Graph Cut

Digital Visual Effects

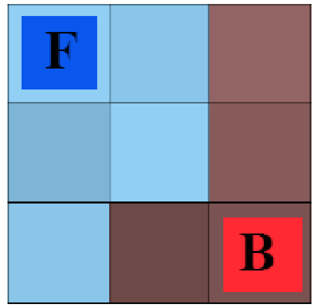
Yung-Yu Chuang

with slides by Fredo Durand, Ramesh Raskar

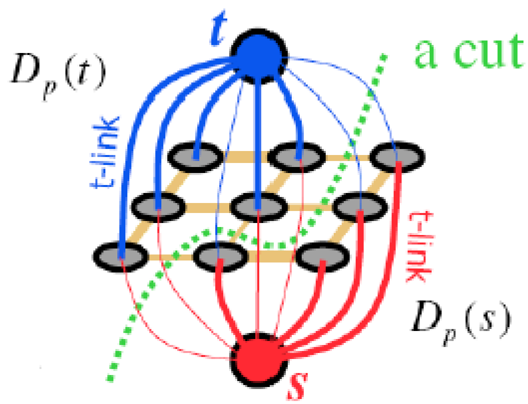
Graph cut

Graph cut

- Interactive image segmentation using graph cut
- Binary label: foreground vs. background
- User labels some pixels
  - similar to trimap, usually sparser
- Exploit
  - Statistics of known Fg & Bg
  - Smoothness of label
- Turn into discrete graph optimization
  - Graph cut (min cut / max flow)

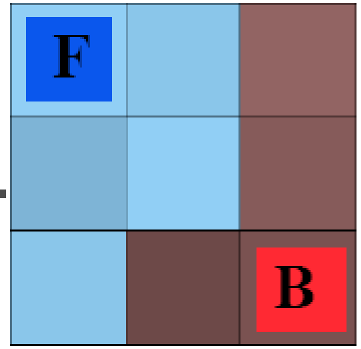


F	F	B
F	F	B
F	B	B

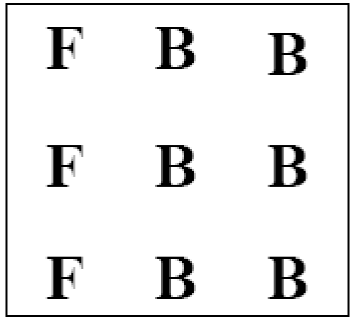


Energy function

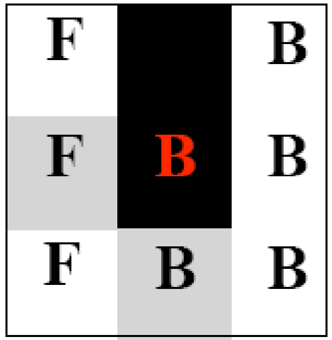
- Labeling: one value per pixel, F or B
- Energy(labeling) = data + smoothness
  - Very general situation
  - Will be minimized
- Data: for each pixel
  - Probability that this color belongs to F (resp. B)
  - Similar in spirit to Bayesian matting
- Smoothness (aka regularization): per neighboring pixel pair
  - Penalty for having different label
  - Penalty is downweighted if the two pixel colors are very different
  - Similar in spirit to bilateral filter



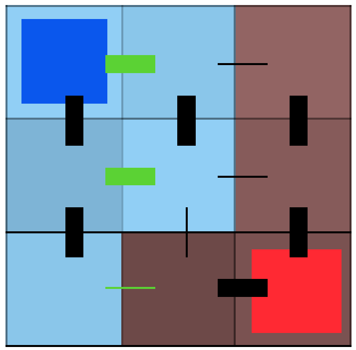
One labeling (ok, not best)



Data

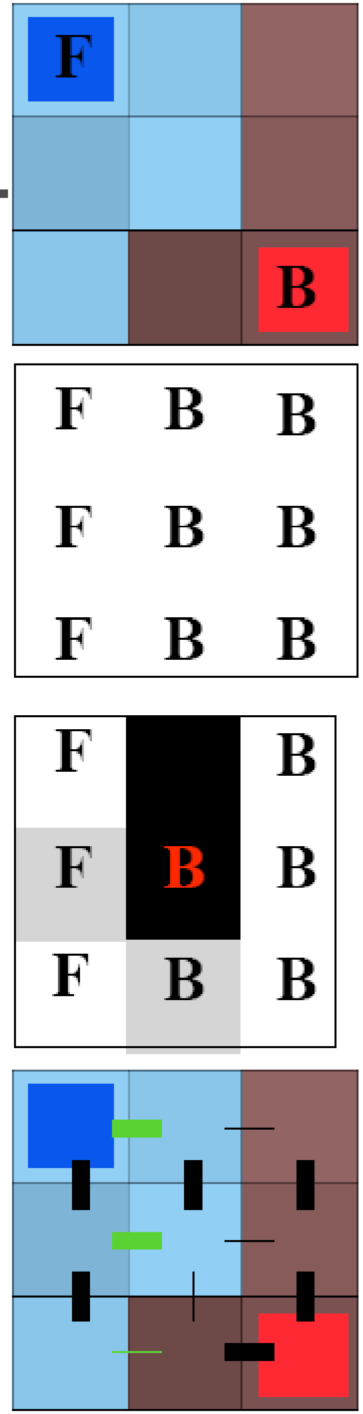


Smoothness



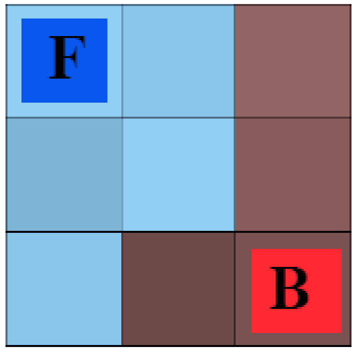
Data term

- A.k.a regional term
  (because integrated over full region)
- $D(L)=\Sigma_i -\log h[L_i](C_i)$
- Where *i* is a pixel
  L_i is the label at *i* (F or B),
  C_i is the pixel value
  $h[L_i]$ is the histogram of the observed Fg (resp Bg)
- Note the minus sign



