Topic 16:

Markov Random Fields (MRFs)

- MRFs & energy minimization
- continuous MRFs
  - quadratic potentials (Gaussian MRFs)
  - robust/non-convex potentials (robust regularization)
- discrete MRFs
  - binary MRFs: Ising model & graph cuts
  - multi-valued MRFs: Potts model & expansion moves
- applications: grabcut, texture synthesis, PEARL

intro to energy minimization

Many vision tasks are naturally posed as energy minimization problems on a rectangular grid of pixels, where the energy comprises a *data term* and a *smoothness* term:

image

$$E(u) = \overbrace{E_{data}(u)}^{\text{consistency}} + \overbrace{E_{smoothness}(u)}^{\text{prior}} .$$

The data term $E_{data}(u)$ expresses our goal that the optimal model u be consistent with the measurements. The smoothness energy $E_{smoothness}(u)$ is derived from our prior knowledge about plausible solutions.

image denoising

(a) (b) (c) (d) (e)

Figure 19.4
Denoising with a Field of Experts. Full image (top) and detail (bottom). (a) Original noiseless image. (b) Image with additive Gaussian noise ($\sigma = 25$); PSNR = 20.29dB. (c) Denoised image using a Field of Experts; PSNR = 28.72dB. (d) Denoised image using the approach of Portilla et al. [375]; PSNR = 28.90dB. (e) Denoised image using nonlocal means [79]; PSNR = 28.21dB.

$$E(u) = E_{data}(u) + E_{smoothness}(u)$$

noise model consistency

neighborhood consistency

l_1	l_2	l_3
l_4	l_5	l_6
l_7	l_8	l_9

dealing with missing data: image inpainting

(a) (b) (c)

Figure 19.6
Inpainting with a Field of Experts. (a) Original image overlaid with text to be removed. (b) Inpainting result from (c) Close-up comparison between a (left), b (middle), and the results of Bertalmío et al. [33] (right).

