Graphics isn't all about 3-D

Jean-François Lalonde

April 21st 2009

Generating images

Generating images

Using images to create images

Let’s do it!

- Step 1: cut




- Step 2: paste

Composite by David Dewey

Issues in compositing

Long process!

Semi-transparent objects

Pixels too large

Our goal

Make our lives easier!

Current state of the art

WARNING

Hidden assumptions

Image from [Jia *et al.*, 2006]

Matting

- Compositing equation

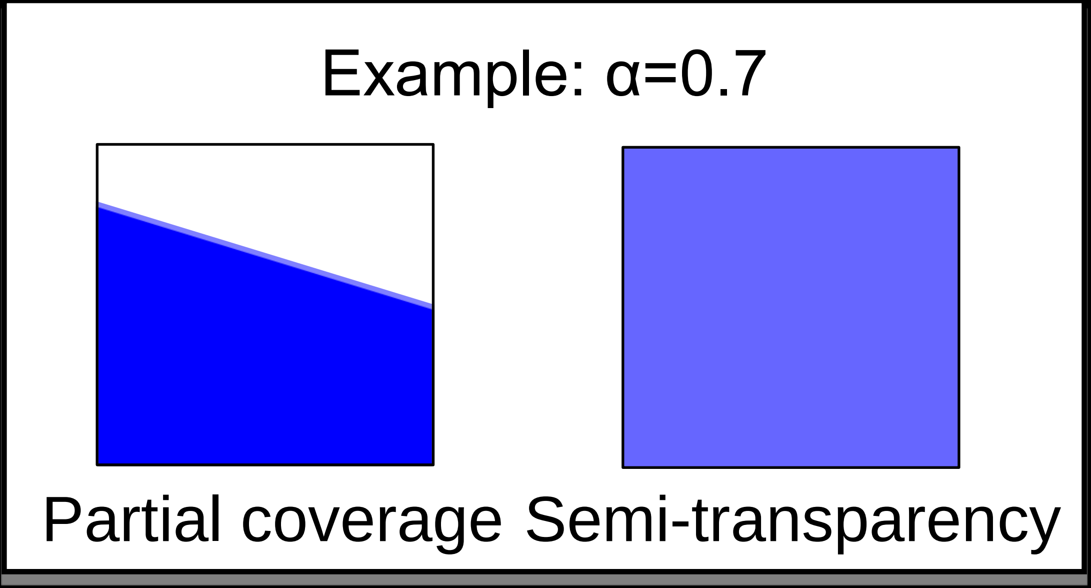
Image α Foreground (1-α) Background

$$I = \alpha F + (1 - \alpha) B, \alpha \in [0, 1]$$

Images from [Sun *et al.*, 2004]

"Pulling the matte"

- Interpretation of α



- Goal: recover F, B, and α automatically
- Why is it hard (impossible)?
  - 1 equation, 3 unknowns!

$$I = \alpha F + (1 - \alpha) B$$

Matting: simplifying assumptions

- Blue-screen matting
  - Petro Vlahos, ‘50s: Ultimatte® still most popular
  - Oscar for lifetime achievement!

Blue-screen matting

$$\begin{aligned} R &= \alpha R_f + (1-\alpha)R_b \\ G &= \alpha G_f + (1-\alpha)G_b \\ B &= \alpha B_f + (1-\alpha)B_b \end{aligned}$$

- Solution #1: no blue! $B_f = 0$
- Solve for alpha:

$$B = (1-\alpha)B_b$$

What is the main limitation here?

- Get R and G

$$\begin{aligned} R &= \alpha R_f + (1-\alpha)R_b \\ G &= \alpha G_f + (1-\alpha)G_b \end{aligned}$$

What about superman?

Blue-screen matting

$$
\begin{aligned}
R &= \alpha R_f + (1-\alpha)R_b \\
G &= \alpha G_f + (1-\alpha)G_b \\
B &= \alpha B_f + (1-\alpha)B_b
\end{aligned}
$$

- Solution #2: gray or flesh

$$R_f = G_f = B_f \qquad R_f = 0.5G_f = 0.5B_f$$

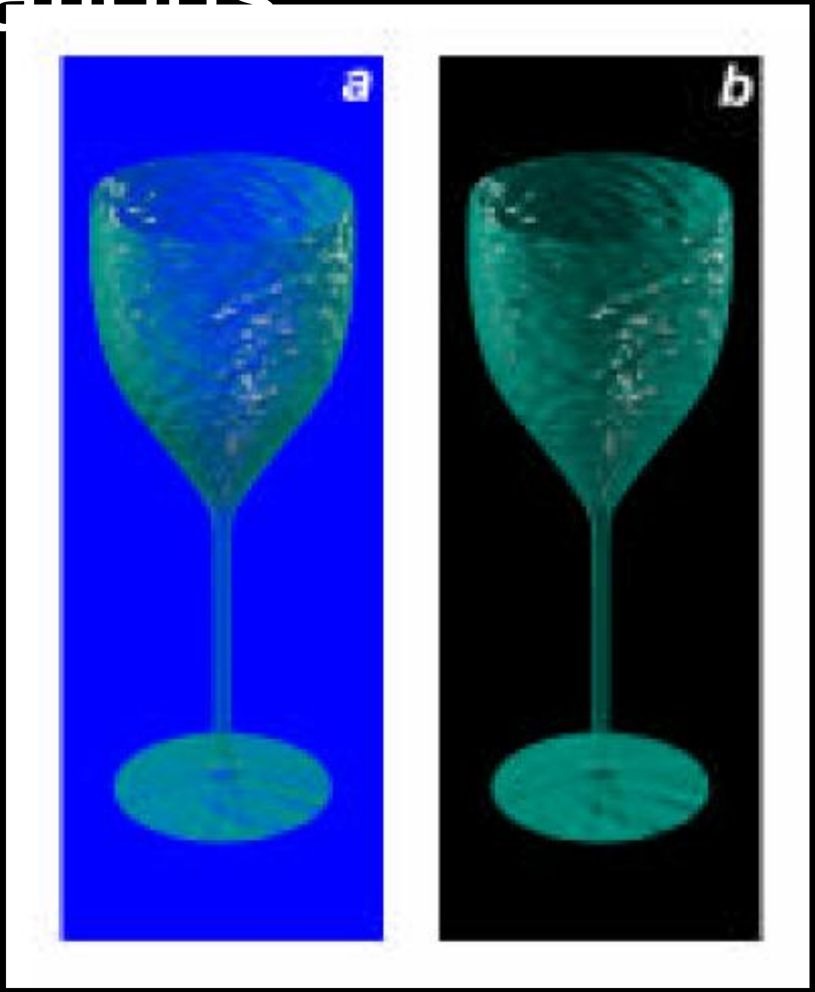
- 3 equations, 2 unknowns

Triangulation Matting

[Smith & Blinn, 1996]

$$I = \alpha F + (1 - \alpha) B$$

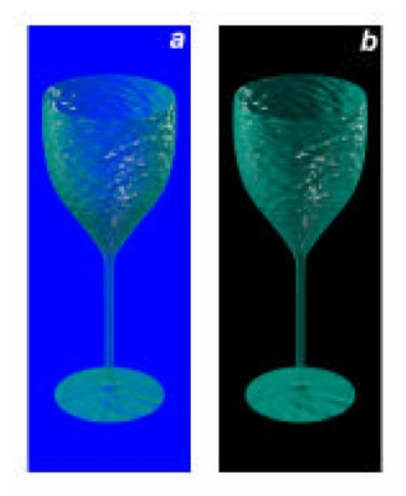
- Instead of reducing unknowns, add more equations



Triangulation matting

$$I = \alpha F + (1 - \alpha) B$$

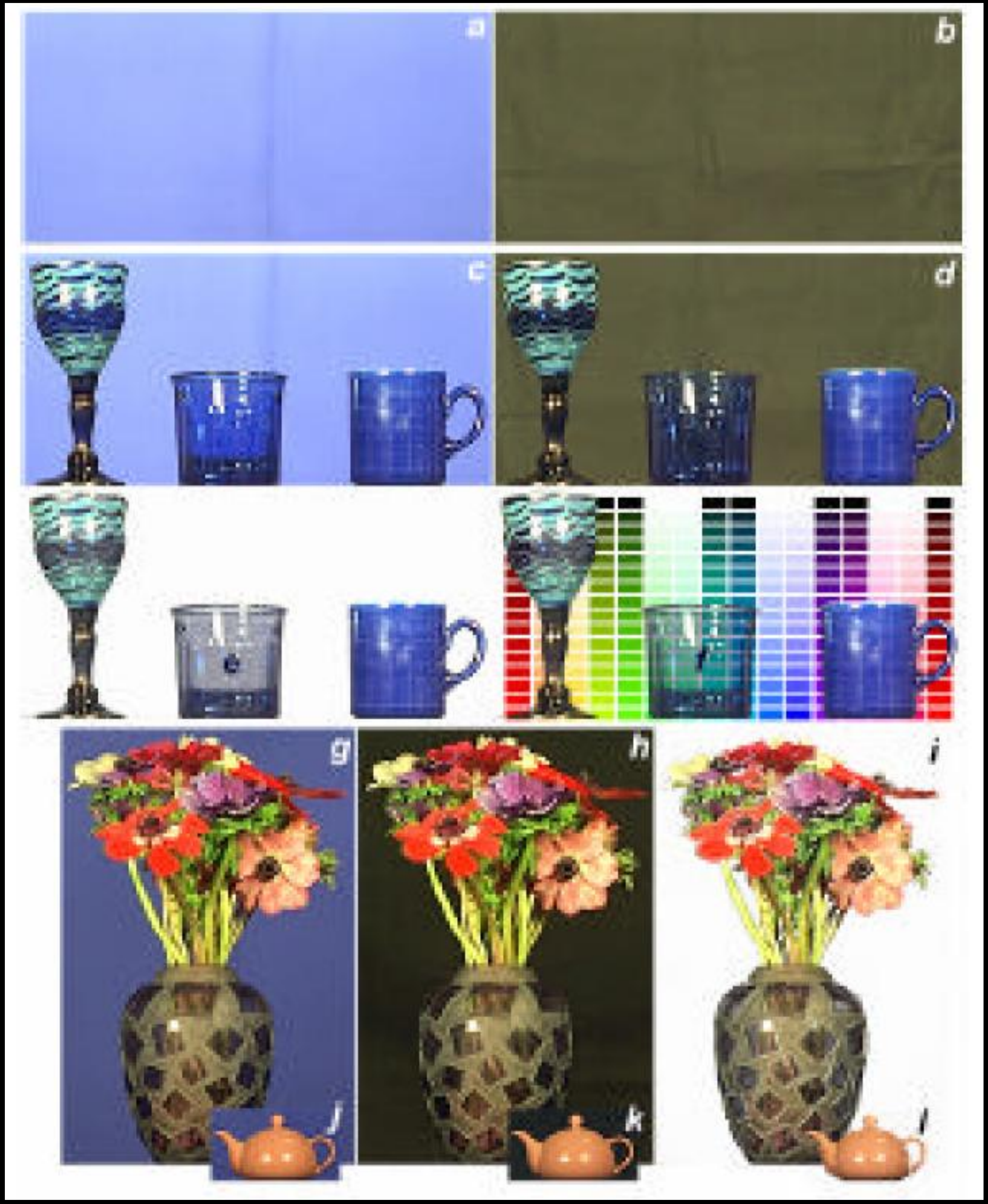
- Same unknowns, more equations!
  - How many?



$$\begin{aligned} R &= \alpha R_f + (1-\alpha) R_{b_1} & R &= \alpha R_f + (1-\alpha) R_{b_2} \\ G &= \alpha G_f + (1-\alpha) G_{b_1} & G &= \alpha G_f + (1-\alpha) G_{b_2} \\ B &= \alpha B_f + (1-\alpha) B_{b_1} & B &= \alpha B_f + (1-\alpha) B_{b_2} \end{aligned}$$

Do we need constant backgrounds?