Hard constraints

- The user has provided some labels
- The quick and dirty way to include constraints into optimization is to replace the data term by a huge penalty if not respected.
- D(L_i)=0 if respected
- D(L_i)=K if not respected
  - e.g. K=- #pixels



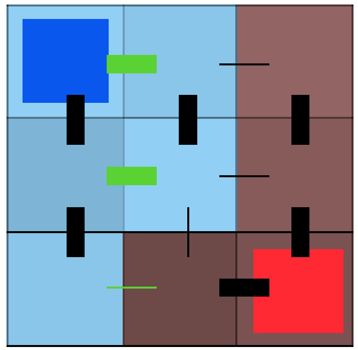
Smoothness term

DigiVFX

- a.k.a boundary term, a.k.a. regularization
- $S(L)=\sum_{\{j,\ i\}\ \text{in}\ N} B(C_i,C_j)\ \delta(L_i-L_j)$
- Where i,j are neighbors
  - e.g. 8-neighborhood (but I show 4 for simplicity)

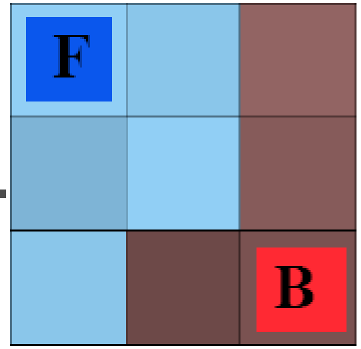
F	B	B
F	B	B
F	B	B

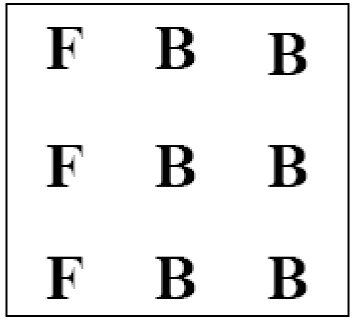
- $\delta(L_i-L_j)$ is 0 if $L_i=L_j$, 1 otherwise
- $B(C_i,C_j)$ is high when C_i and C_j are similar, low if there is a discontinuity between those two pixels
  - e.g. $\exp(-||C_i-C_j||^2/2\sigma^2)$
  - where σ can be a constant or the local variance
- Note positive sign

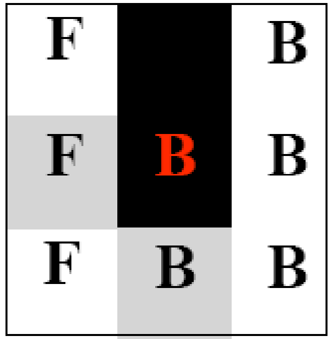


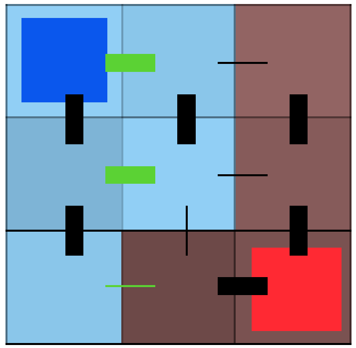
Optimization

- E(L)=D(L)+λ S(L)
- λ is a black-magic constant
- Find the labeling that minimizes E
- In this case, how many possibilities?
  - 2^9 (512)
  - We can try them all!
  - What about megapixel images?