texture synthesis

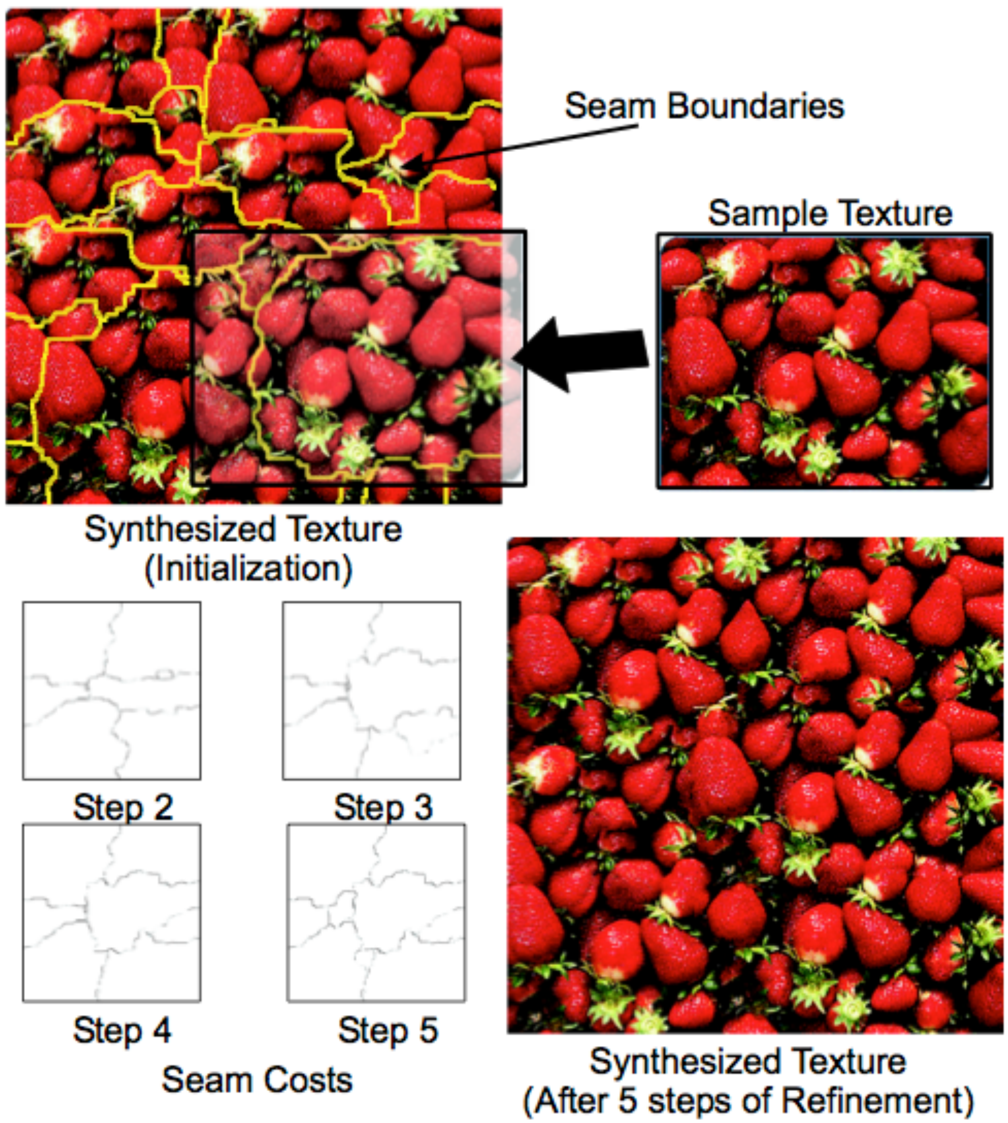
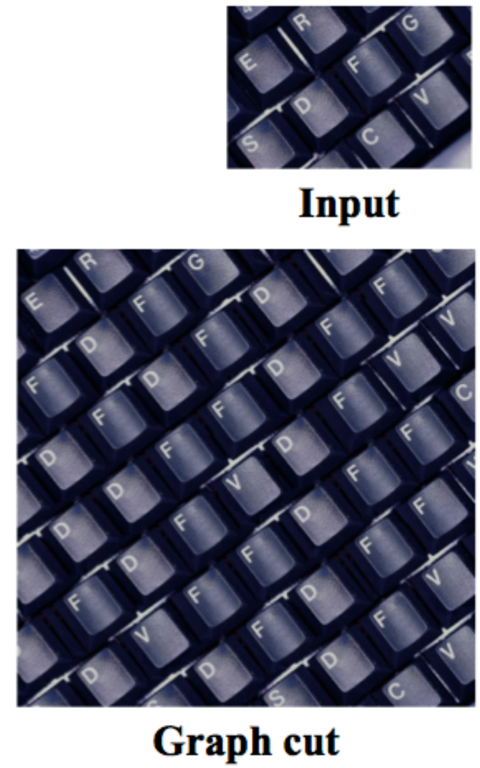


Figure 5: This figure illustrates the process of synthesizing a larger texture from an example input texture. Once the texture is initialized, we find new patch locations appropriately so as to refine the texture. Note the irregular patches and seams. Seam error measures that are used to guide the patch selection process are shown. This process is also shown in the video.



texture synthesis

igure 8: 2D texture synthesis results. We show results for textured and natural images. The smaller images are the example images used for ynthesis. Shown are CHICK PEAS, TEXT, NUTS, ESCHER, MACHU PICCHU©Adam Brostow, CROWDS and SHEEP from left to right and top o bottom.