Some results



From [Smith & Blinn, 1996]

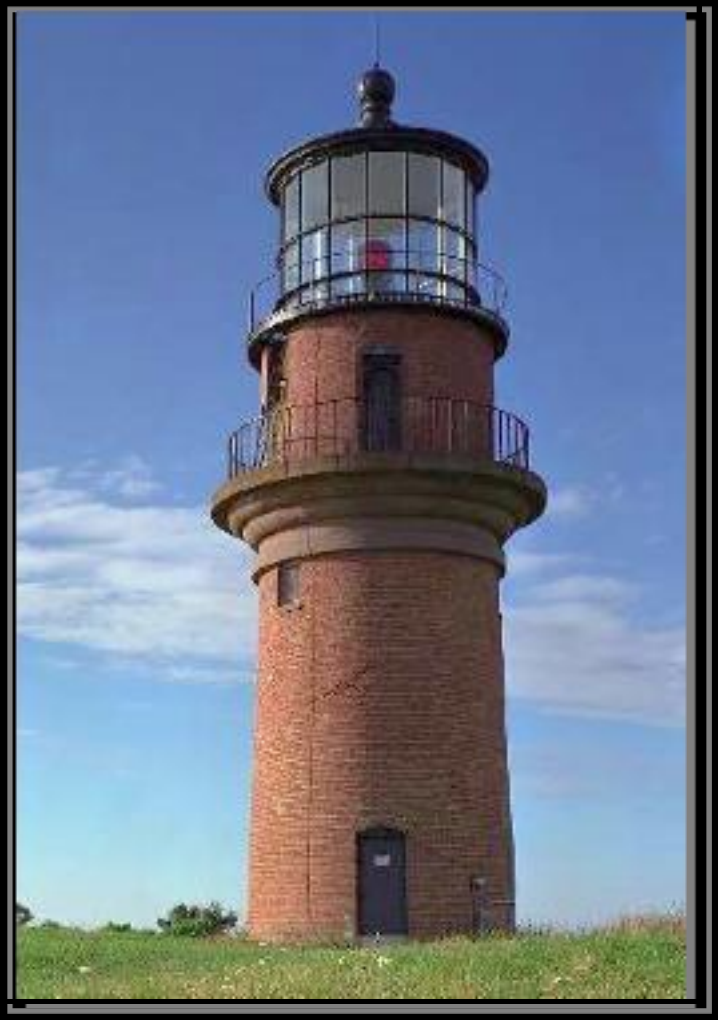
Matting

Single image AND complex background?

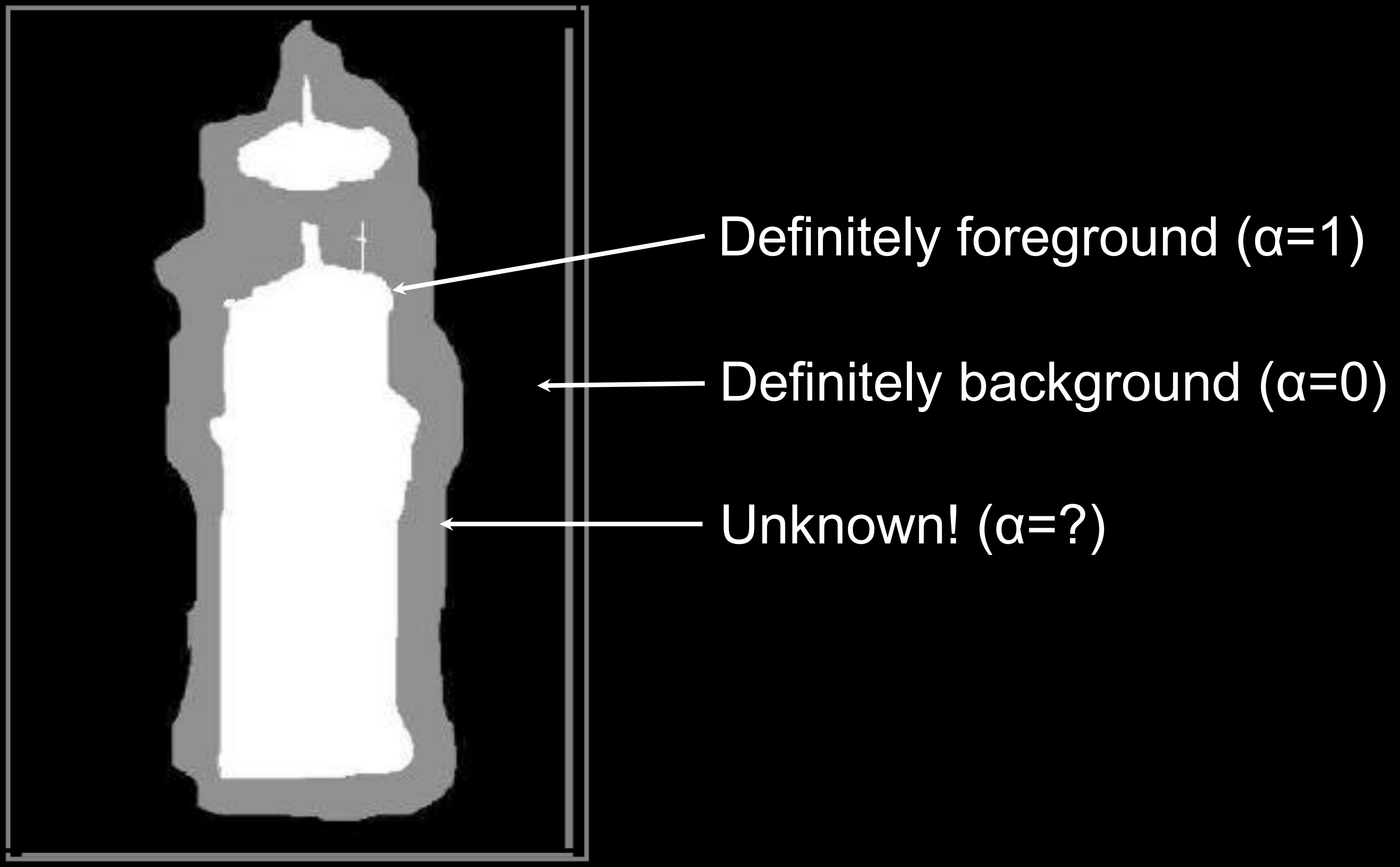
Image α Foreground (1-α) Background

Matting: simplifying assumptions

Input image



UI: Trimap

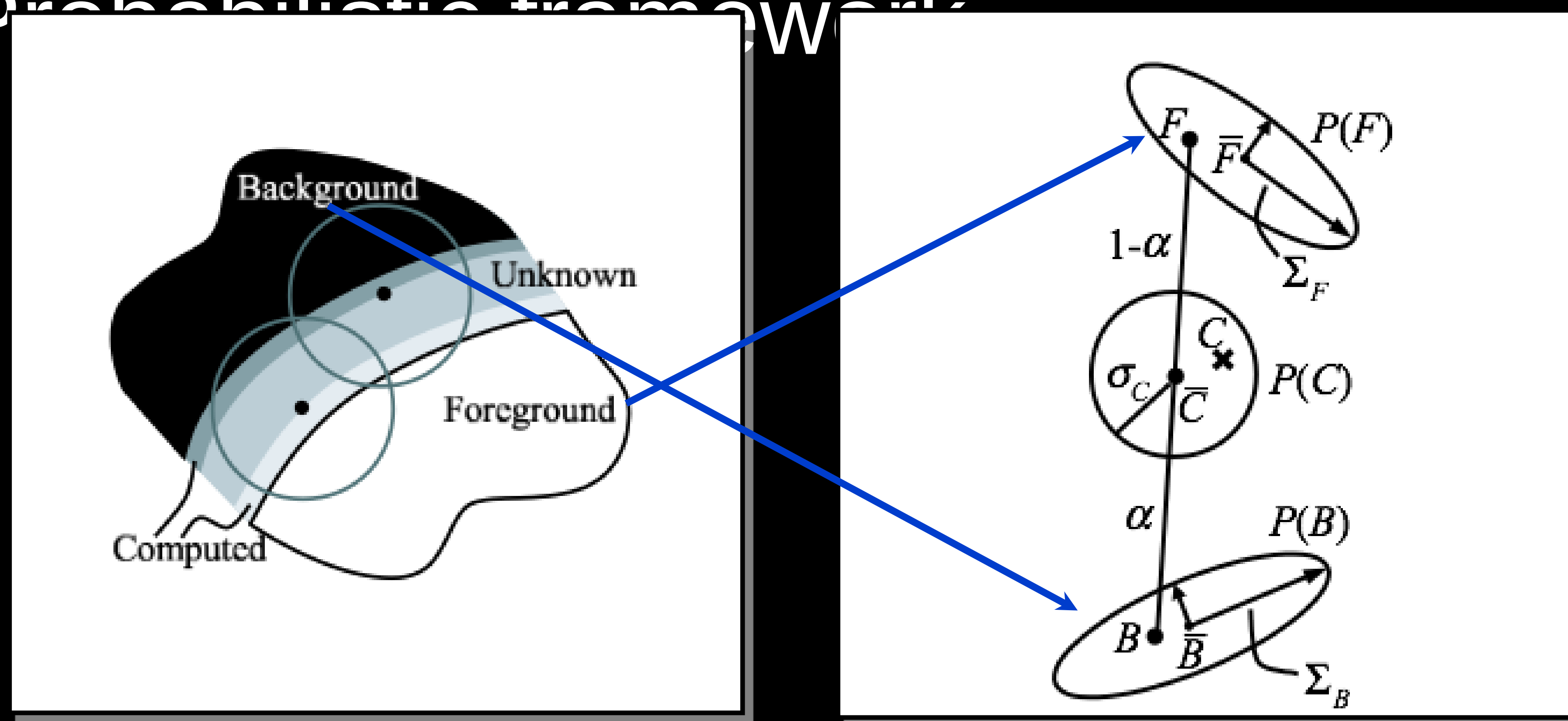


Assume some knowledge about foreground and background

Images from [Chuang *et al.*, 2001]

Bayesian Matting

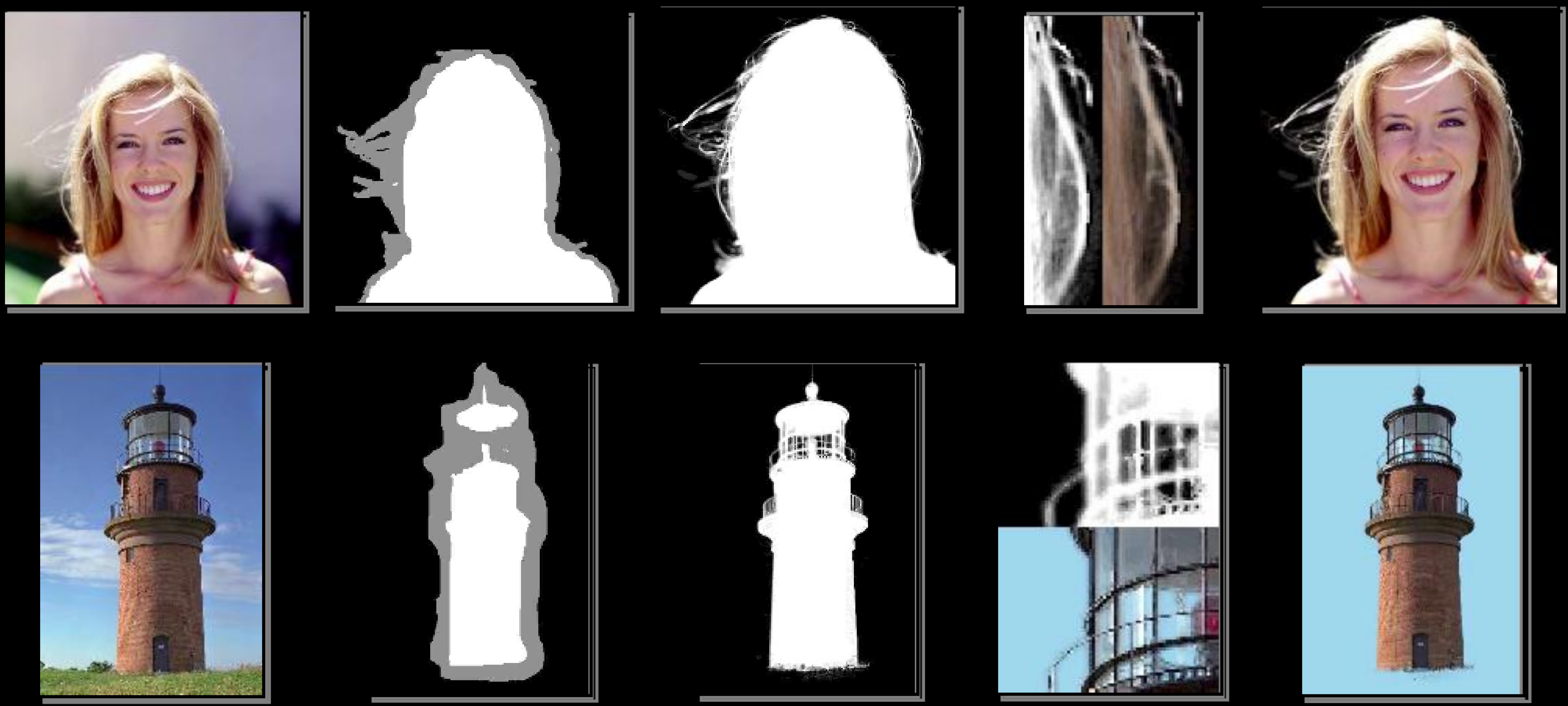
- Model foreground and background color distributions
- Probabilistic framework



From [Chuang *et al.*, 2001]

