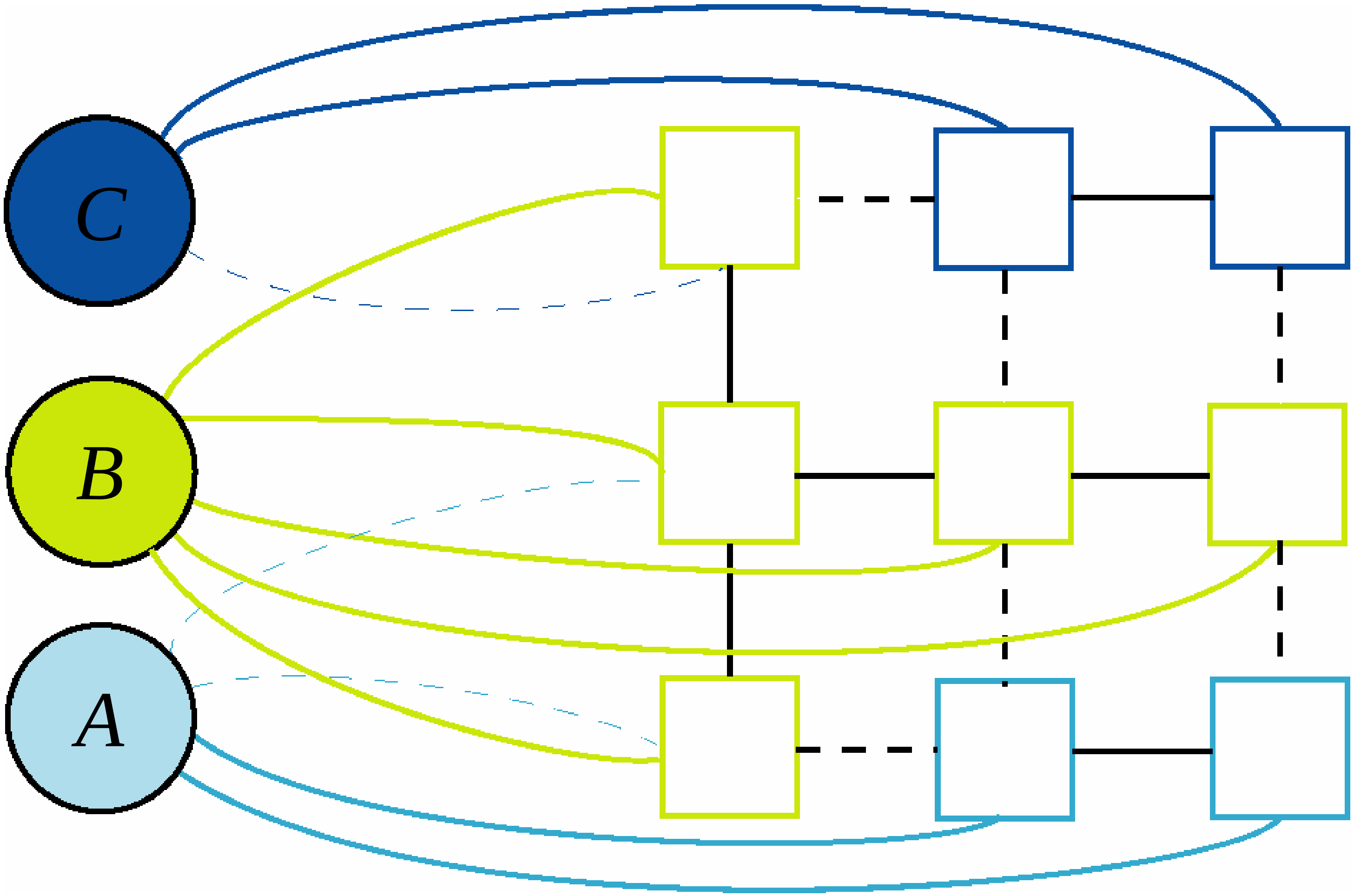
Compute min cut on subgraph



- The cost of cutting an n-link clearly depends on *A* and *B*

End of iteration

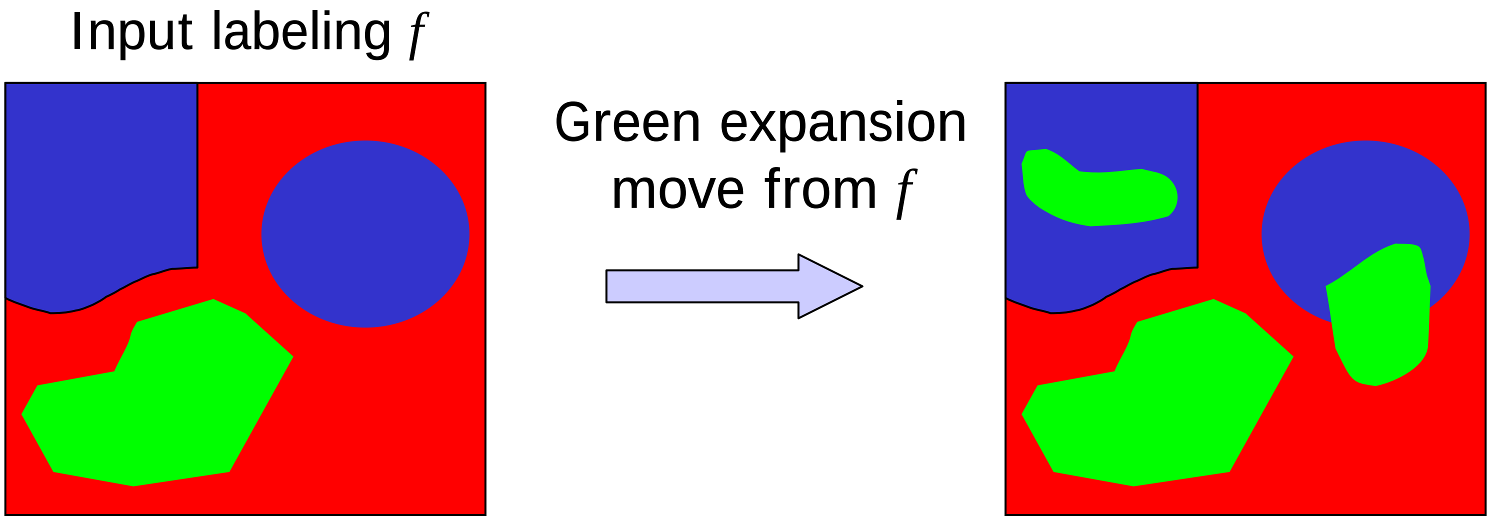
- Return the other nodes to the graph



Expansion move algorithm

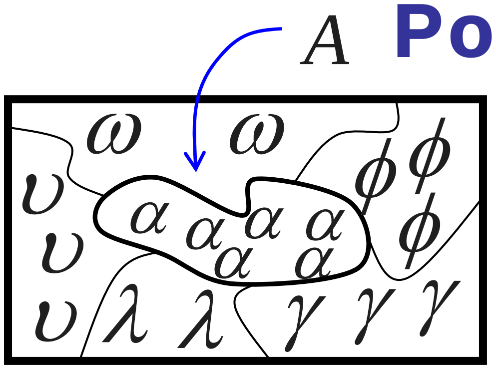
- Most powerful graph cut method
  - Widely used in practice
- Interesting theoretical properties also
  - 2-approximation for Potts model

Expansion move algorithm

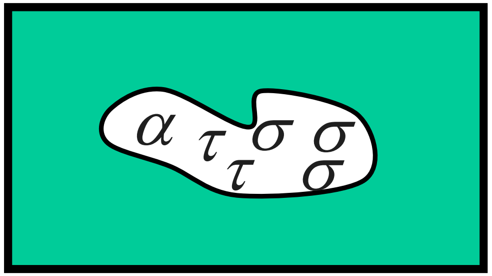


– Find green expansion move that most decreases E
  - Move there, then find the best blue expansion move, etc
  - Done when no α-expansion move decreases the energy, for any label α

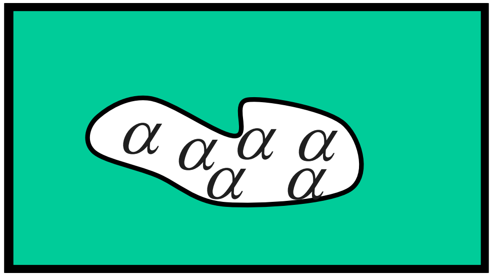
Potts model error bound



optimal solution f^*



local minimum $\hat{f}$



$$f^{\alpha} = \begin{cases} \alpha & p \in A \\ \hat{f}_p & p \notin A \end{cases}$$

$$\hat{f} \mapsto f^{\alpha} \quad \Rightarrow \quad E\left(\hat{f}\right) \le E\left(f^{\alpha}\right)$$

$$E_A\left(\hat{f}\right) \le E_A\left(f^{\alpha}\right) \le E_A\left(f^{*}\right)$$

Summing up over all labels:

$$E\left(\hat{f}\right) \le E\left(f^{*}\right) + E_{\partial}\left(f^{*}\right) \le 2E\left(f^{*}\right)$$