automatic image segmentation

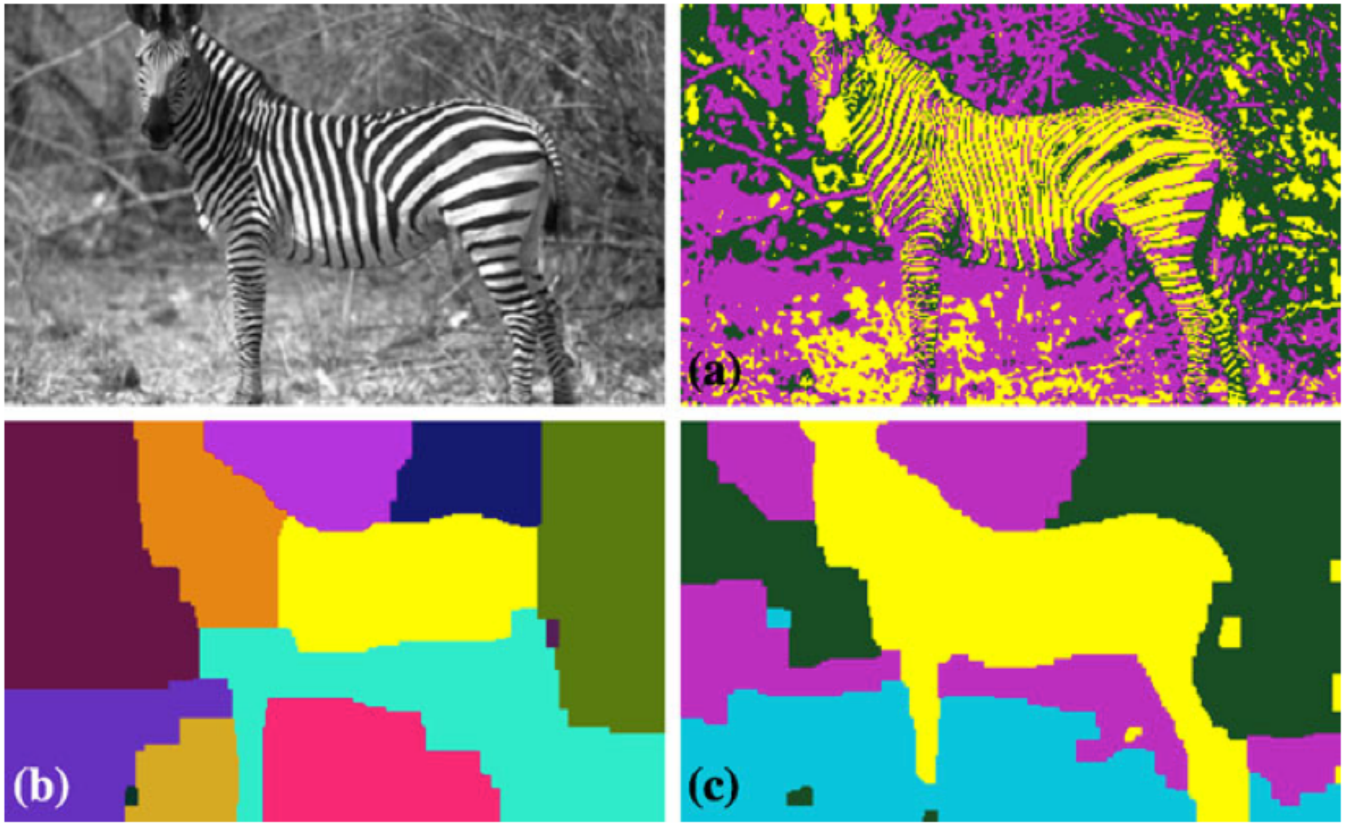


Fig. 3 Unsupervised segmentation using histogram models. Energy (1) clusters in color space, so segments (**a**) are incoherent. Energy (2) clusters over pixels and must either over-segment or over-smooth (**b**), just as in Zabih and Kolmogorov (2004). Our energy ($\star$) balances these criteria (**c**) and corresponds to Zhu and Yuille (1996) for segmentation