Bayesian Matting Results

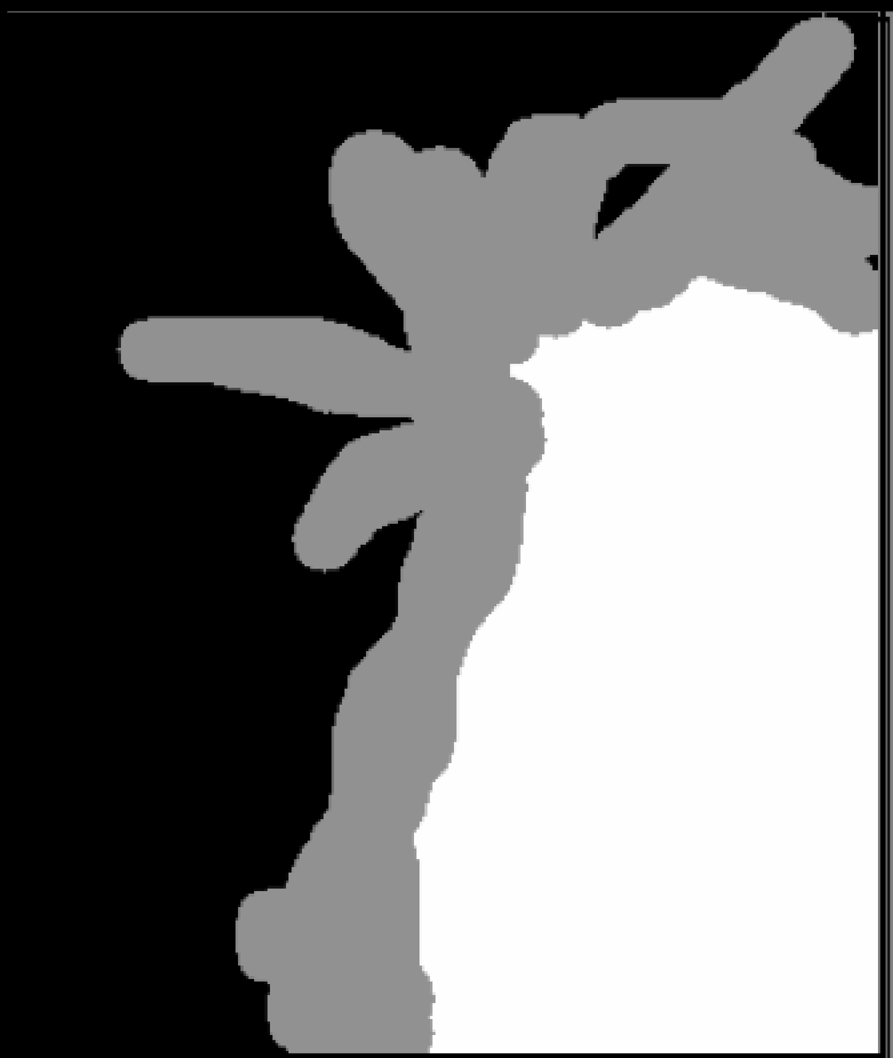
Input image | Input trimap | Alpha matte | Detail | Composite



Images from [Chuang *et al.*, 2001]

From matting to cutting

Trimap

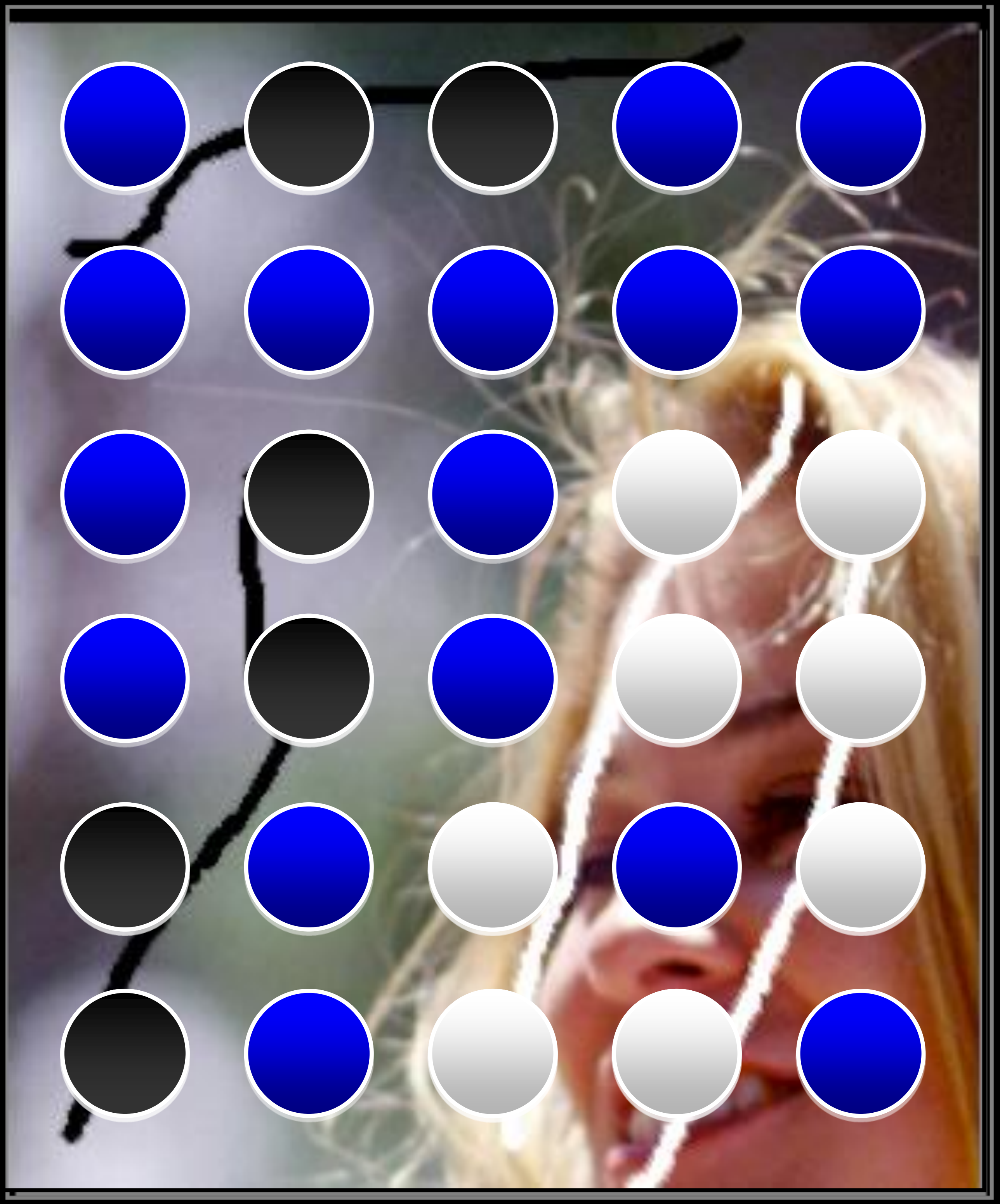


Scribbles

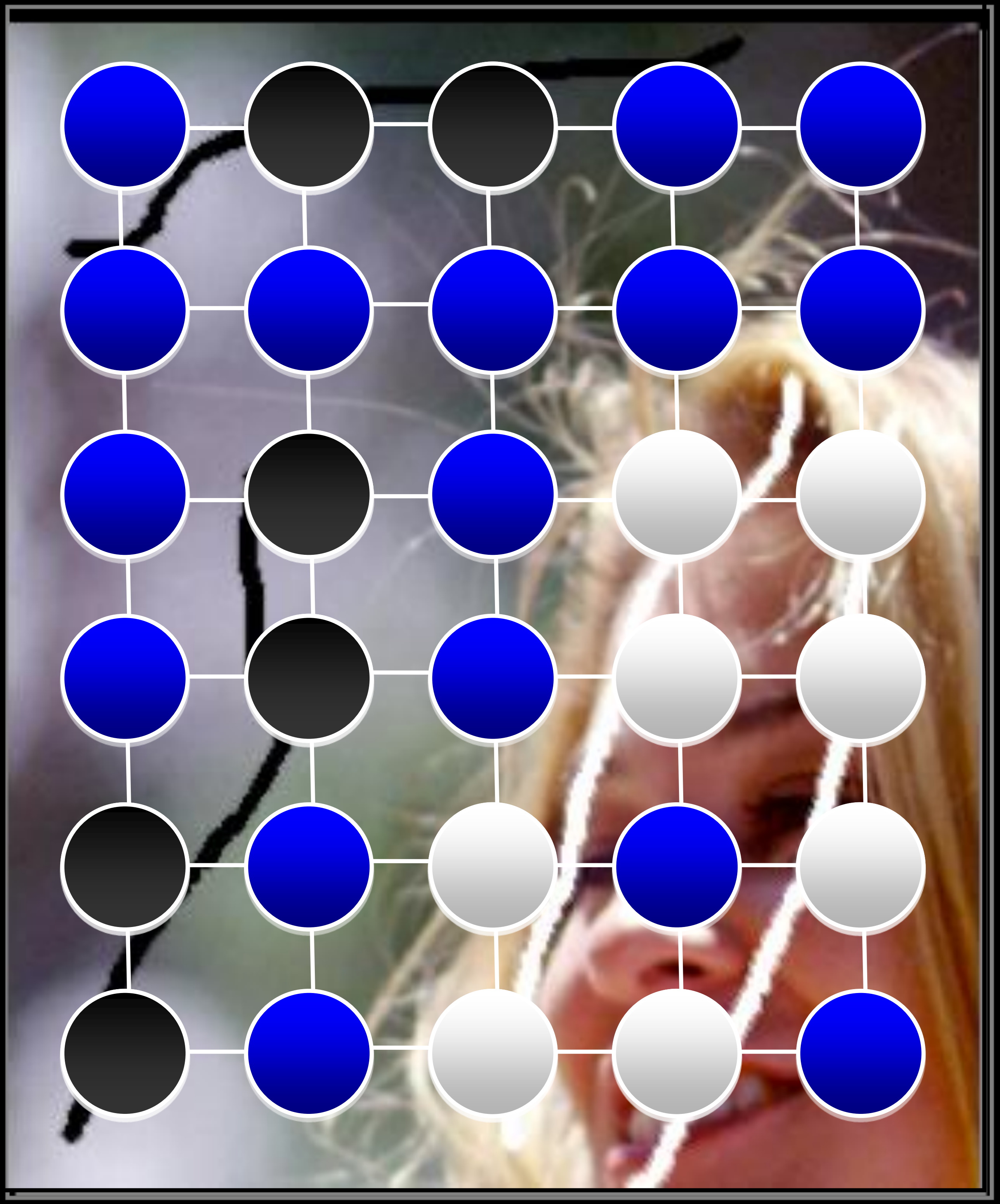
Considerably simpler!

Images from [Levin *et al.*, 2006]

Cutting

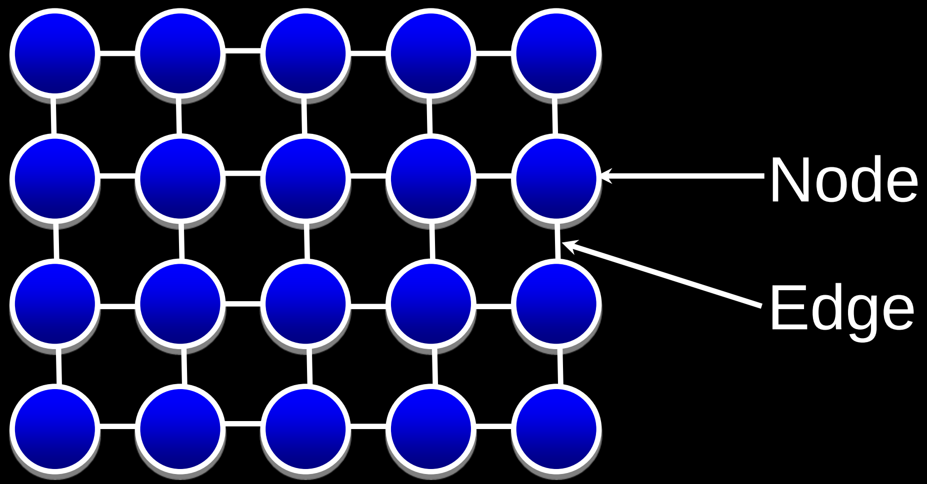


Cutting



Graph cut

- Problem definition:
  - Graph
    - (Not necessarily 2-D lattice)
    - Assign label to each node



  - Energy function

Label of node i

Labels of nodes i and j

$$E(X) = \sum_{i \in \mathcal{V}} E_1(x_i) + \lambda \sum_{(i,j) \in \mathcal{E}} E_2(x_i, x_j)$$