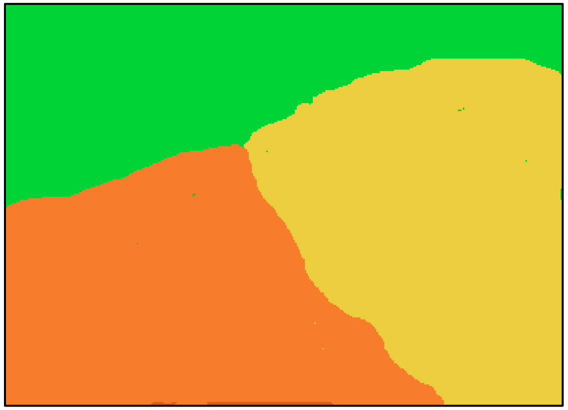
Expansion moves in action

initial solution

- -expansion
- -expansion
- -expansion
- -expansion
- -expansion
- -expansion
- -expansion

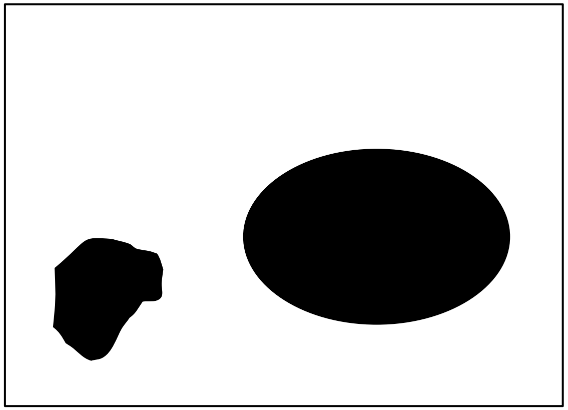
For each move we choose expansion that gives the largest decrease in the energy: **binary energy minimization subproblem**

Binary sub-problem



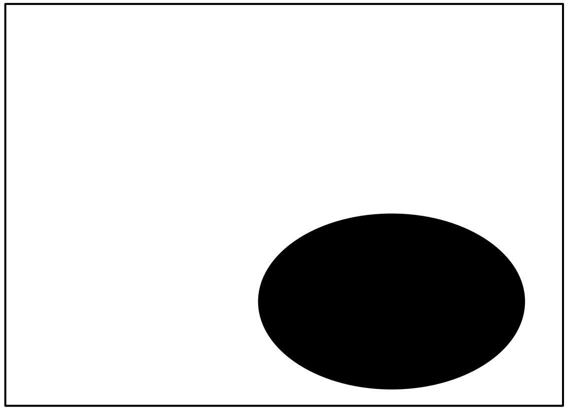
Input labeling

Expansion move



Binary image

Expansion move energy



Goal: find the binary image with lowest energy

Binary image energy is a restriction of E

- Depends on f,α

Graph cuts solution

- This can be done as long as V has a specific form (works for arbitrary D)
- Regularity constraint [KZ PAMI '04]
  - Can find cheapest α-expansion from f if

$$V(\alpha,\alpha) + V(f(p),f(q)) \leq V(f(p),\alpha) + V(\alpha,f(q))$$

Regular choices of V

- Suppose that V is a metric

$$
\begin{aligned}
V(\alpha, \alpha) &= 0 \\
V(\alpha, \beta) &= V(\beta, \alpha) \\
V(\beta, \phi) &\leq V(\beta, \alpha) + V(\alpha, \phi)
\end{aligned}
$$

- Then what?

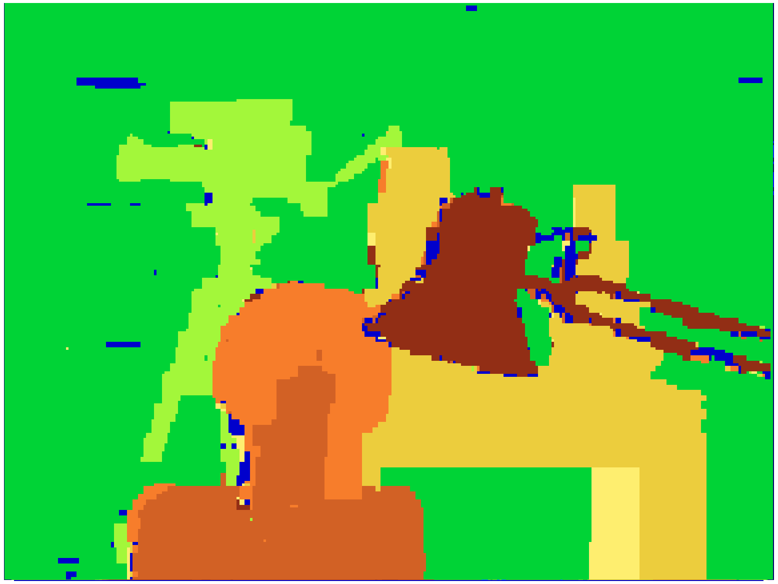
$$
\begin{aligned}
V(\alpha, \alpha) \;+\; V(f(p), f(q)) \quad &\leq \\
V(f(p), \alpha) \;+\; V(\alpha, f(q))
\end{aligned}
$$

What V's are metrics?