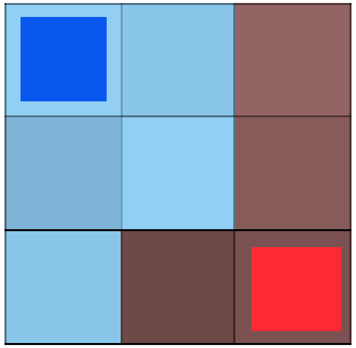




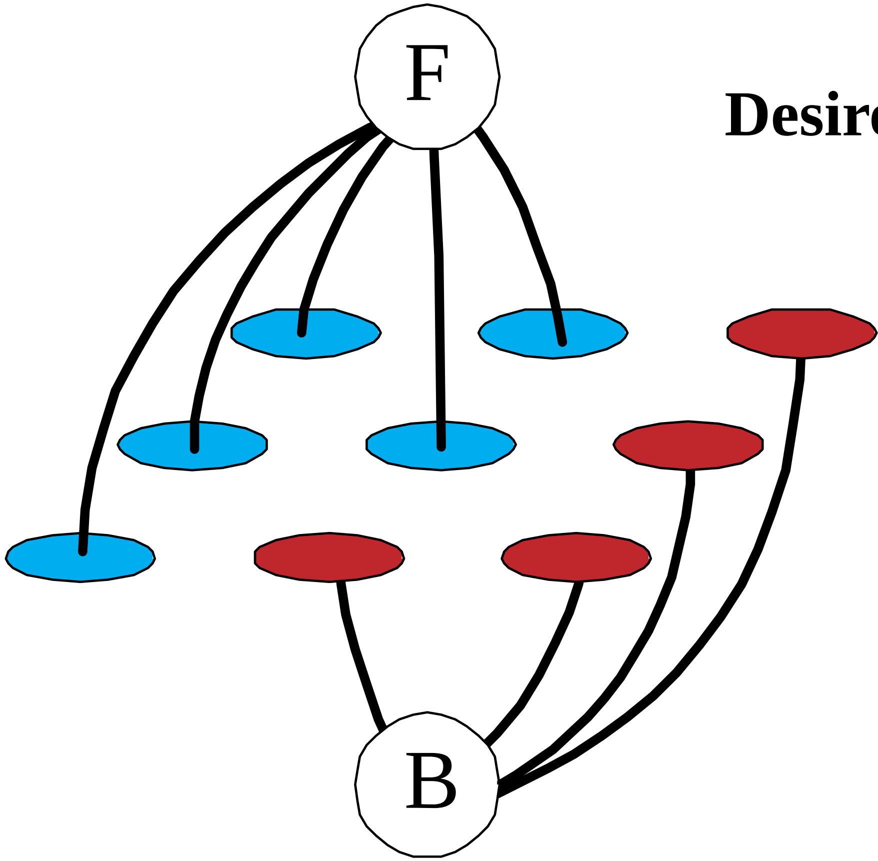


Labeling as a graph problem

- Each pixel = node
- Add two nodes F & B
- Labeling: link each pixel to either F or B



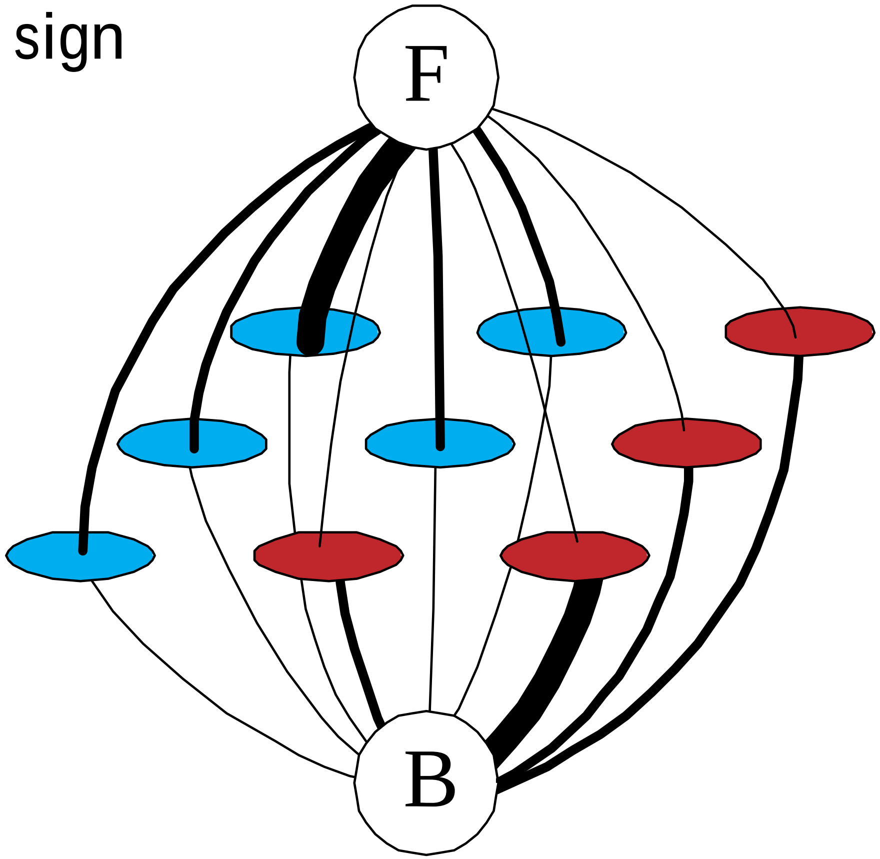
F	F	B
F	F	B
F	B	B



Desired result

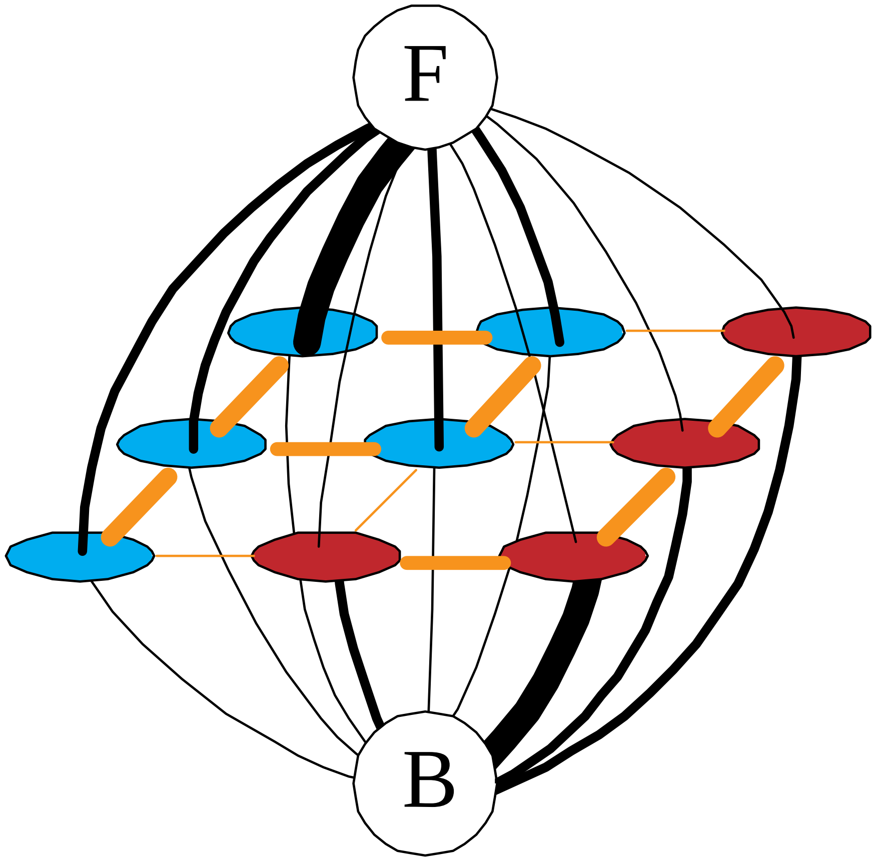
Data term

- Put one edge between each pixel and F & G
- Weight of edge = minus data term
  - Don't forget huge weight for hard constraints
  - Careful with sign