Unary term

Pairwise term

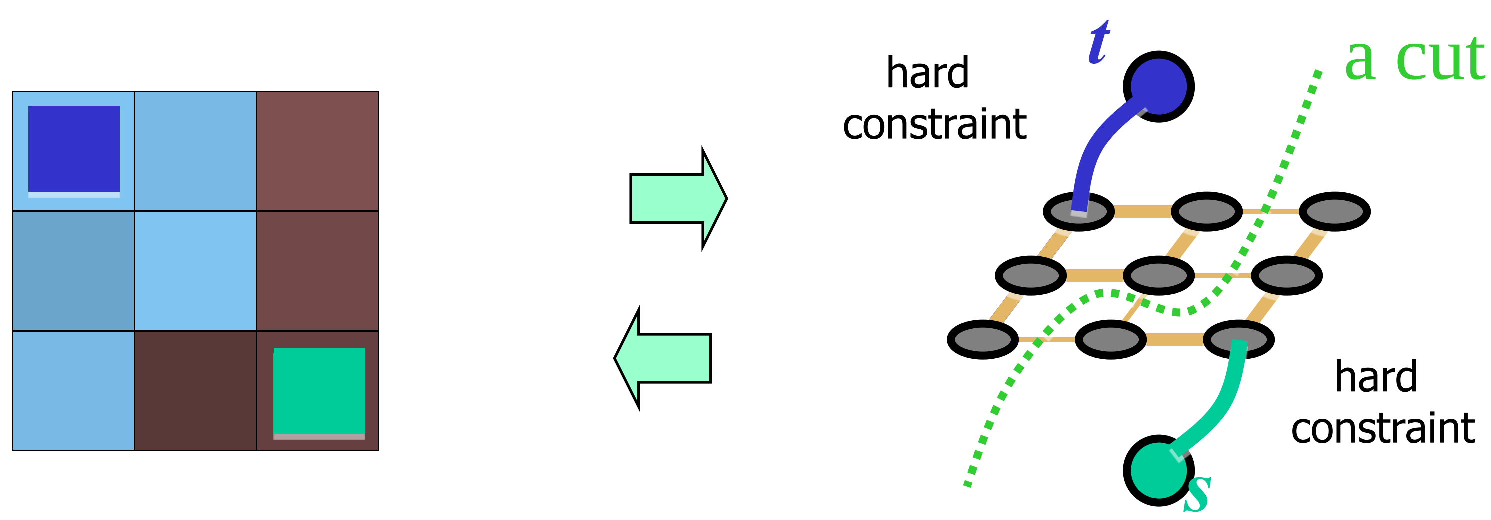
  - Find labeling that minimize energy

Graph cut

- Find labeling that minimizes energy

$$E(X) = \sum_{i \in \mathcal{V}} E_1(x_i) + \lambda \sum_{(i,j) \in \mathcal{E}} E_2(x_i, x_j)$$

- Assign weights and compute min cut



Minimum cost cut can be computed in polynomial time

(max-flow/min-cut algorithms)

Images from [Boykov & Jolly, 2001]

Graph cut for image segmentation

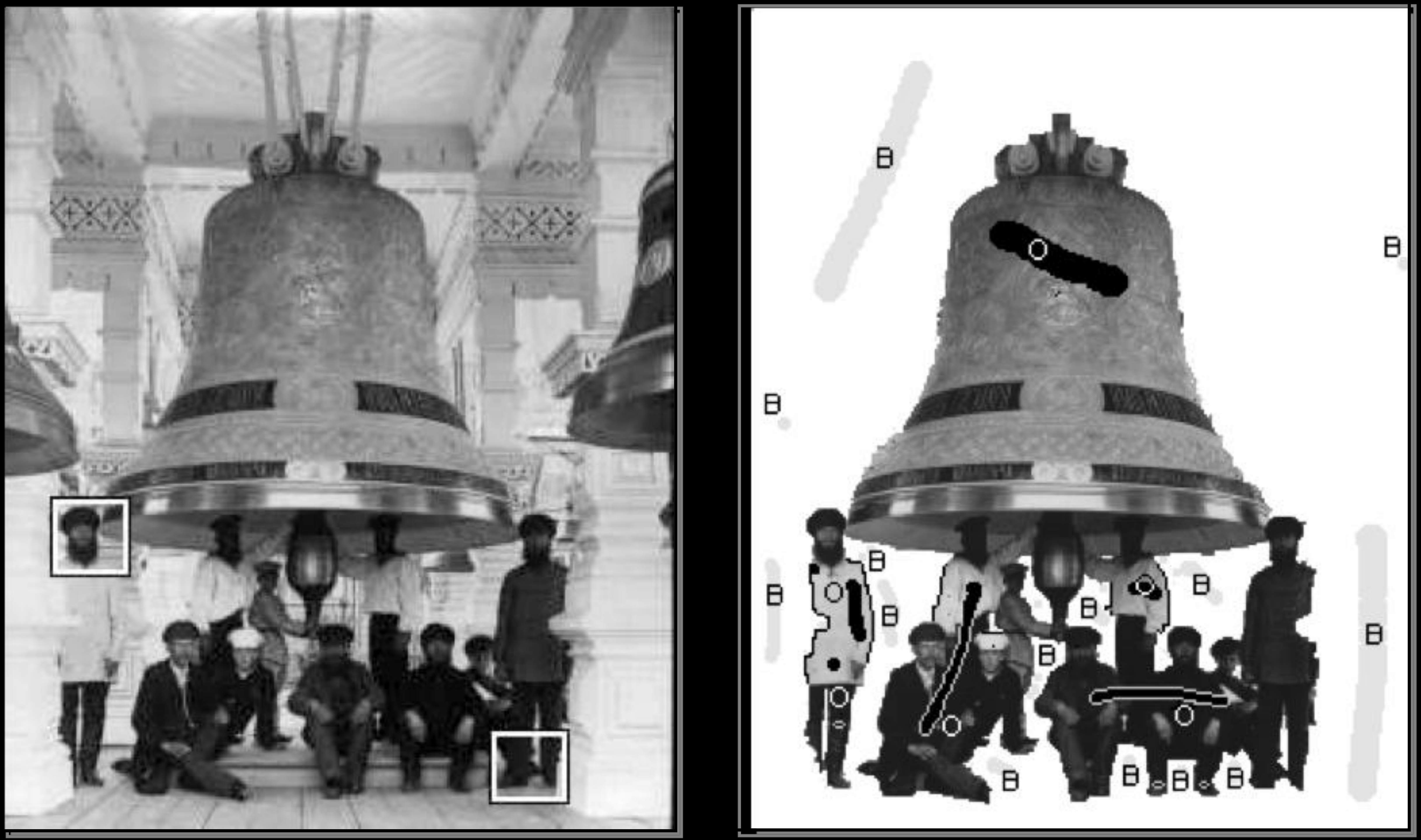
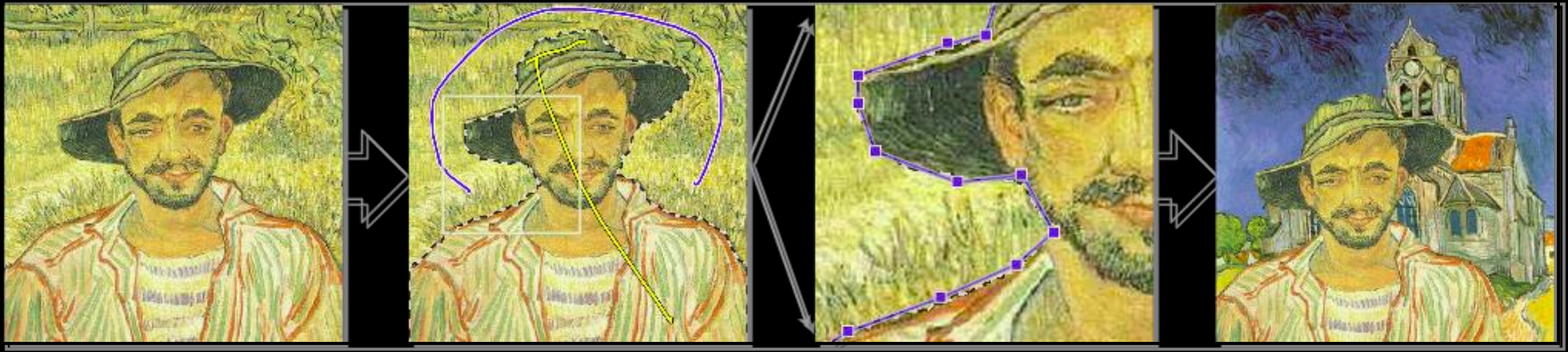


Image from [Boykov & Jolly, 2001]

Using Graph Cut

Lazy Snapping

[Li *et al.*, 2004]



Input Scribbles Refinement Result

- Only works for simple outlines
  - no matting

Image from [Li *et al.*, 2004]

Improvements over Graph Cuts

GrabCut

[Rother *et al.*, 2004]

- Iterative energy minimization
- May require scribbles
- Allow for (slightly) more complex objects
  - Border matting

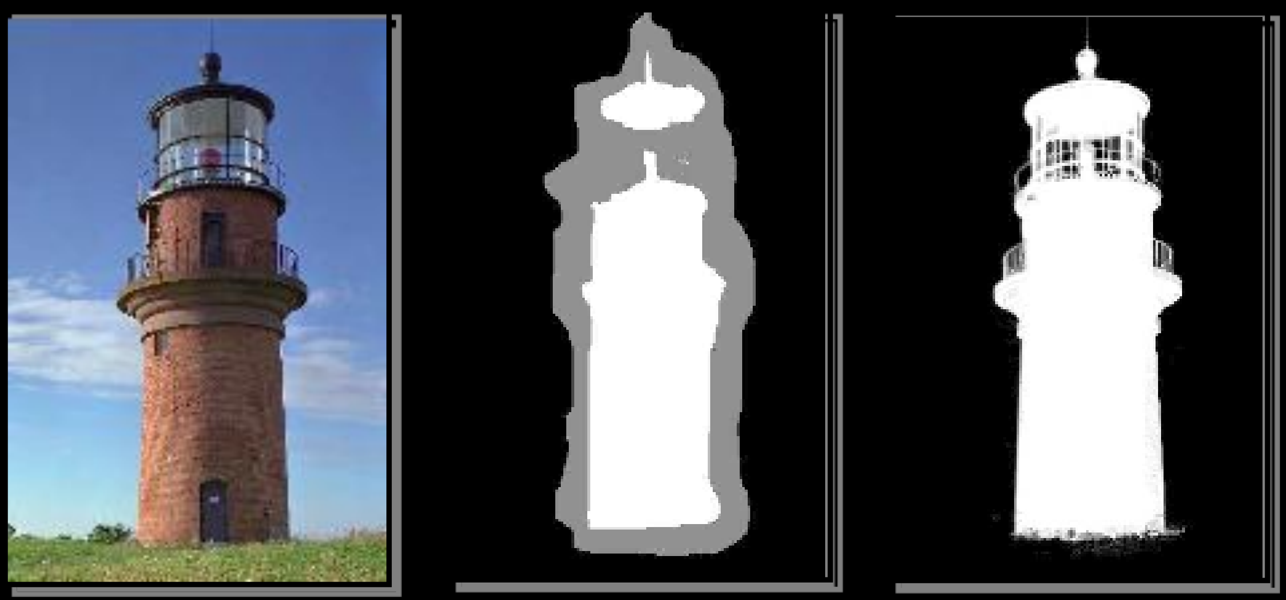
Image from [Rother *et al.*, 2004]

Interactive matting

[Wang *et al.*, 2007]