- Important examples
  - Potts model
  - L1 distance
  - Truncated L1 distance
  - NOT the L2 distance (with or without truncation)
- There is an easy way to check the triangle inequality

Old methods: How fast do you want the wrong answer?

Right answers



Slow (annealing)
Graph cuts
Fast

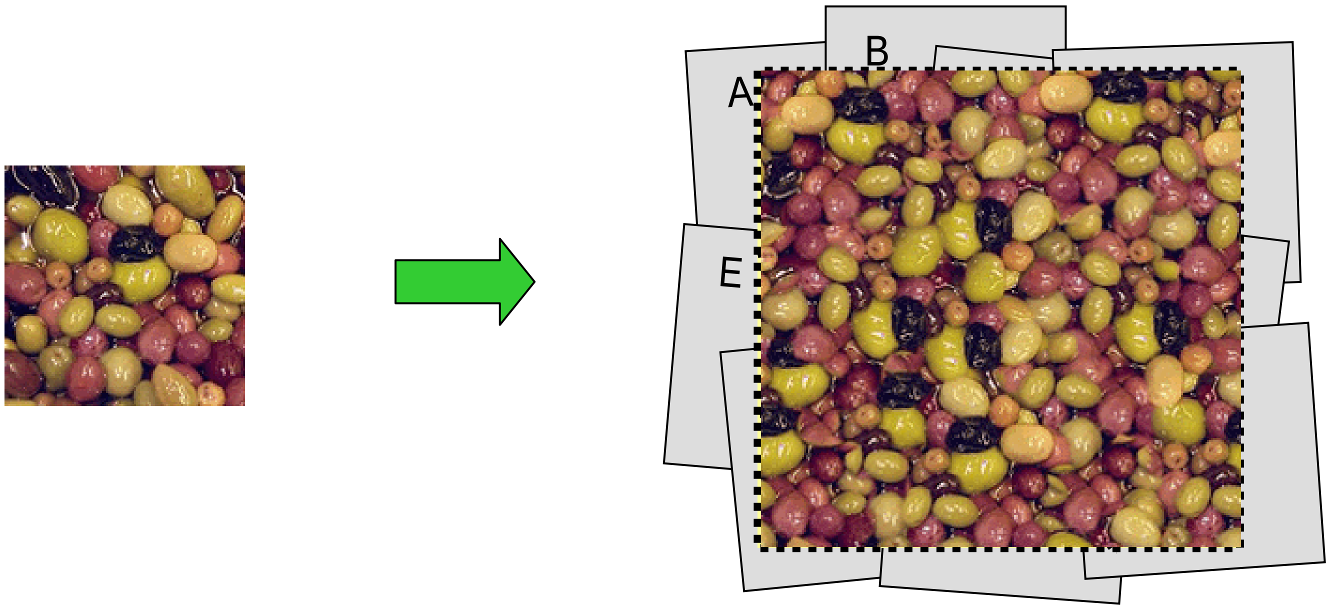
Statistical performance

	Tsukuba			Sawtooth			Venus			Map	
	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\overline{\mathcal{T}}}$	$B_{\mathcal{D}}$	$B_{\overline{\mathcal{O}}}$	$B_{\mathcal{D}}$
20 Layered	1.58 *3*	1.06 *4*	8.82 *3*	**0.34** *1*	**0.00** *1*	**3.35** *1*	1.52 *3*	2.96 *10*	2.62 *2*	0.37 *6*	5.24 *6*
*4 Graph cuts	1.94 *5*	1.09 *5*	9.49 *5*	1.30 *6*	0.06 *3*	6.34 *6*	1.79 *7*	2.61 *8*	6.91 *4*	0.31 *4*	3.88 *4*
19 Belief prop.	**1.15** *1*	**0.42** *1*	**6.31** *1*	0.98 *5*	0.30 *5*	4.83 *5*	1.00 *2*	0.76 *2*	9.13 *6*	0.84 *10*	5.27 *7*
11 GC+occl.	1.27 *2*	0.43 *2*	6.90 *2*	0.36 *2*	**0.00** *1*	3.65 *2*	2.79 *12*	5.39 *13*	**2.54** *1*	1.79 *13*	10.08 *12*
10 Graph cuts	1.86 *4*	1.00 *3*	9.35 *4*	0.42 *3*	0.14 *4*	3.76 *3*	1.69 *6*	2.30 *6*	5.40 *3*	2.39 *16*	9.35 *10*
8 Multiw. cut	8.08 *17*	6.53 *14*	25.33 *18*	0.61 *4*	0.46 *8*	4.60 *4*	**0.53** *1*	**0.31** *1*	8.06 *5*	0.26 *3*	3.27 *3*
12 Compact win.	3.36 *8*	3.54 *8*	12.91 *9*	1.61 *9*	0.45 *7*	7.87 *7*	1.67 *5*	2.18 *4*	13.24 *9*	0.33 *5*	3.94 *5*
14 Realtime	4.25 *12*	4.47 *12*	15.05 *13*	1.32 *7*	0.35 *6*	9.21 *8*	1.53 *4*	1.80 *3*	12.33 *7*	0.81 *9*	11.35 *15*
*5 Bay. diff.	6.49 *16*	11.62 *19*	12.29 *7*	1.45 *8*	0.72 *9*	9.29 *9*	4.00 *14*	7.21 *16*	18.39 *13*	**0.20** *1*	2.49 *2*