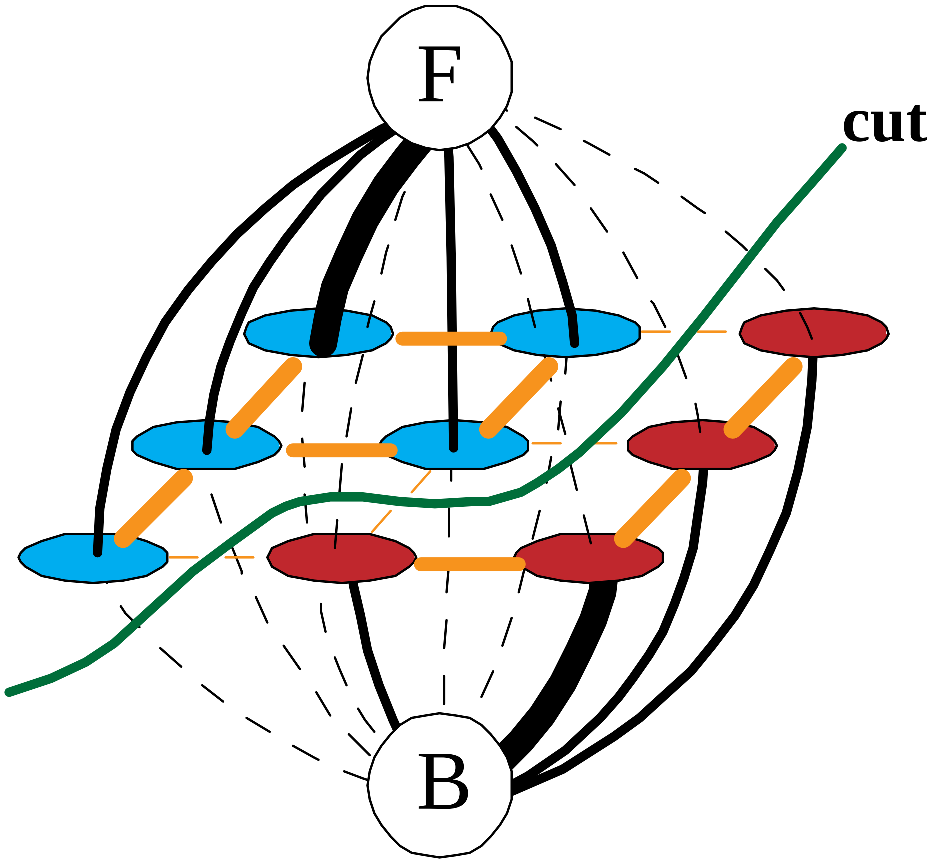
Smoothness term

- Add an edge between each neighbor pair
- Weight = smoothness term



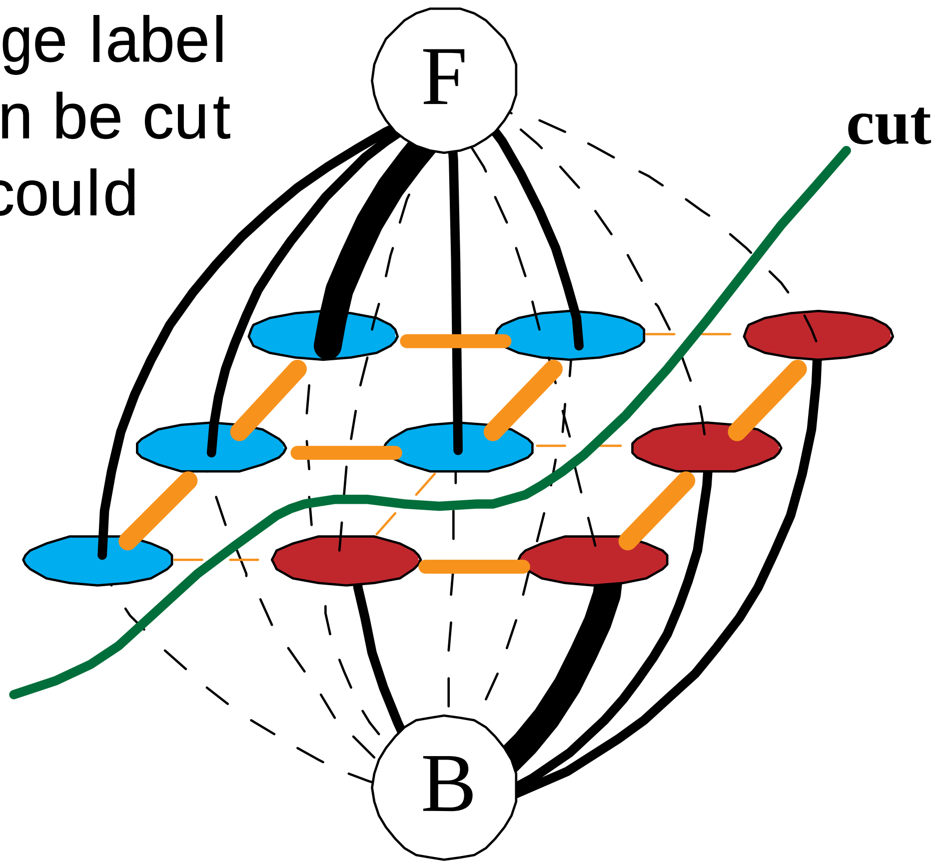
Min cut

- Energy optimization equivalent to min cut
- Cut: remove edges to disconnect F from B
- Minimum: minimize sum of cut edge weight