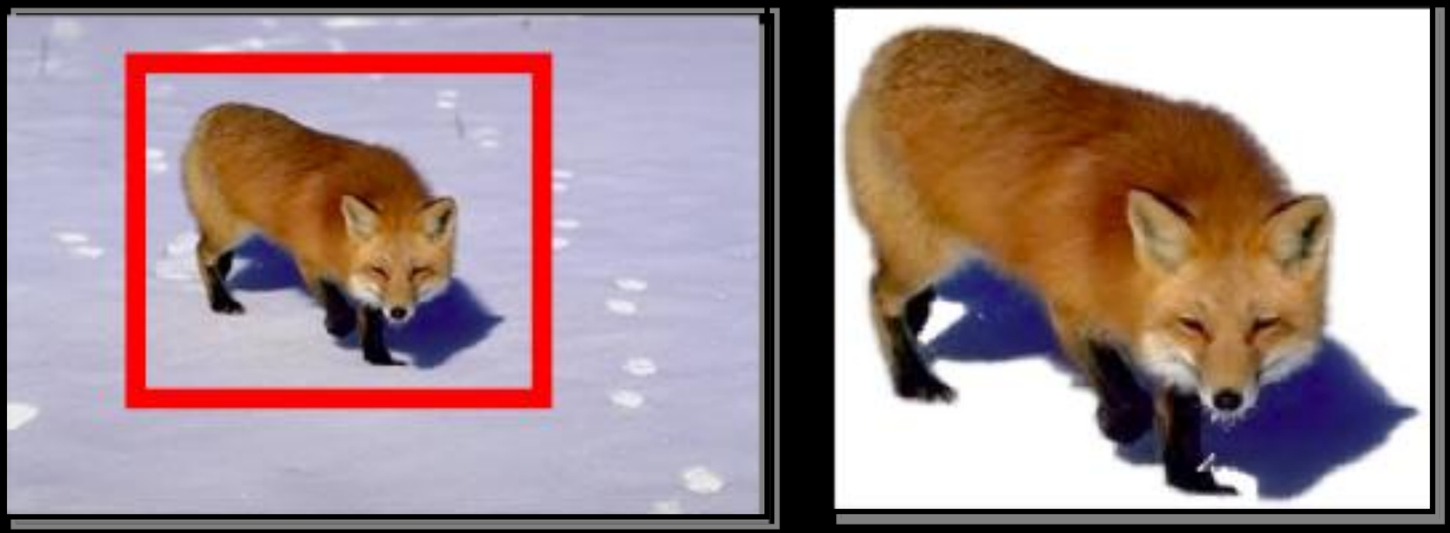
Summary: segmentation

- Matting



- [ ] Complex user interaction
- [x] Complex shapes

Cutting

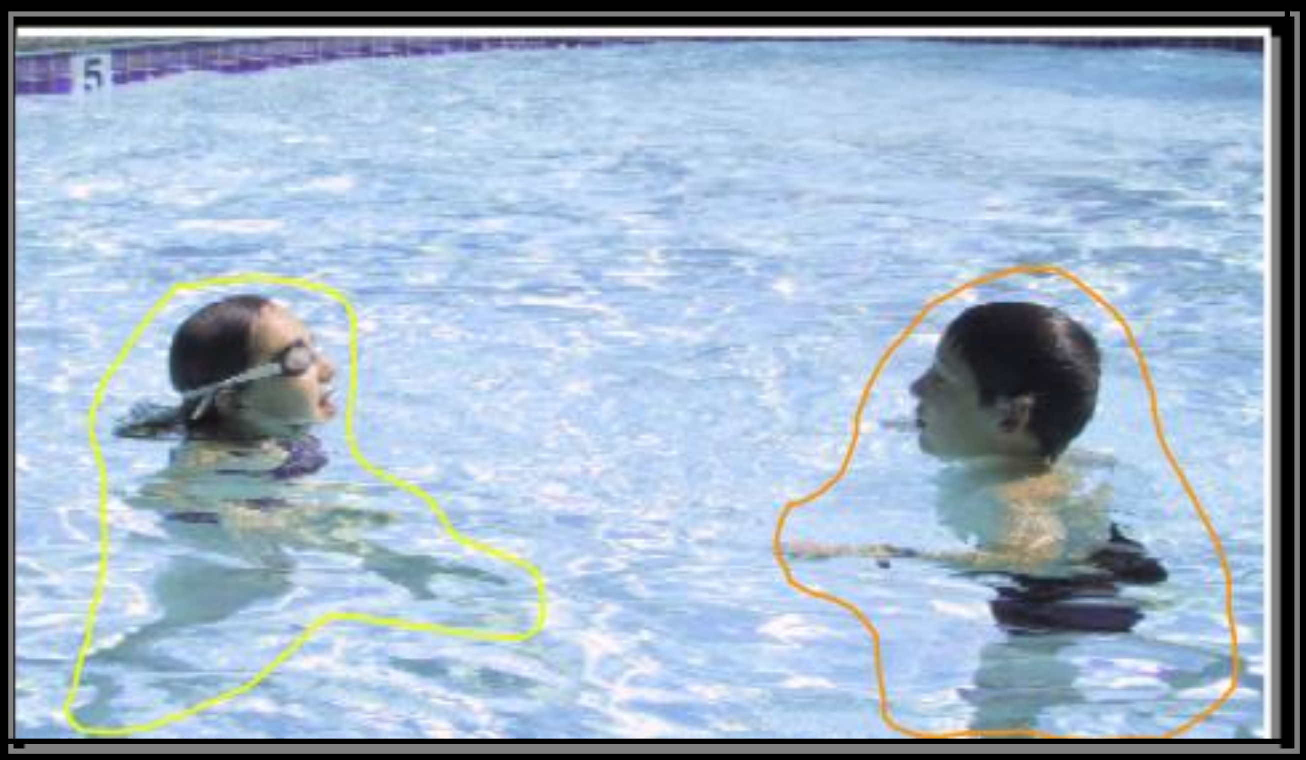


- [ ] Simple shapes
- [x] Simple user interaction

Images from [Chuang *et al.*, 2001] and [Rother *et al.*, 2004]

Considering the destination image

Input



Destination image

Result

Visible seam!

Images from [Perez *et al.*, 2003]

Poisson blending: idea

Input

Destination

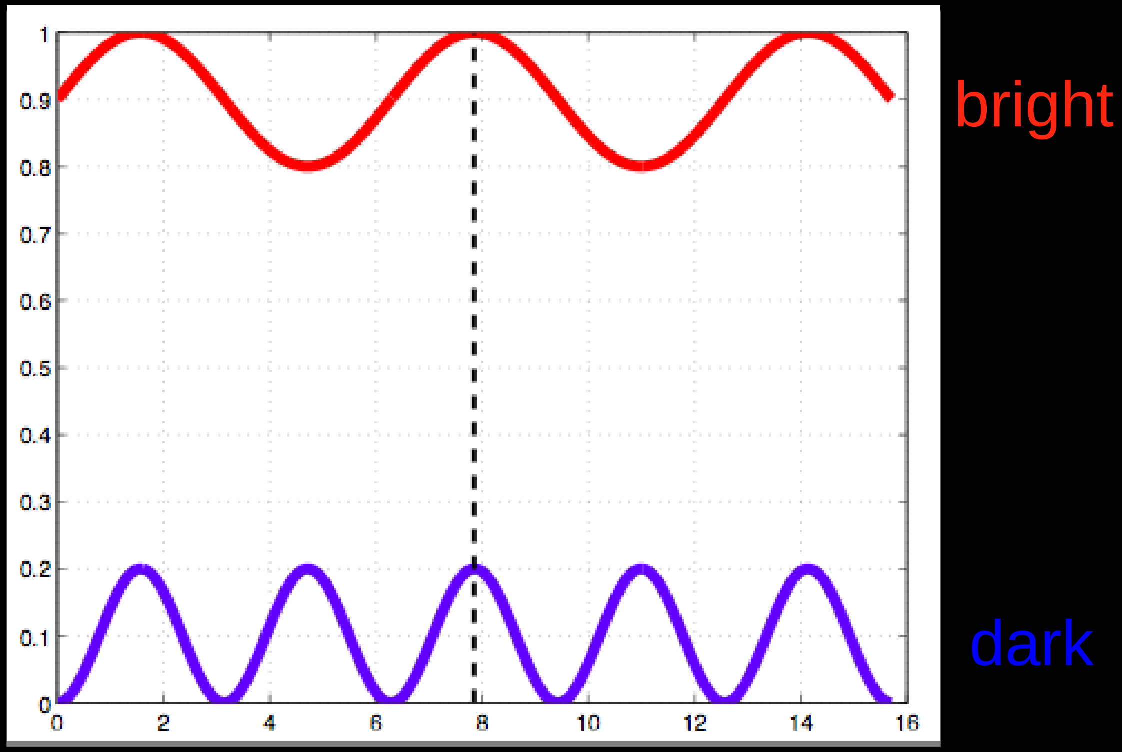
Enforce boundary color (seamless result)

Enforce same gradient than input

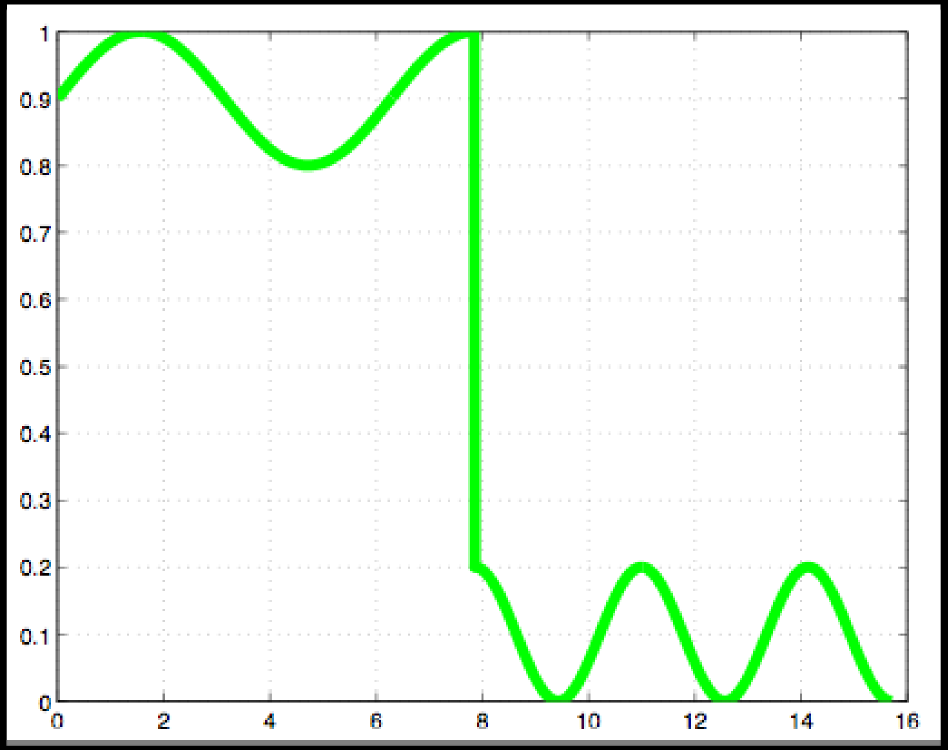
Result

ges from [Perez *et al.*, 2003]

Why gradients?