interactive image segmentation

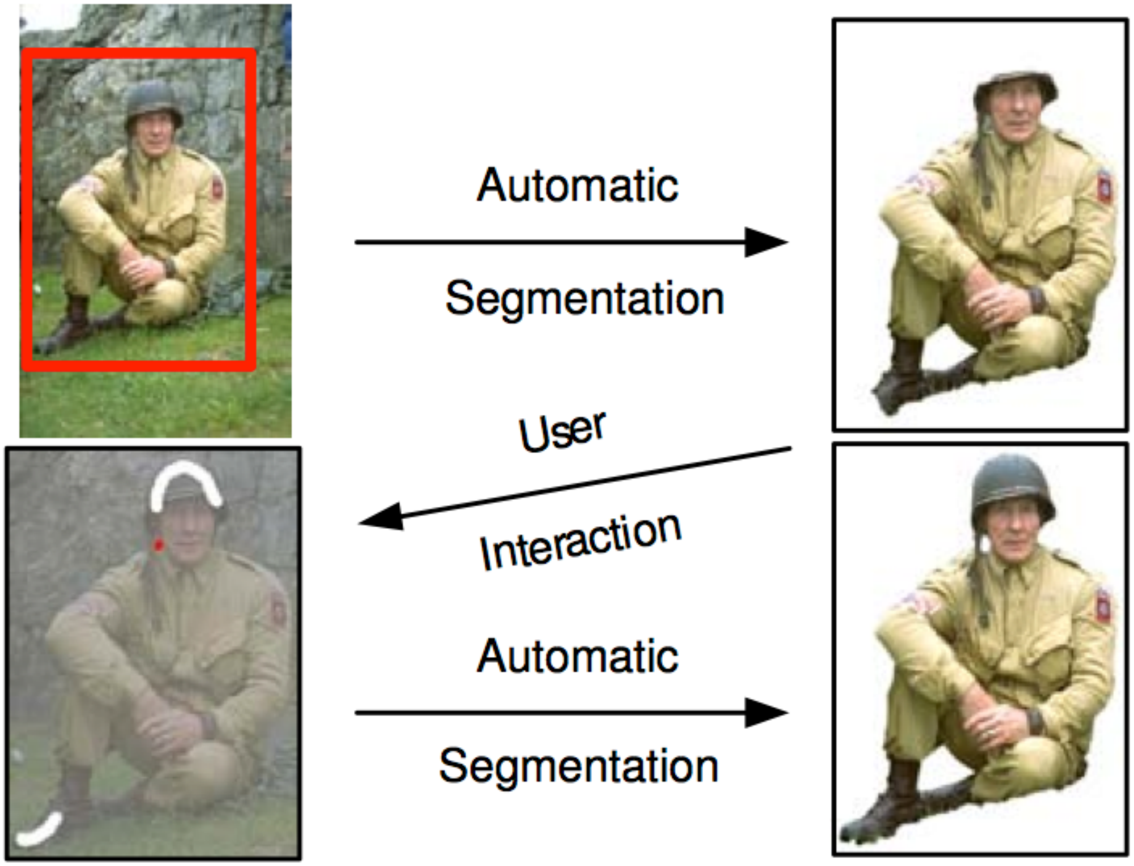


Figure 5: **User editing.** After the initial user interaction and segmentation (top row), further user edits (fig. 3) are necessary. Marking roughly with a foreground brush (white) and a background brush (red) is sufficient to obtain the desired result (bottom row).