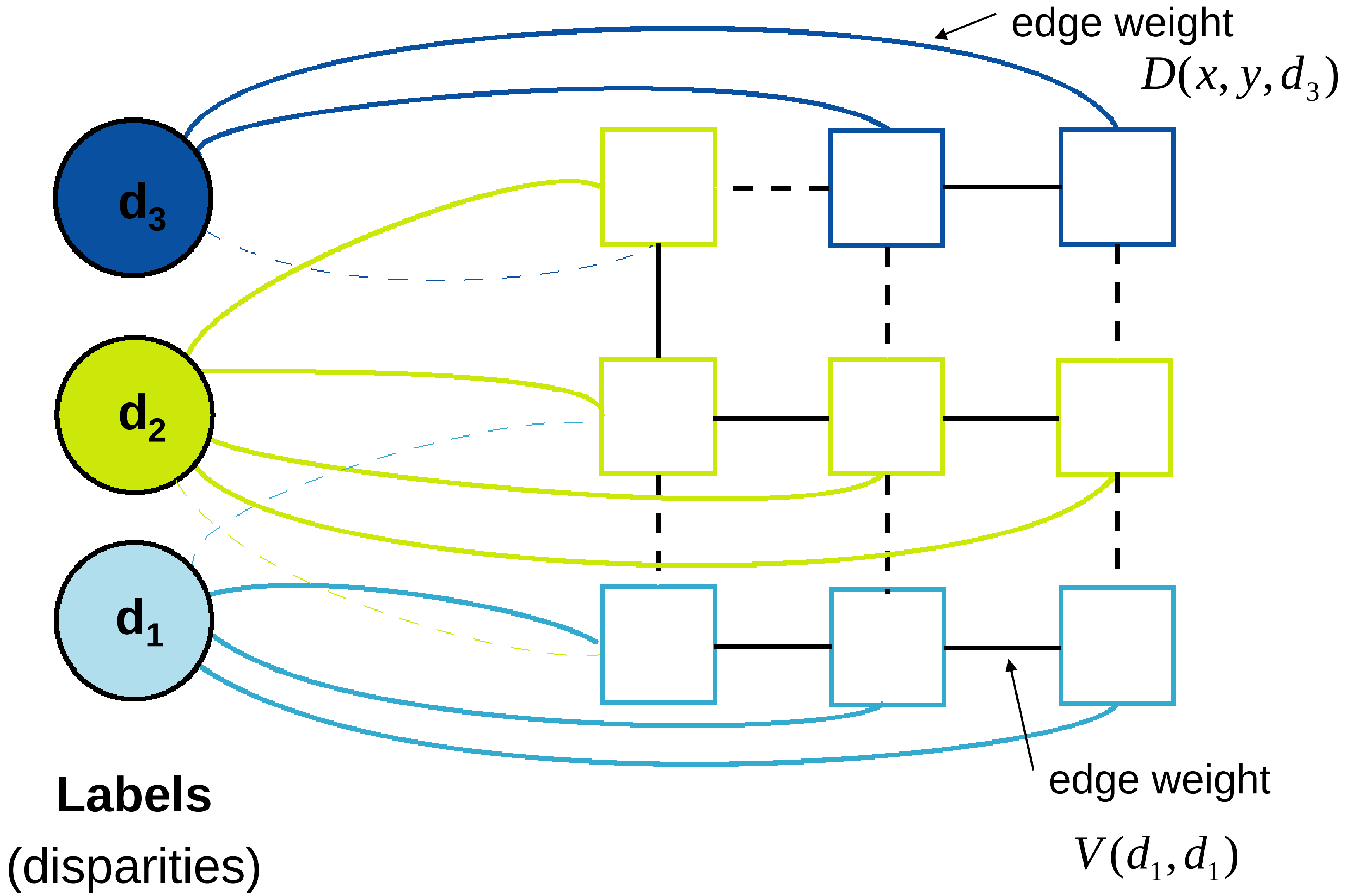


Min cut <=> labeling

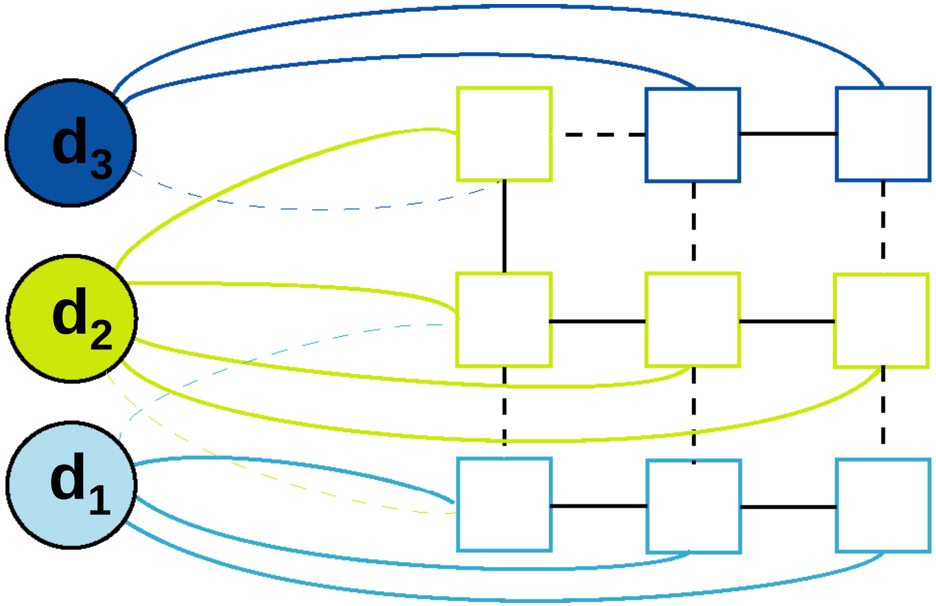
- In order to be a cut:
  - For each pixel, either the F or G edge has to be cut
- In order to be minimal
  - Only one edge label per pixel can be cut (otherwise could be added)