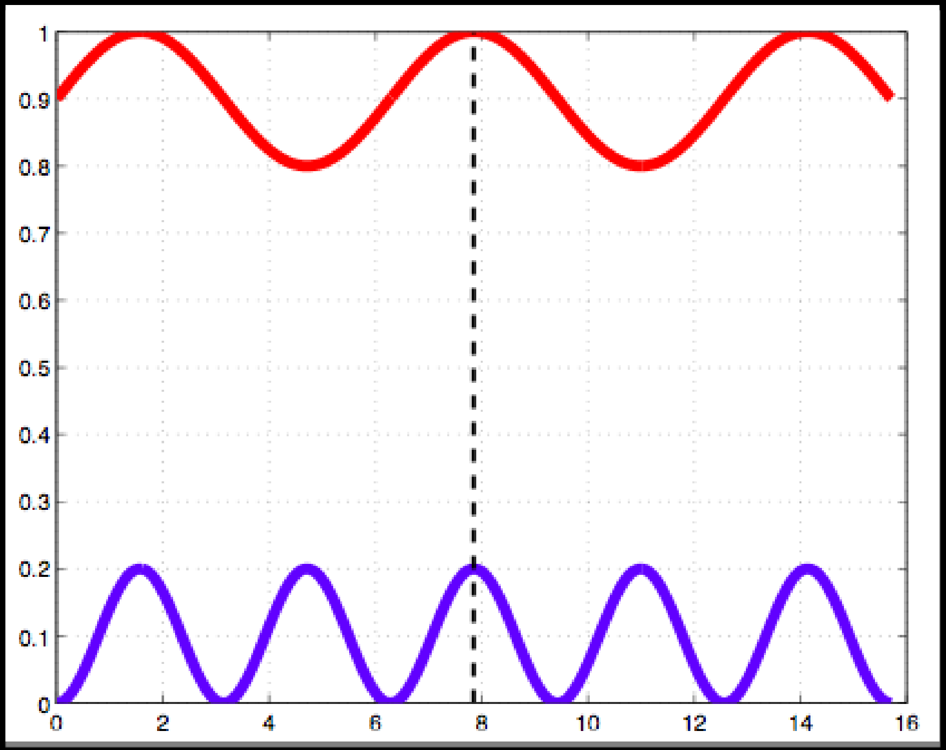


Regular blending

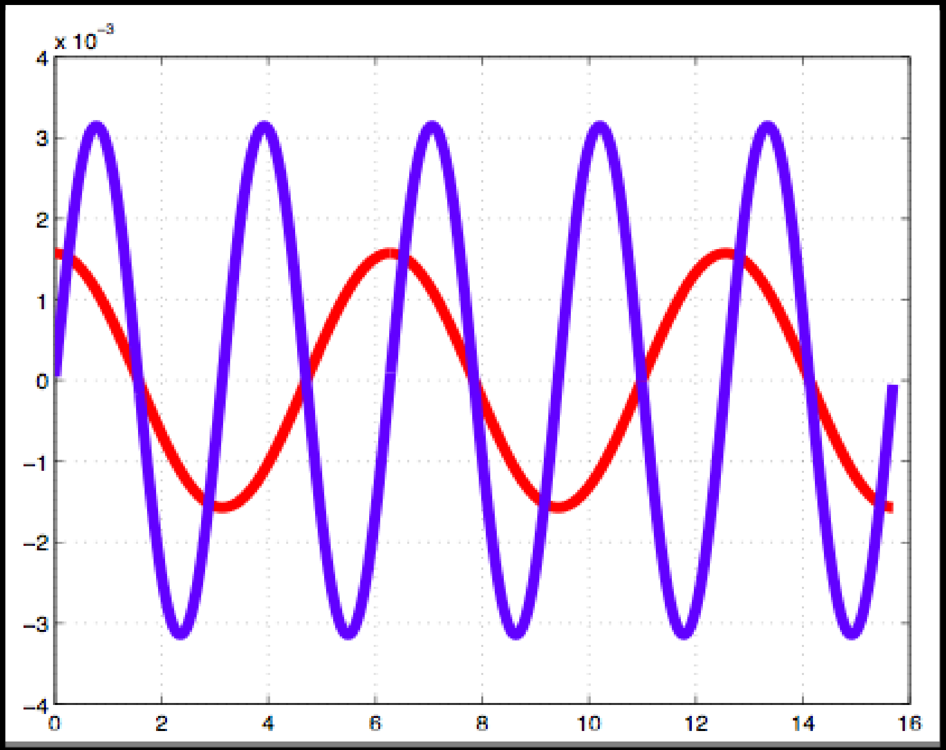


1-D example

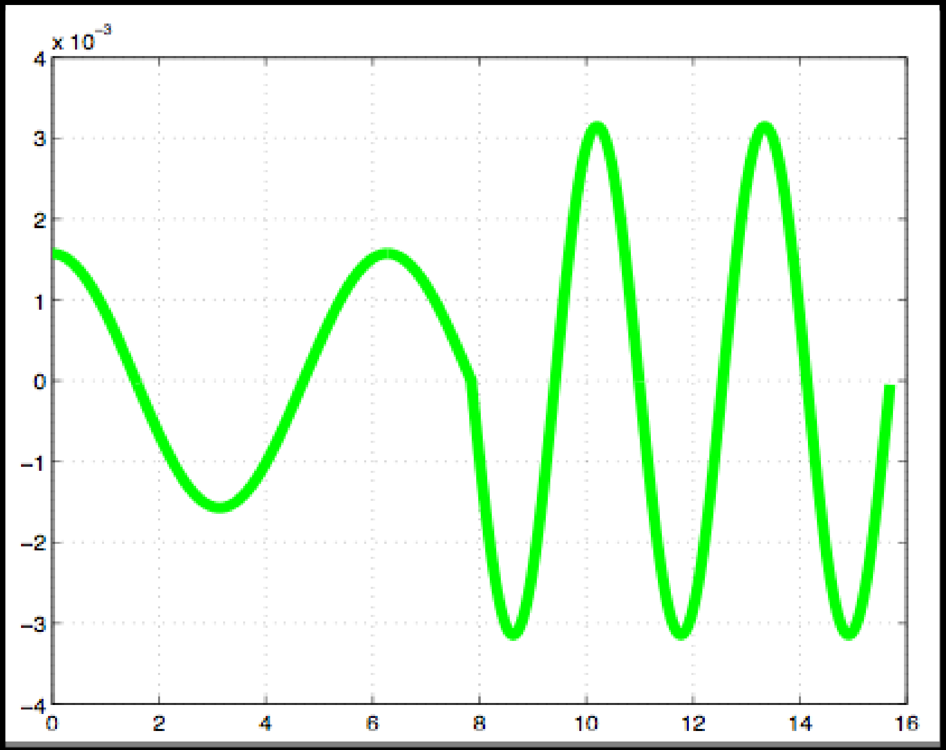
Original signals



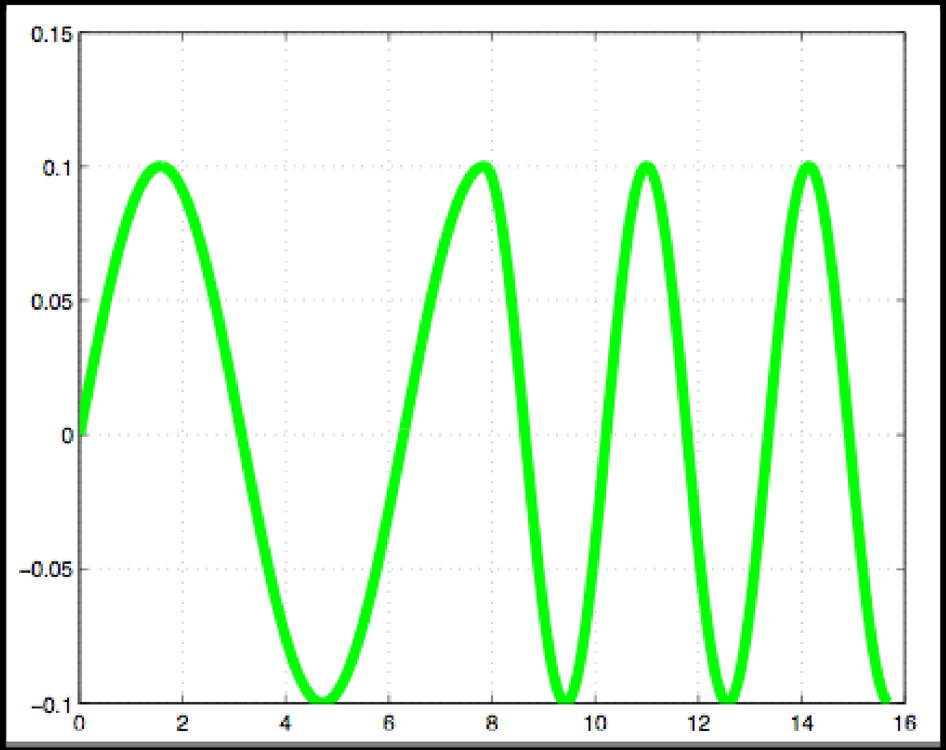
Derivatives



Blending derivatives

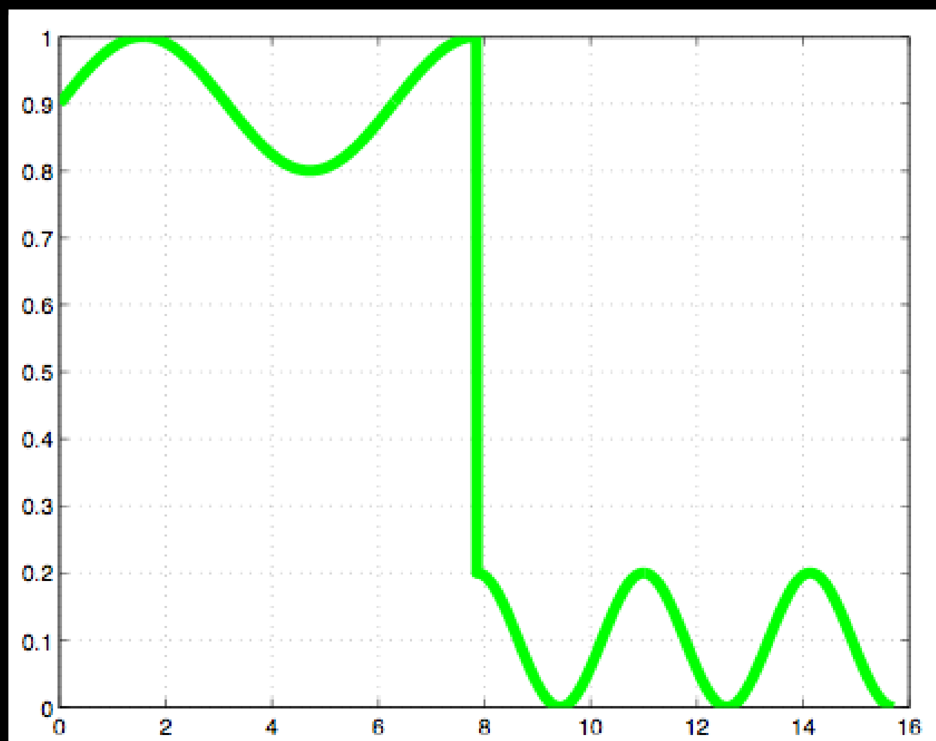


Reintegration results

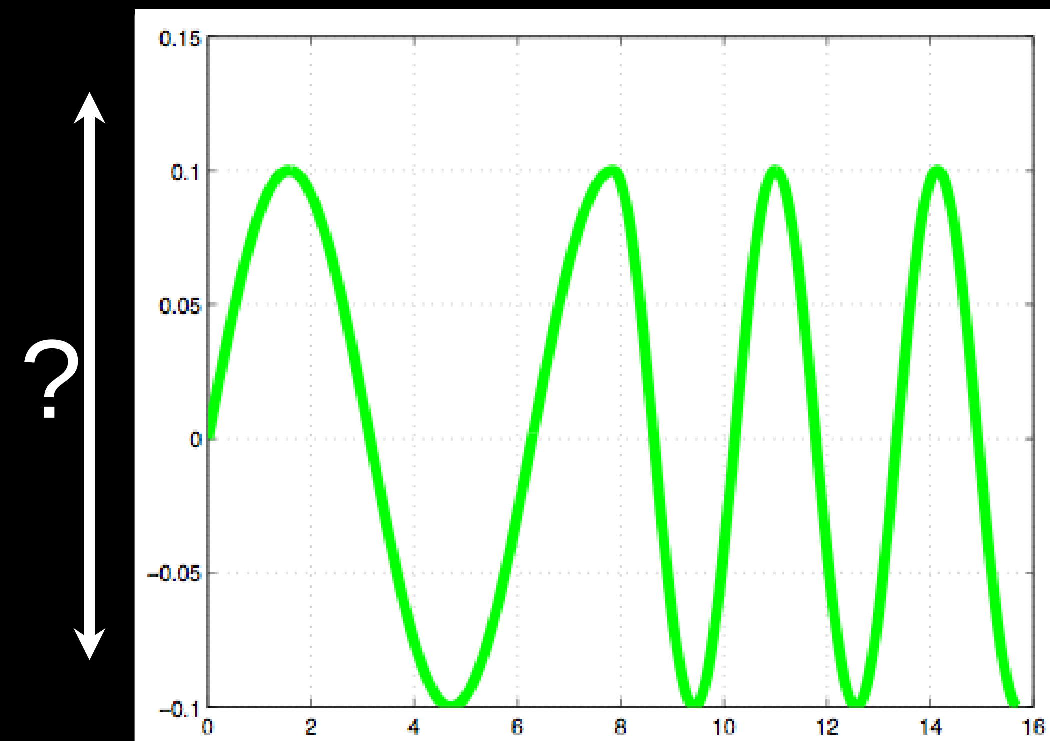


1-D example

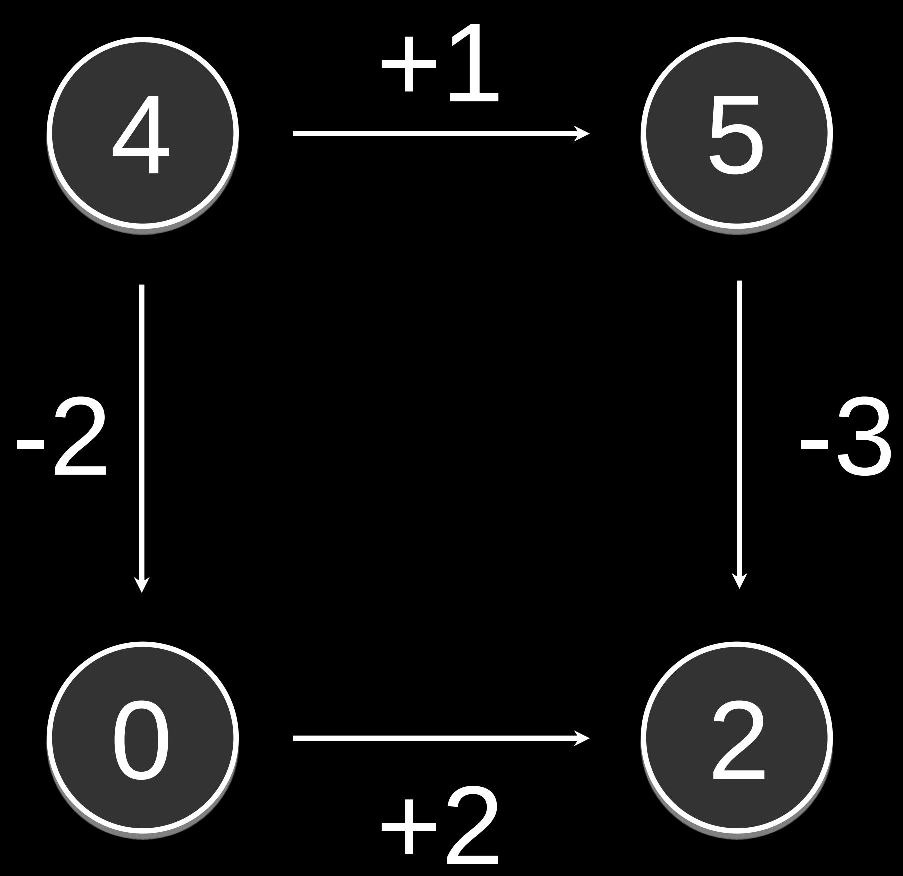
Intensity domain



Gradient domain



2-D: not so easy



Non integrable: sum over a loop ≠ 0

Actually happens all the time in practice

2-D: some notation

I

g_x

g_y

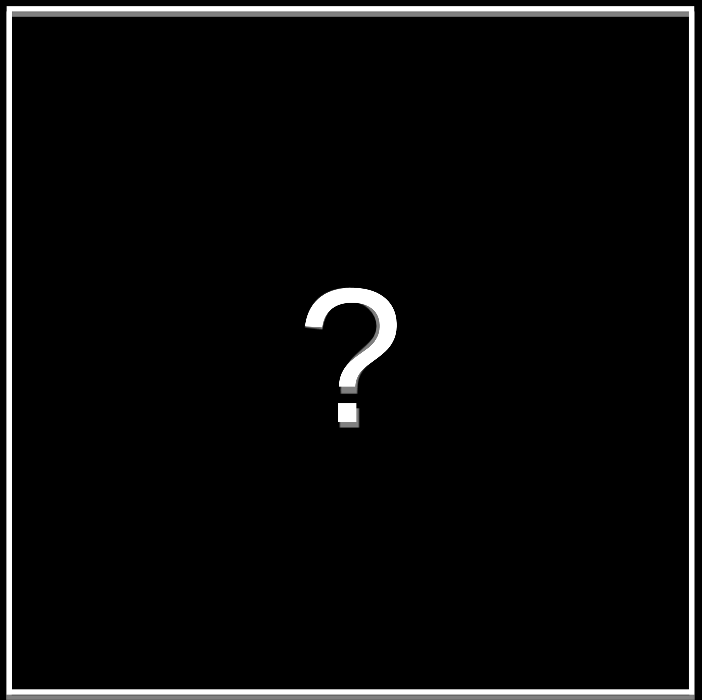
- Finite differences

$$
\begin{aligned}
g_x(x,y) &= I(x+1,y) - I(x,y) \\
g_y(x,y) &= I(x,y+1) - I(x,y)
\end{aligned}
$$

2-D: a (possible) solution

g_x

g_y



F

- Least-squares solution:

$$F^* = \arg\min_F \sum_x (g_x(x,y) - (F(x+1,y) - F(x,y))^2 + \sum_y (g_y(x,y) - (F(x,y+1) - F(x,y))^2$$

2-D: a (popular) solution

$$F^* = \arg\min_F \sum_x (g_x(x,y) - (F(x+1,y) - F(x,y))^2 + \sum_y (g_y(x,y) - (F(x,y+1) - F(x,y))^2$$

- Solution: Poisson equation
- Popular because:
  - Solution is obtained by solving a linear system of equations