stereo disparity estimation

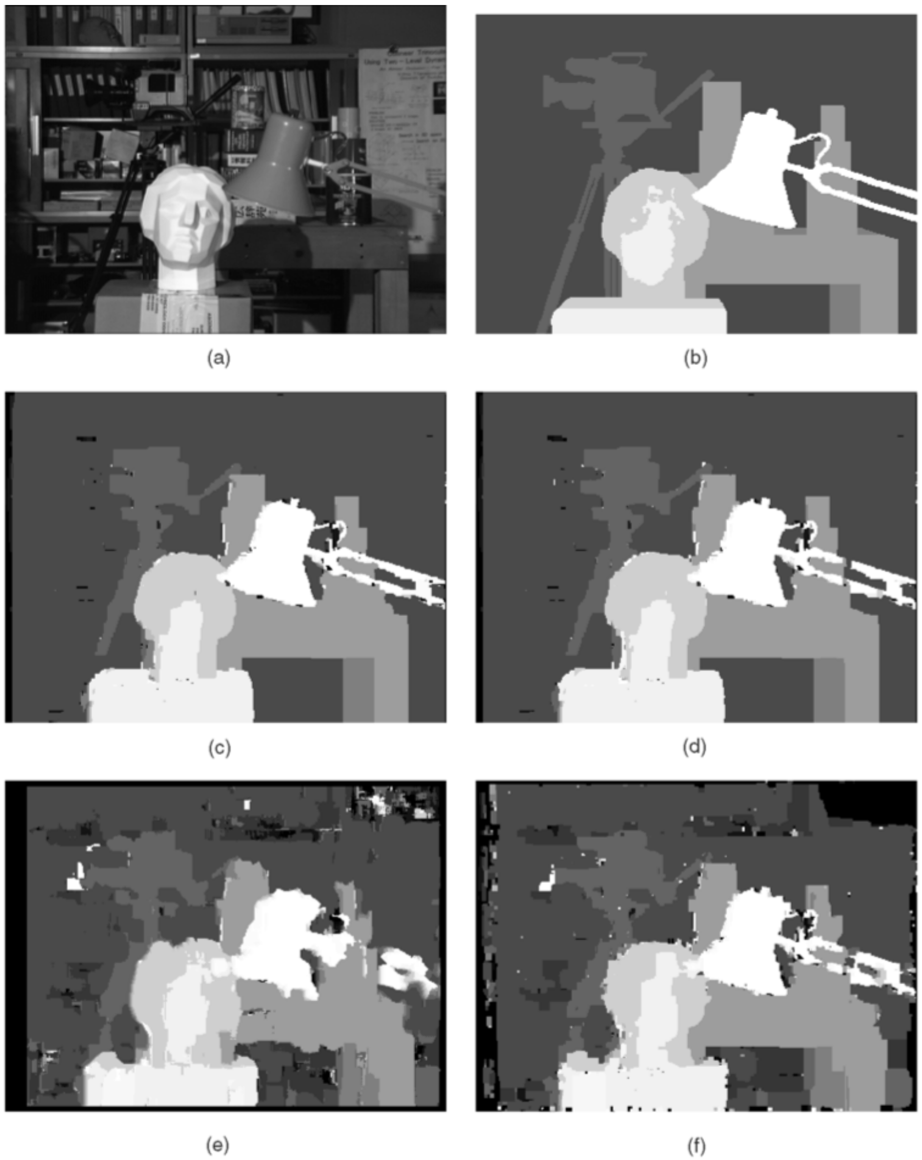


Fig. 10. Real imagery with ground truth. (a) Left image: 384x288, 15 labels. (b) Ground truth. (c) Swap algorithm. (d) Expansion algorithm. (e) Normalized correlation. (f) Simulated annealing.