9 Cooperative	3.49 *9*	3.65 *9*	14.77 *11*	2.03 *10*	2.29 *14*	13.41 *13*	2.57 *11*	3.52 *11*	26.38 *17*	0.22 *2*	**2.37** *1*
*1 SSD+MF	5.23 *15*	3.80 *10*	24.66 *17*	2.21 *11*	0.72 *10*	13.97 *15*	3.74 *13*	6.82 *15*	12.94 *8*	0.66 *8*	9.35 *10*
15 Stoch. diff.	3.95 *10*	4.08 *11*	15.49 *15*	2.45 *14*	0.90 *11*	10.58 *10*	2.45 *9*	2.41 *7*	21.84 *15*	1.31 *12*	7.79 *9*
13 Genetic	2.96 *6*	2.66 *7*	14.97 *12*	2.21 *12*	2.76 *16*	13.96 *14*	2.49 *10*	2.89 *9*	23.04 *16*	1.04 *11*	10.91 *14*
7 Pix-to-pix	5.12 *14*	7.06 *17*	14.62 *10*	2.31 *13*	1.79 *12*	14.93 *17*	6.30 *17*	11.37 *18*	14.57 *10*	0.50 *7*	6.83 *8*
6 Max flow	2.98 *7*	2.00 *6*	15.10 *14*	3.47 *15*	3.00 *17*	14.19 *16*	2.16 *8*	2.24 *5*	21.73 *14*	3.13 *17*	15.98 *18*
*3 Scanl. opt.	5.08 *13*	6.78 *15*	11.94 *6*	4.06 *16*	2.64 *15*	11.90 *11*	9.44 *19*	14.59 *19*	18.20 *12*	1.84 *14*	10.22 *13*
*2 Dyn. prog.	4.12 *11*	4.63 *13*	12.34 *8*	4.84 *19*	3.71 *19*	13.26 *12*	10.10 *20*	15.01 *20*	17.12 *11*	3.33 *18*	14.04 *17*
17 Shao	9.67 *18*	7.04 *16*	35.63 *19*	4.25 *17*	3.19 *18*	30.14 *20*	6.01 *16*	6.70 *14*	43.91 *20*	2.36 *15*	33.01 *20*
16 Fast Correl.	9.76 *19*	13.85 *20*	24.39 *16*	4.76 *18*	1.87 *13*	22.49 *18*	6.48 *18*	10.36 *17*	31.29 *18*	8.42 *20*	12.68 *16*
18 Max surf.	11.10 *20*	10.70 *18*	41.99 *20*	5.51 *20*	5.56 *20*	27.39 *19*	4.36 *15*	4.78 *12*	41.13 *19*	4.17 *19*	27.88 *19*

Expansion move applications

- Lots of them, in vision and in graphics
- We'll look at a few really "cute" examples
  - I.e., from SIGGRAPH
  - Texture synthesis ("graphcut textures")
  - Photomontage
  - Panoramic video textures
- Then two important ones from vision
  - Stereo with sloped and curved surfaces
    - Via the Potts model!
  - Multi-camera stereo