  - Can be solved (somewhat) efficiently
    - ‘\’ in matlab
    - FFT
    - Multi-grid solvers (approximate, but really fast!)

Results & limitations

Image editing

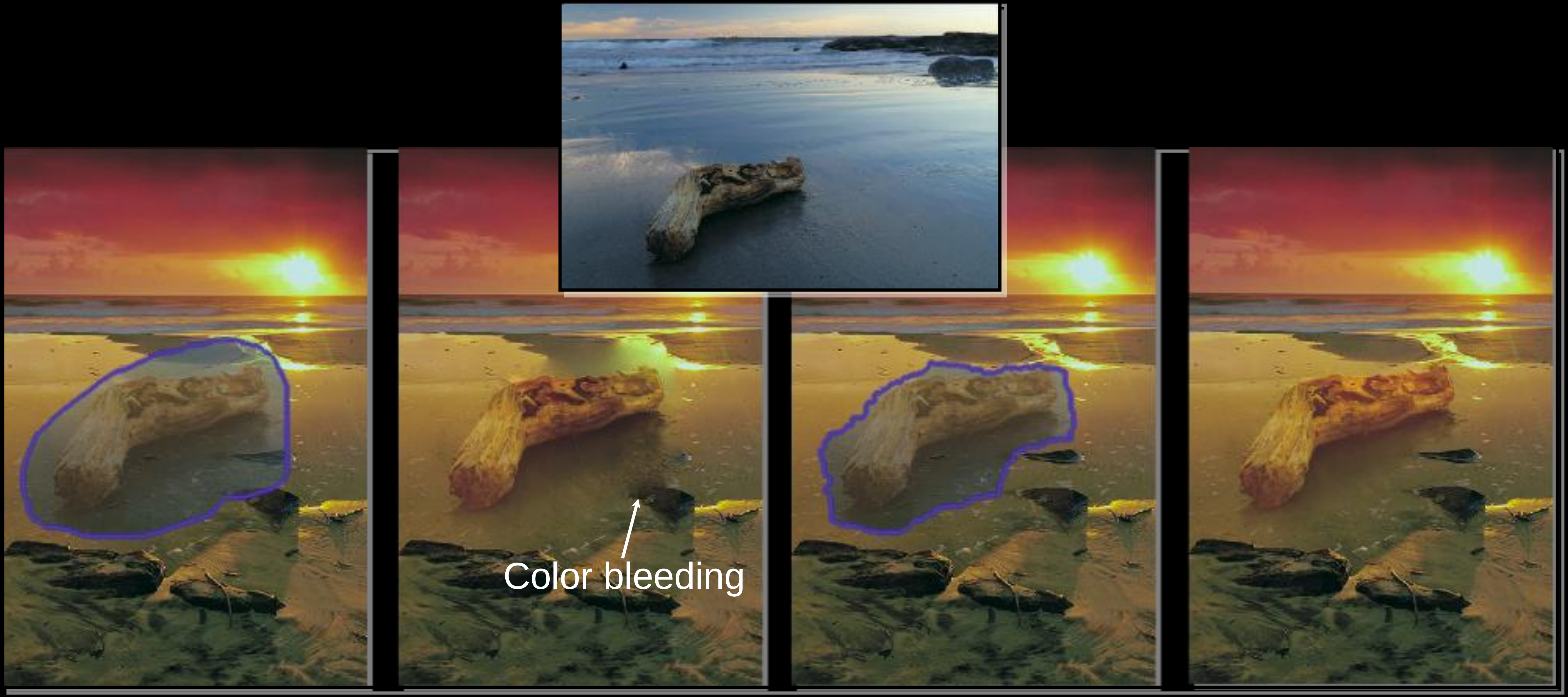
- Some limitations
  - Images need to be very well aligned
  - Differences in background “bleed through”

Images from [Perez *et al.*, 2003]

Poisson blending: improvements

Drag-and-Drop Pasting

[Jia *et al.*, 2006]



User-selected boundary | Poisson blending | Refined boundary | Poisson blending

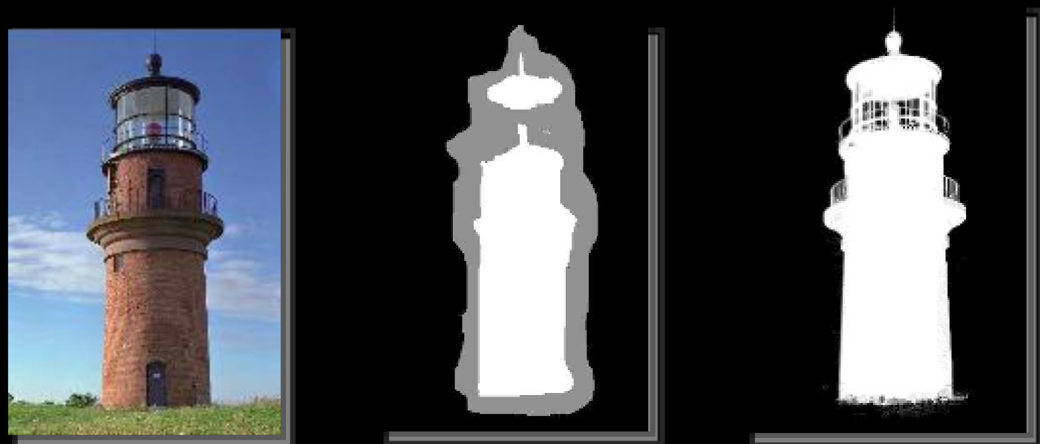
Images from [Jia *et al.*, 2006]

Gradients

- Can do many more cool things!
  - Remove shadows
  - Compress high dynamic range
  - Stitch images together into mosaic
- Even in 3-D
  - Edit meshes
  - Video editing
- ...

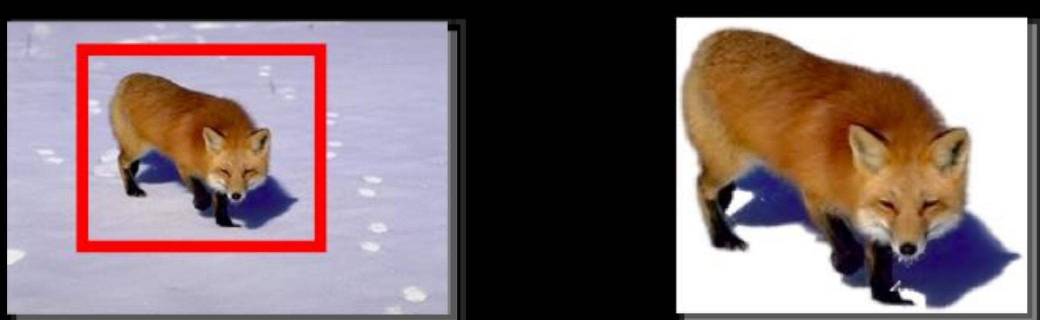
So far...

- Matting



- [x] Complex shapes
- [ ] Complex user interaction

Graph cuts



- [x] Simple user interaction
- [ ] Simple shapes

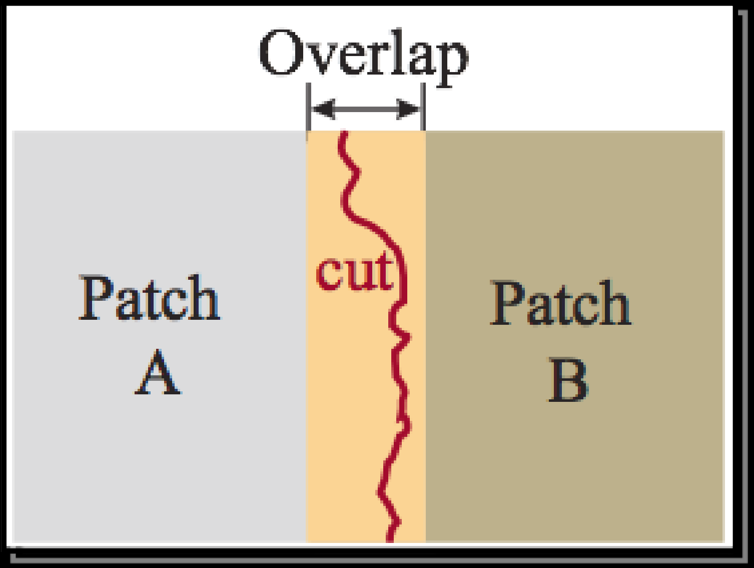
Blending

- [x] Simple UI + recoloring
- [ ] Needs input seam, with similar backgrounds

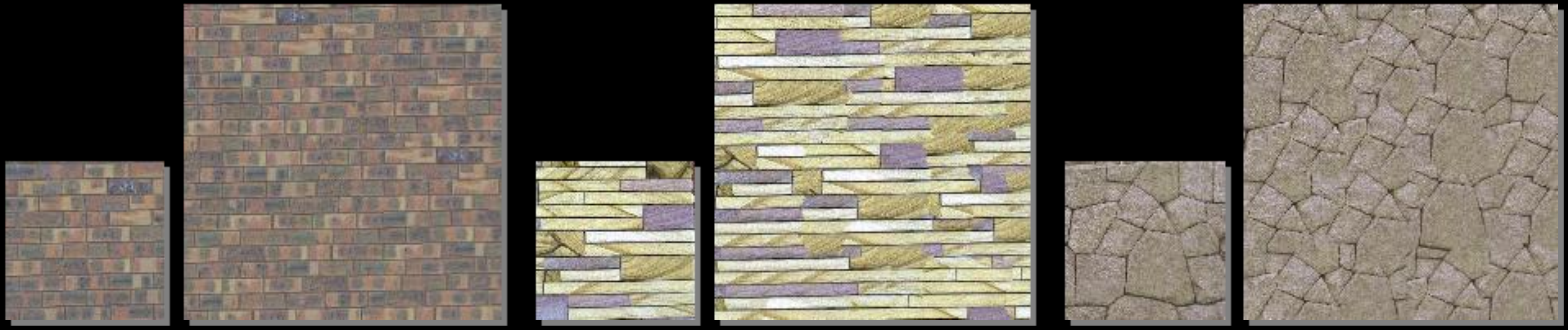
Images from [Chuang *et al.*, 2001], [Rother *et al.*, 2004] and [Jia *et al.*, 2006]