Energy minimization via graph cuts

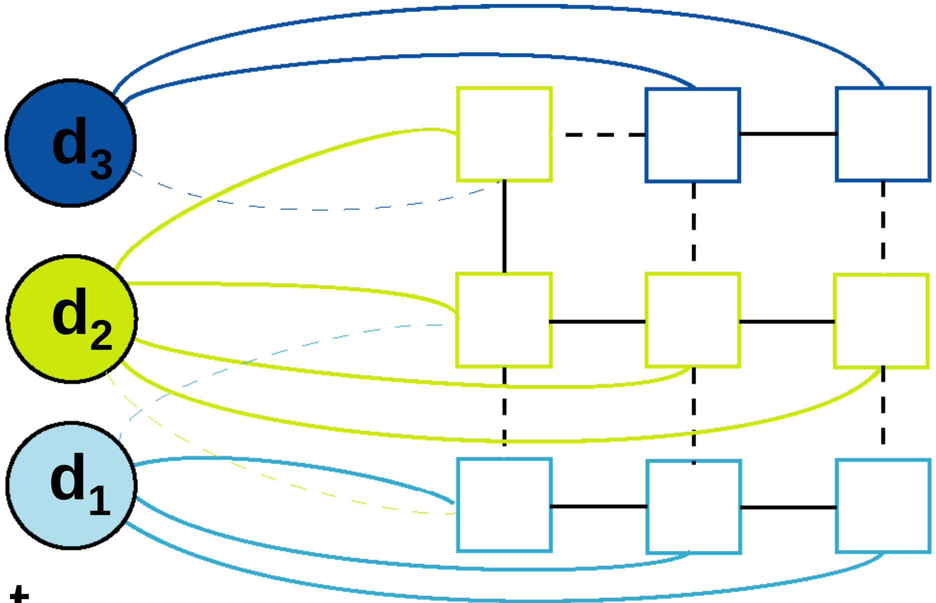


Energy minimization via graph cuts



- Graph Cost
  - Matching cost between images
  - Neighborhood matching term
  - Goal: figure out which labels are connected to which pixels

Energy minimization via graph cuts



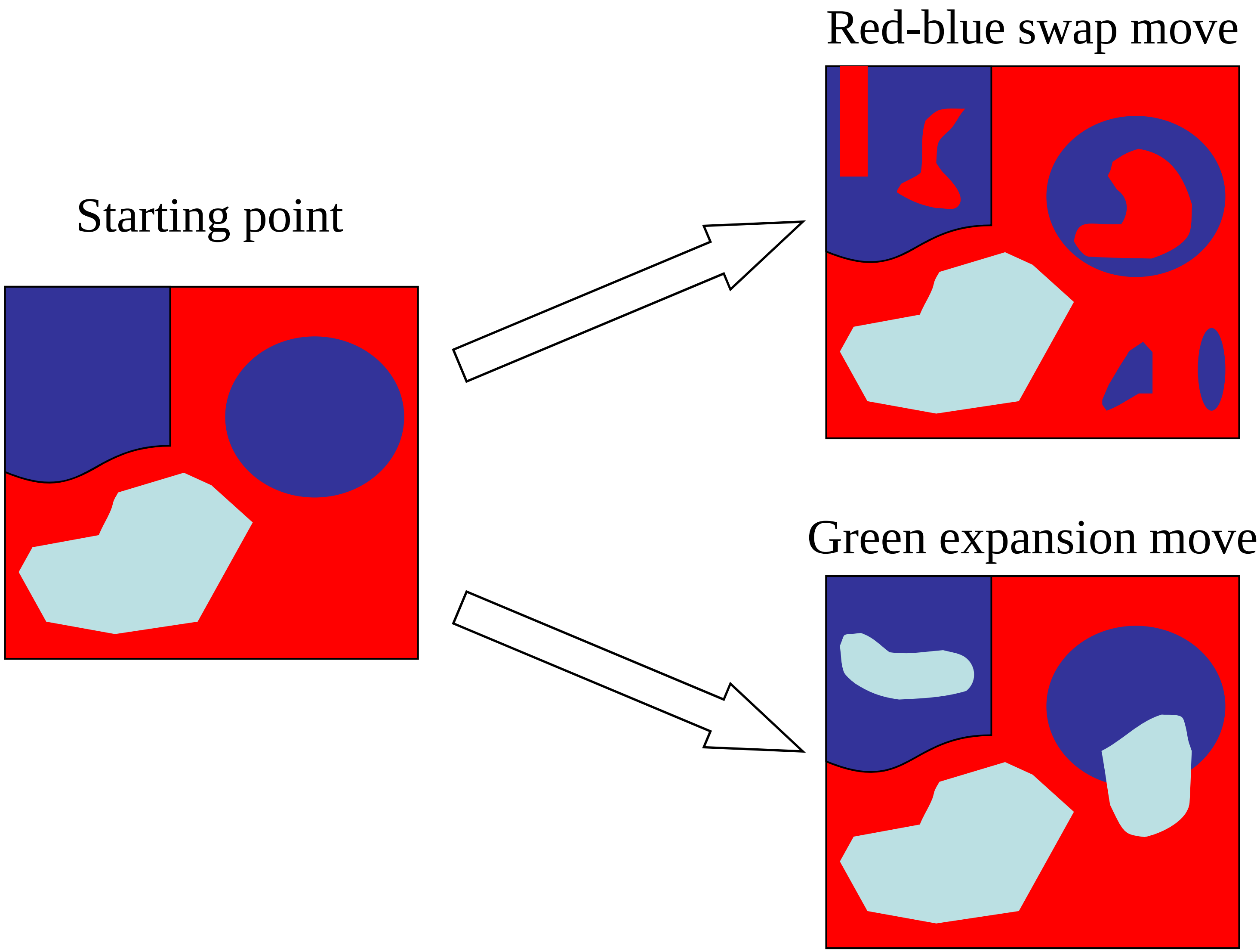
- Graph Cut
  - Delete enough edges so that
    - each pixel is (transitively) connected to exactly one label node
  - Cost of a cut: sum of deleted edge weights
  - Finding min cost cut equivalent to finding global minimum of energy function