Application: texture synthesis

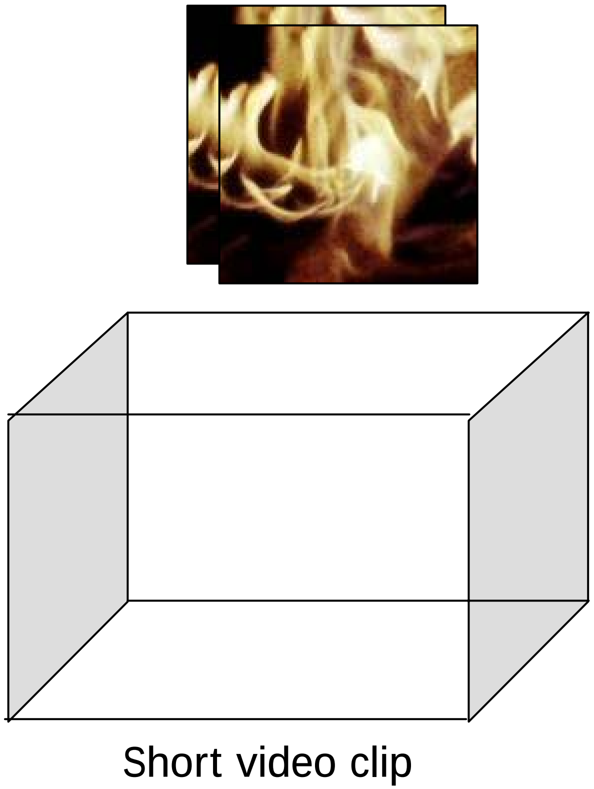


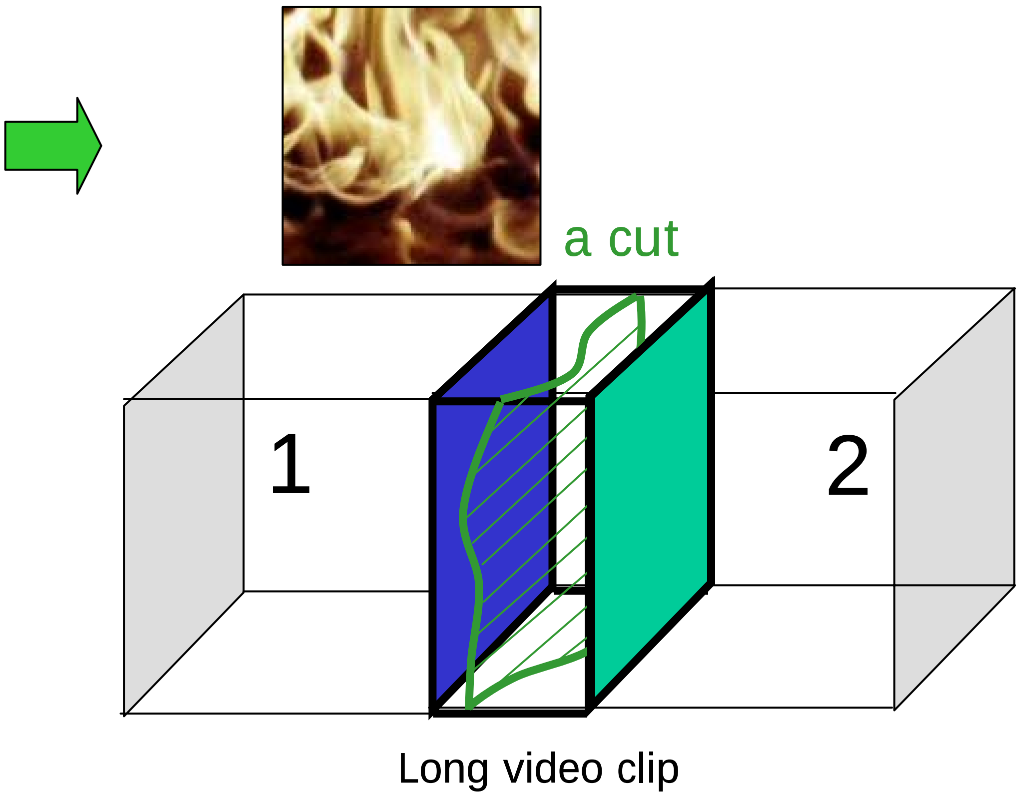
"Graphcut textures"