Image stitching: finding best seam

- Do not care about object segmentation

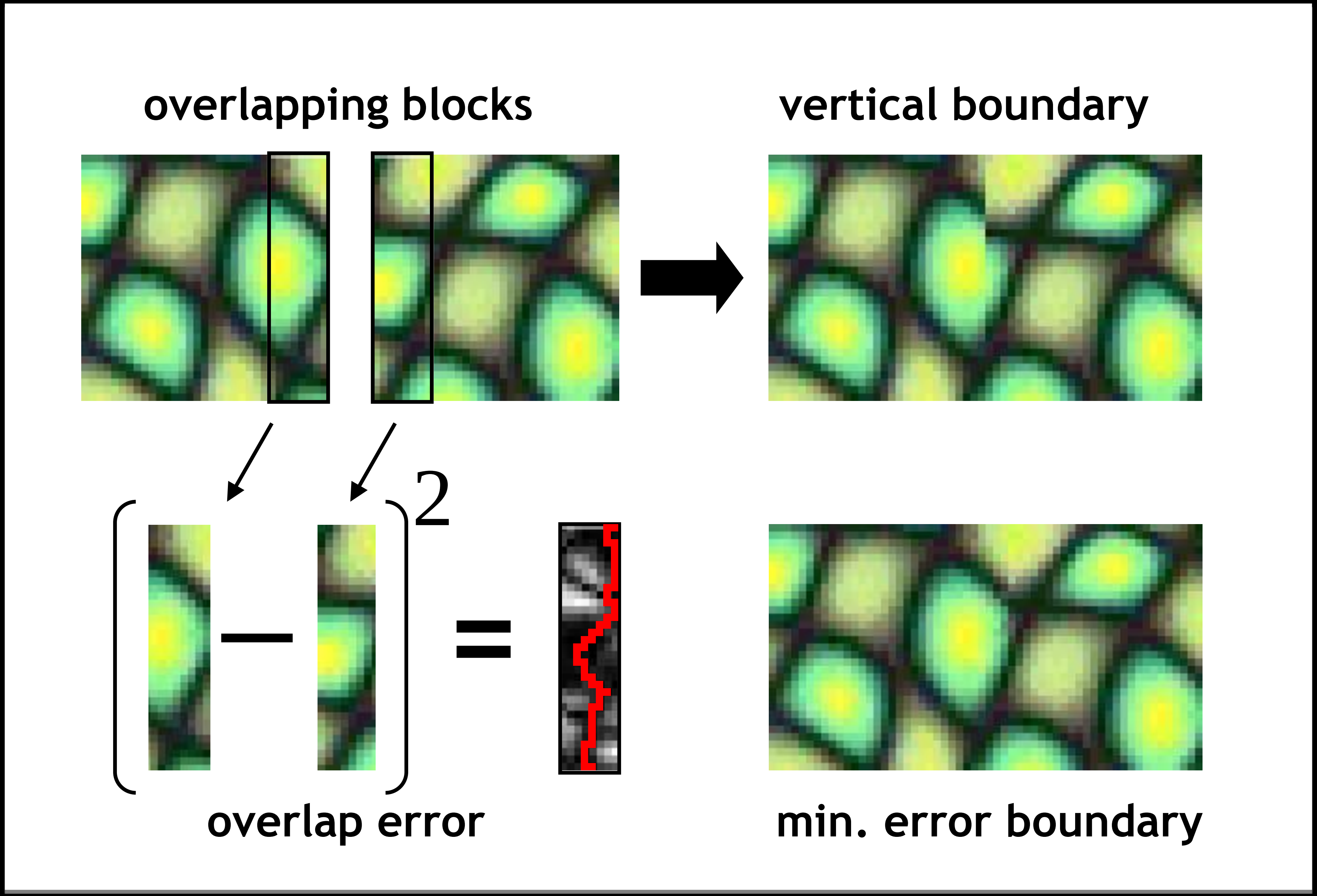


- Image Quilting [Efros & Freeman, 2001]



Images from [Kwatra *et al.*, 2003] and [Efros & Freeman, 2001]

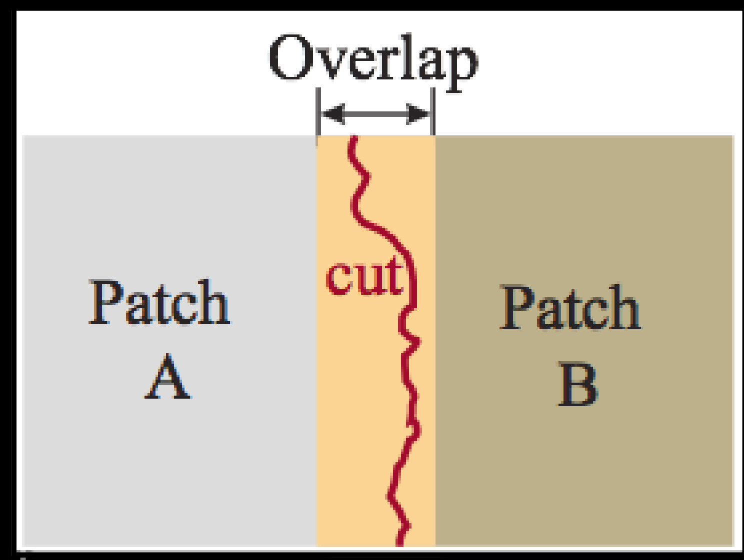
Minimal error boundary



From [Efros & Freeman, 2001]

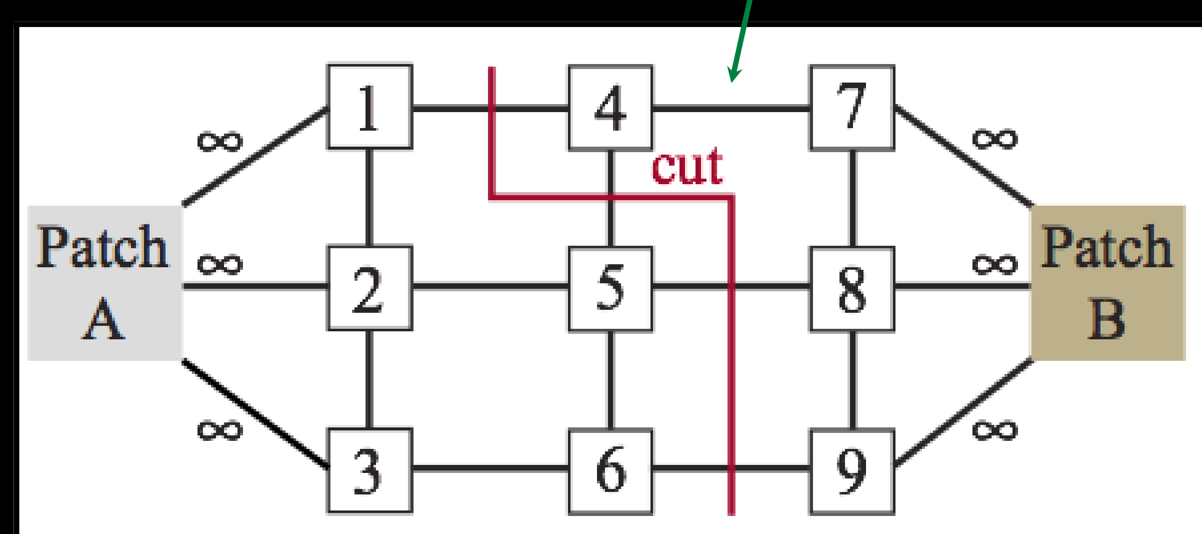
Image stitching: finding best seam

- Dynamic programming can't handle loops



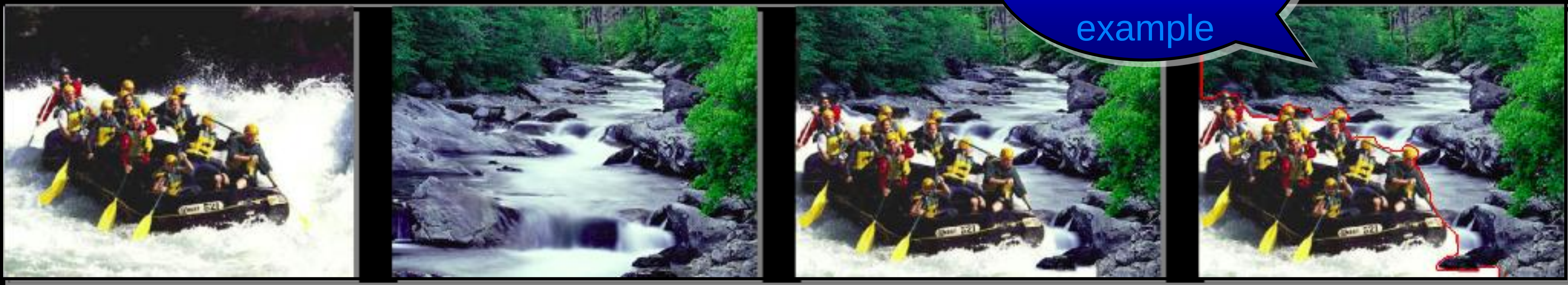
- Graph cut textures [Kwatra *et al.*, 2003]

$$||\mathbf{A}(4) - \mathbf{B}(4)|| + ||\mathbf{A}(7) - \mathbf{B}(7)||$$



Images from [Kwatra *et al.*, 2003]

Image stitching



Side note: they also use this for texture synthesis

Images from [Kwatra *et al.*, 2003]

Putting it all together

Interactive Digital Photomontage

[Agarwala *et al.*, 2004]

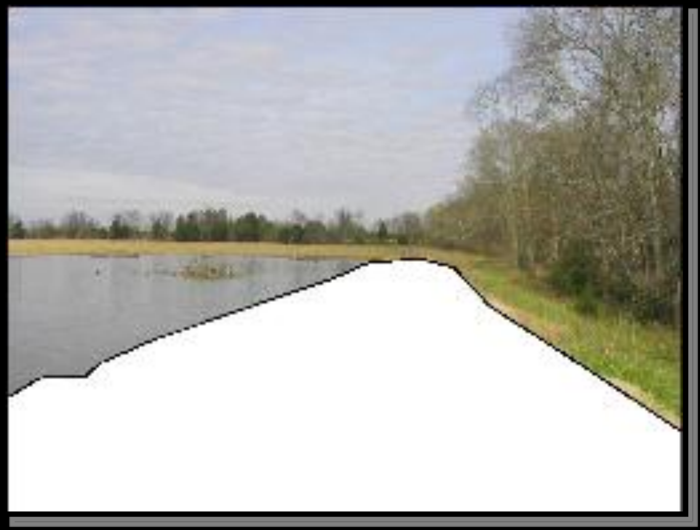
Now what?

- What if the best seam is still bad?
  - We’re approaching the limits of matting/cutting/blending
  - Remember: under-constrained problem!
- Say we have a large image database

Finding the best match

Database Driven Image Completion

[Hays & Efros, 2007]



Input

Potential matches

Output

Images from [Hays & Efros, 2007]

Finding the best match

Photo Clip Art

[Lalonde *et al.*, 2007]

So far

Segmentation

- Matting

Graph cuts

Blending

Seam finding

Framework

Databases

Database Driven Techniques

Thank you!

Some references...

- [Boykov & Jolly, 2001] Boykov, Y. and Jolly, M.-P. Interactive Graph Cuts for Optimal Boundary & Region Segmentation in N-D images, ICCV 2001
- [Chuang *et al.*, 2001] Chuang, Y.-Y., Curless, B., Salesin, D.H. and Szeliski, R. A Bayesian Approach to Digital Mapping. CVPR 2001
- [Efros & Freeman, 2001] Efros, A.A. and Freeman, W. Image Quilting for Texture Synthesis and Transfer. SIGGRAPH 2001
- [Hays & Efros 2007] Hays, J. and Efros, A.A. Scene Completion using Millions of Photographs, SIGGRAPH 2007
- [Jia *et al*., 2006] Jia, J., Sun, J., Tang, C.-K. and Shum, H.-Y., Drag-and-Drop Pasting. SIGGRAPH 2006
- [Kwatra *et al.*, 2003] Kwatra, V., Schold, A., Essa, I., Turk, G. and Bobick, A. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. SIGGRAPH 2003
- [Lalonde *et al.*, 2007] Lalonde, J.-F., Hoiem, D., Efros A.A., Rother, C., Winn J., and Criminisi, A. Photo Clip Art. SIGGRAPH 2007
- [Levin *et al.*, 2004] Levin, A., Zomet, A., Peleg, S. and Weiss, Y. Seamless image stitching in the gradient domain. ECCV 2004
- [Levin *et al.*, 2006] Levin, A., Lichinski, D. and Weiss, Y. A Closed Form Solution to Natural Image Parsing. CVPR 2006