Computing a multiway cut

- With 2 labels: classical min-cut problem
  - Solvable by standard flow algorithms
    - polynomial time in theory, nearly linear in practice
  - More than 2 terminals: NP-hard

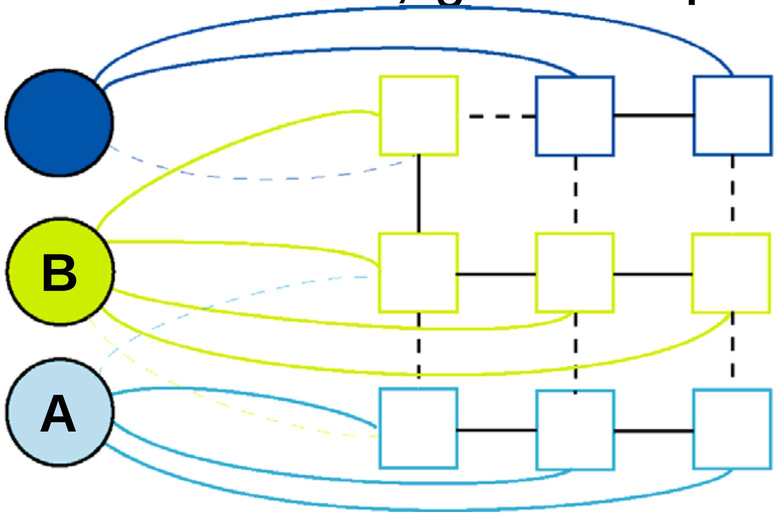
    [Dahlhaus *et al.*, STOC '92]
- Efficient approximation algorithms exist
  - Within a factor of 2 of optimal
  - Computes local minimum in a strong sense
    - even very large moves will not improve the energy
  - Yuri Boykov, Olga Veksler and Ramin Zabih, Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

Move examples

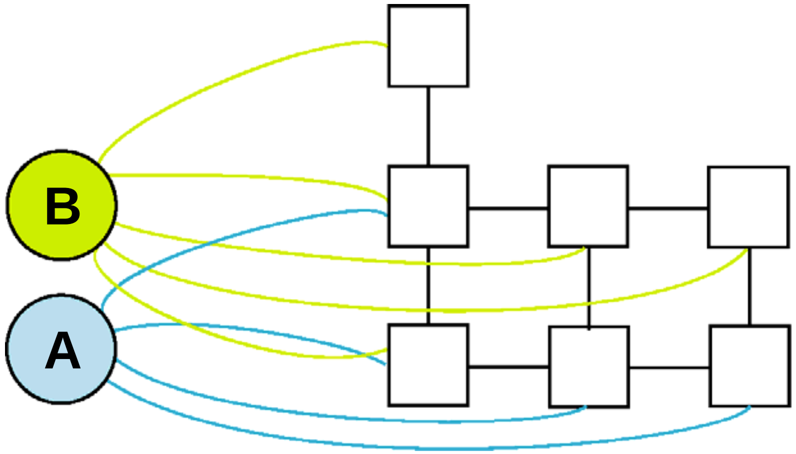


The swap move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label pair *(A,B)* in some order
   - 2.1 Find the lowest *E* labeling within a single *AB*-swap
   - 2.2 Go there if *E* is lower than the current labeling
3. If *E* did not decrease in the cycle, we're done
   Otherwise, go to step 2



Original graph



AB subgraph
(run min-cut on this graph)

The expansion move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label *A* in some order
   2.1 Find the lowest *E* labeling within a single *A*-expansion
   2.2 Go there if it *E* is lower than the current labeling
3. If *E* did not decrease in the cycle, we're done
   Otherwise, go to step 2

GrabCut
Interactive Foreground Extraction using Iterated Graph Cuts

Carsten Rother
Vladimir Kolmogorov
Andrew Blake

Microsoft Research Cambridge-UK

Demo

- video

Interactive Digital Photomontage

DigiVFX

- Combining multiple photos
- Find seams using graph cuts
- Combine gradients and integrate

Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, Michael Cohen, “Interactive Digital Photomontage”, SIGGRAPH 2004

set of originals

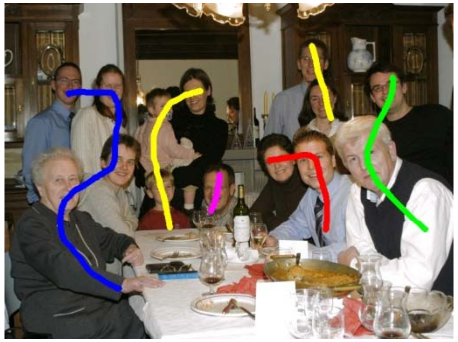
actual

photomontage

perceived

Source images

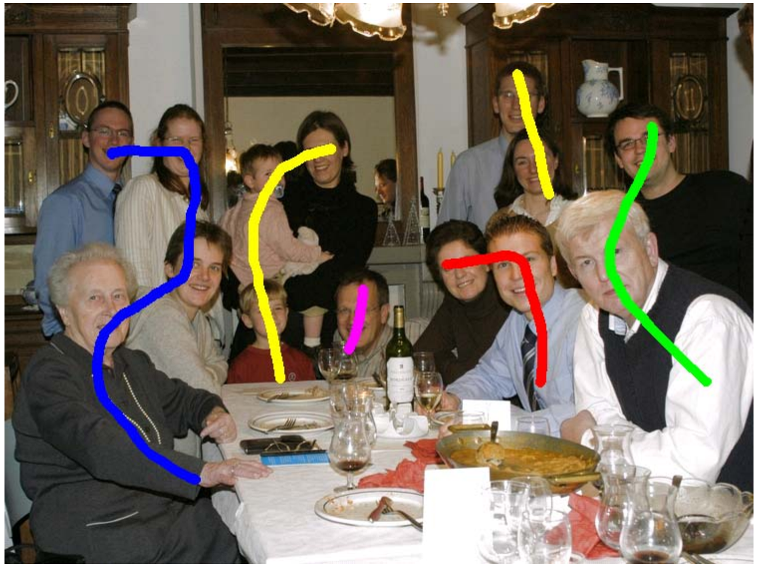
Brush strokes



Computed labeling

Composite

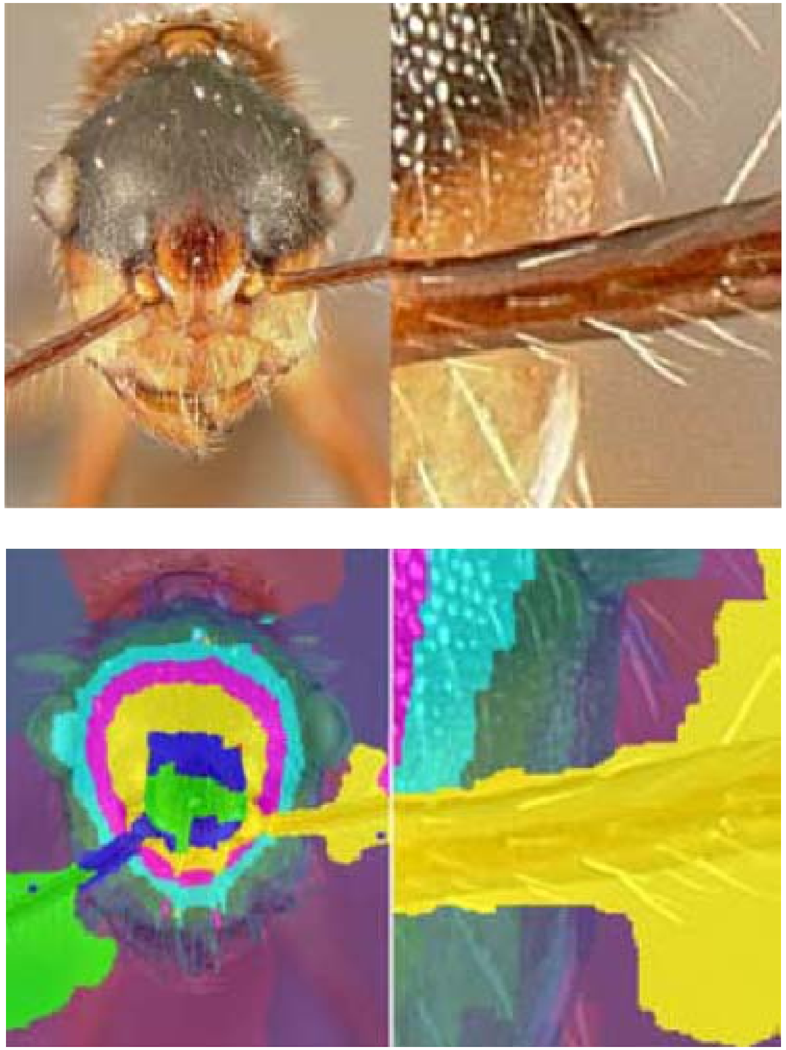
Brush strokes



Computed labeling

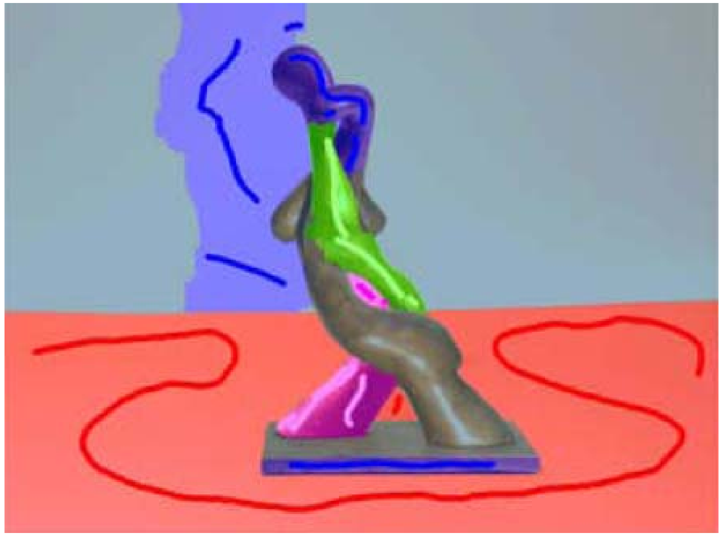
Interactive Digital Photomontage

- Extended depth of field



Interactive Digital Photomontage

- Relighting



Interactive Digital Photomontage

Demo

- video