(Kwatra, Schodl, Essa, Bobick SIGGRAPH2003)

http://www.cc.gatech.edu/cpl/projects/graphcuttextures/

Graphcuts video textures





Another example

original short clip

synthetic infinite texture

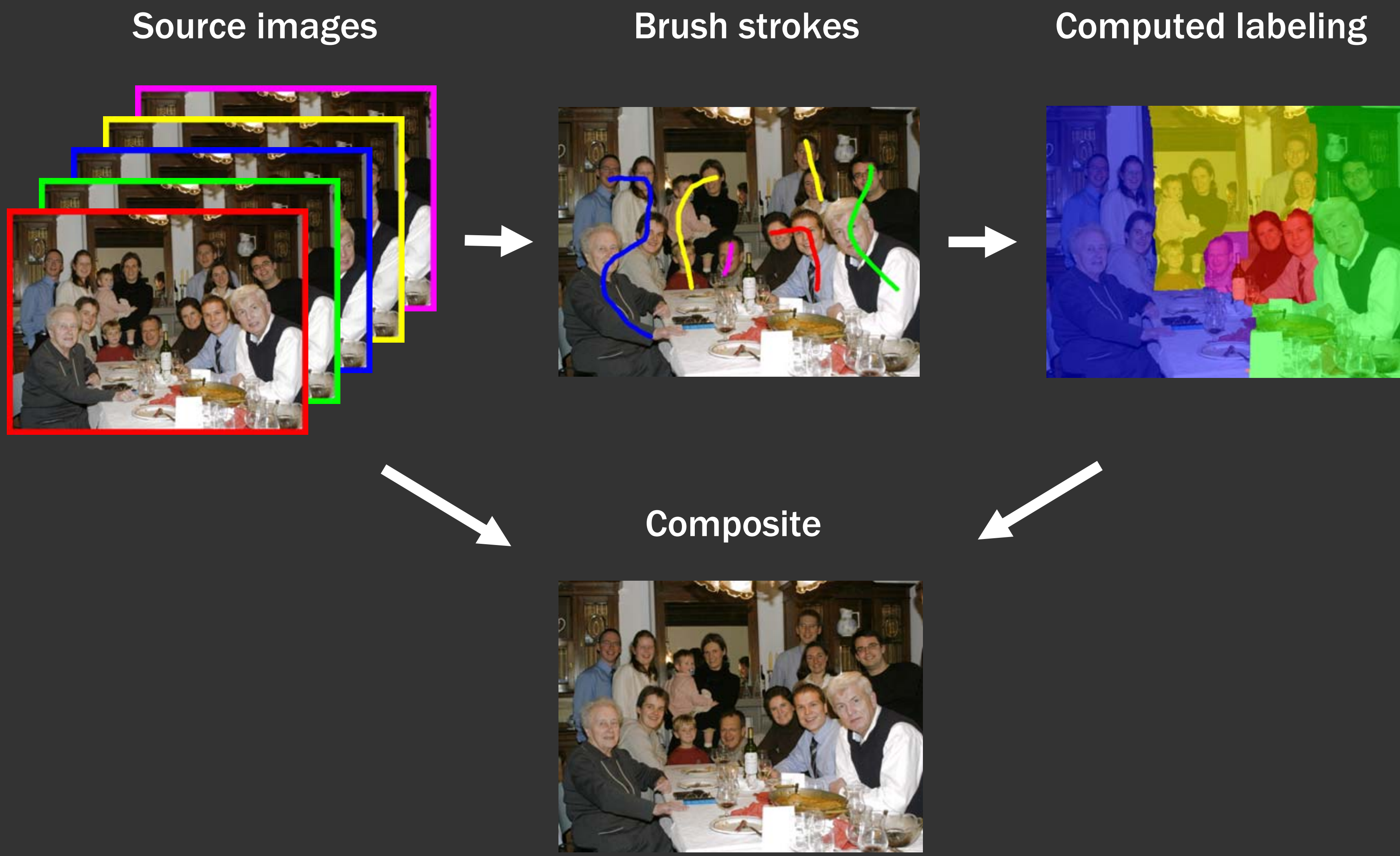
Interactive Digital Photomontage

Aseem Agarwala, Mira Dontcheva,
Maneesh Agrawala, Steven Drucker, Alex Colburn,
Brian Curless, David Salesin, Michael Cohen