stereo disparity estimation

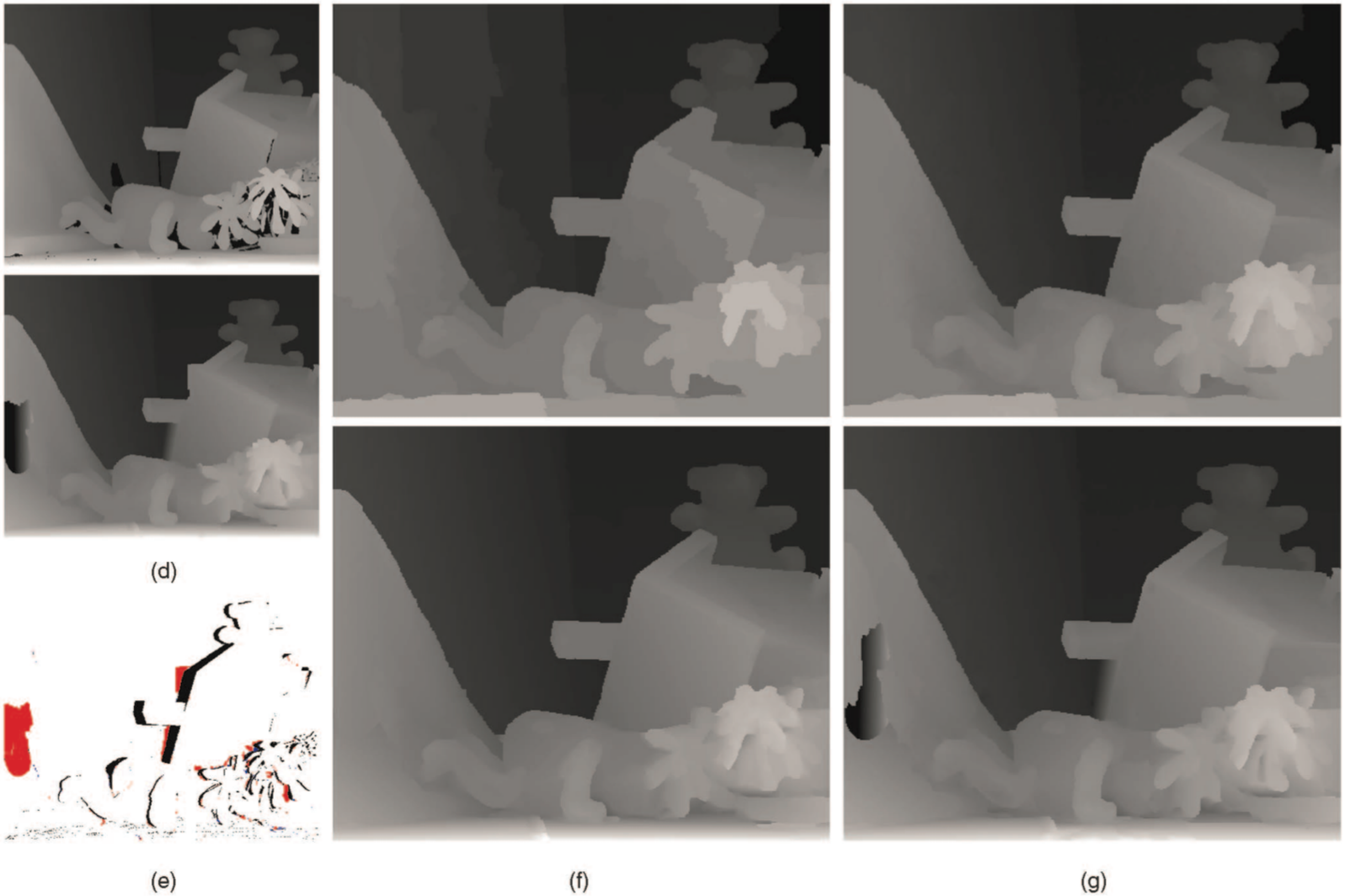
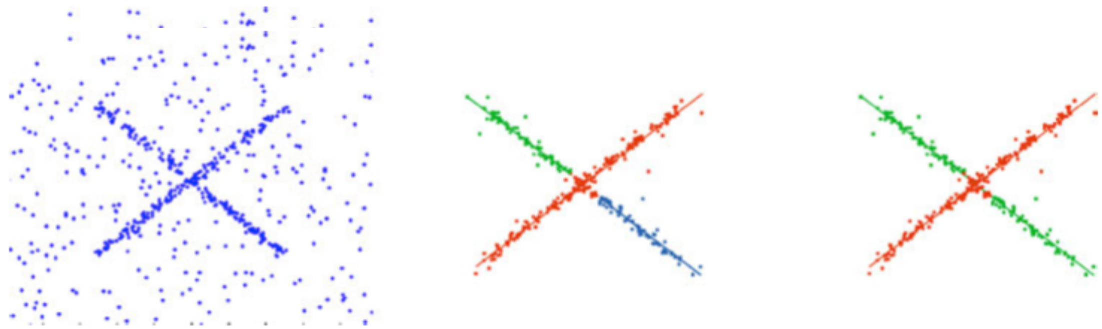


Fig. 7. Results for the Middlebury Teddy sequence. (a) Ground truth. (b) 1op linear. (c) 1op quadratic. (d) 2op linear no-vis. (e) Occlusions. (f) 2op linear. (g) 2op quadratic.

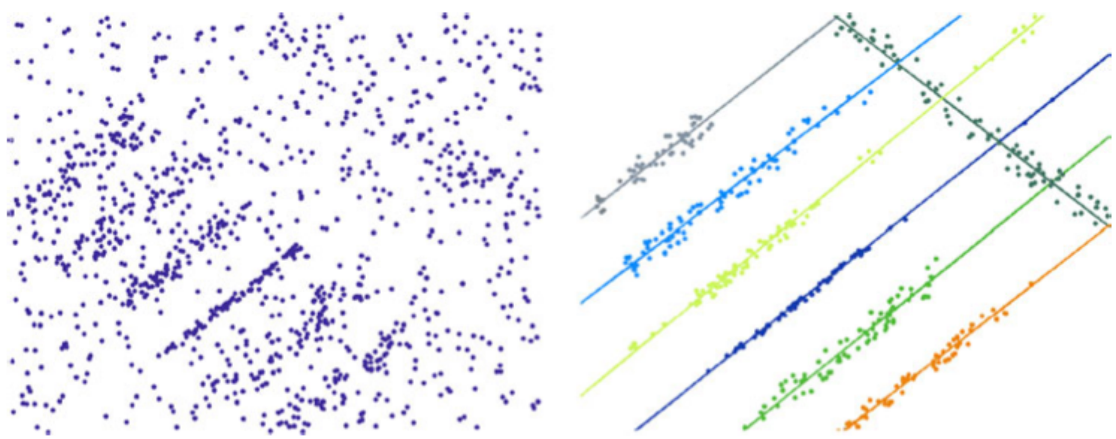
model fitting



(a) data (300 outliers) (b) minimum of energy (3) (c) *merging*, energy (5)