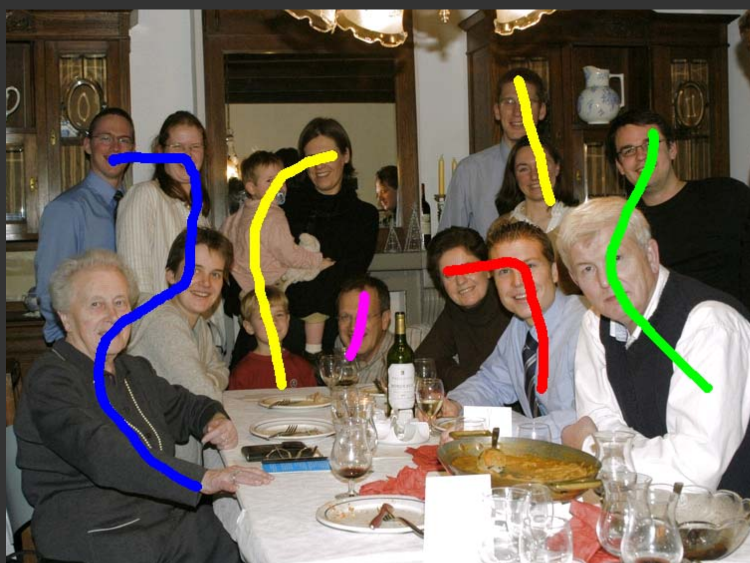
University of Washington & Microsoft Research

set of originals / actual

photomontage / perceived


Source images
Brush strokes
Computed labeling
Composite

Brush strokes

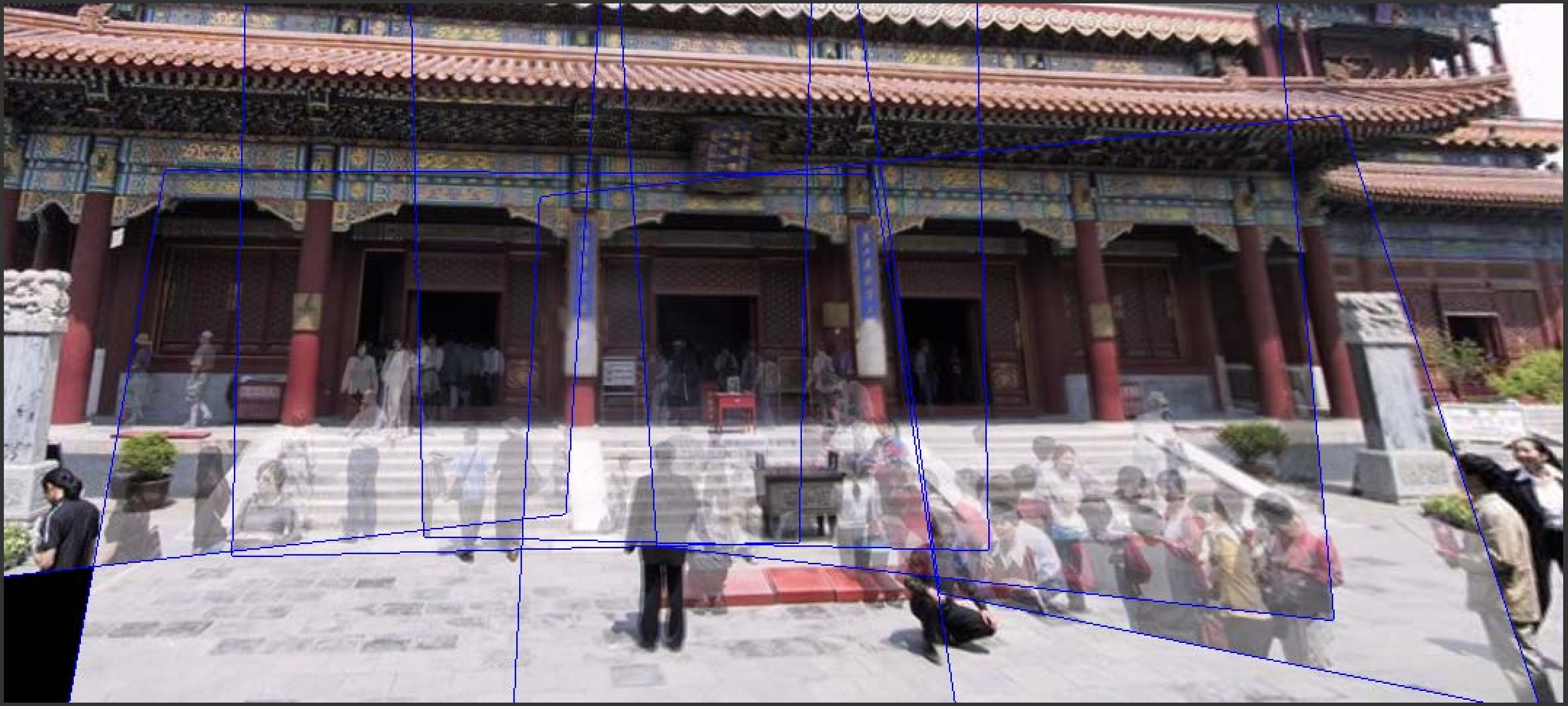


Computed labeling

Image objective

Demos

courtesy of P. Debevec

Time

Time

Time

Time