Fig. 10 (Color online) Intersecting lines example. Optimization of energy (3) may leave spatially isolated groups of inliers assigned to 2 models even if their parameters are infinitesimally close (**b**). Per-label costs in energy (5) solves this problem (**c**) but requires either an additional merging operation or an extension of α-expansion (Delong et al. 2011). This example may also suggest EM-style soft assignments of labels to points near the intersection. However, in vision (Sect. 3) we get occlusions (not intersections), which better motivate our "hard" assignments of labels

Fig. 11 (Color online) Fitting lines with different noise levels. The inliers in (**a**) were generated with different levels of Gaussian noise. 40% of the data are outliers. In (**b**) PEARL estimated labels combining geometric model parameters with unknown noise variances using error measure (6). Previous multi-model fitting methods use fixed thresholds to identify inliers, which would not work in this case



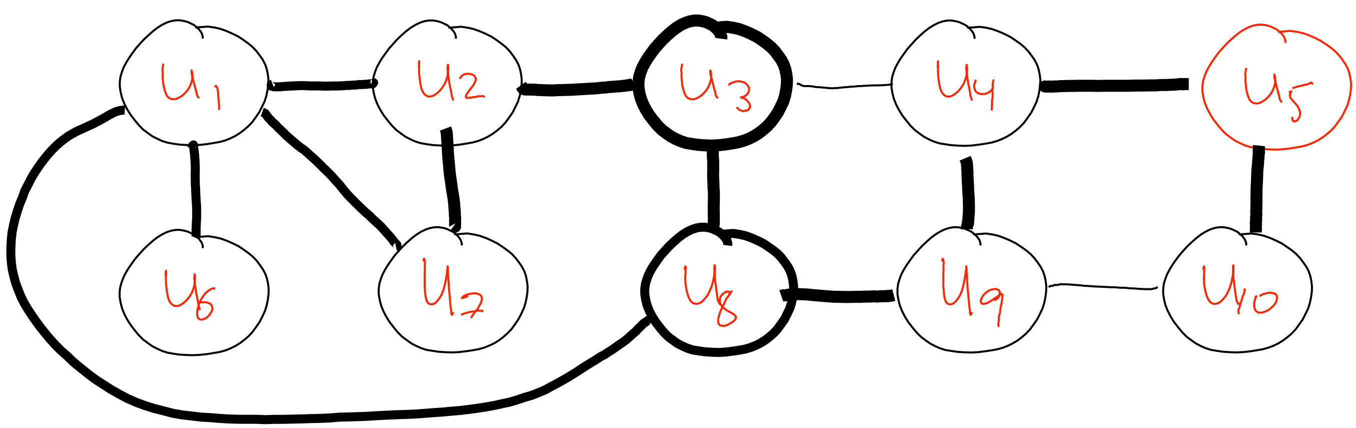
(a) models with different noise levels (b) PEARL result based on (6)

Markov Random Fields

A Markov Random Field (MRF) is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

- $\mathcal{V} = \{1, 2, ..., N\}$ is the set of *nodes*, each of which is associated with a random variable (RV), u_j, for $j = 1...N$.
- The neighbourhood of node i, denoted $\mathcal{N}_i$, is the set of nodes to which i is adjacent; i.e., $j \in \mathcal{N}_i$ if and only if $(i, j) \in \mathcal{E}$.
- The Markov Random field satisfies

$$p(u_i \,|\, \{u_j\}_{j \in \mathcal{V} \backslash i}) = p(u_i \,|\, \{u_j\}_{j \in \mathcal{N}_i}) \,. \tag{1}$$



$\mathcal{N}_3 = \{2, 4, 8\}$

$\mathcal{N}_1 = \{2, 6, 7, 8\}$

Markov Random Fields: conditional independence

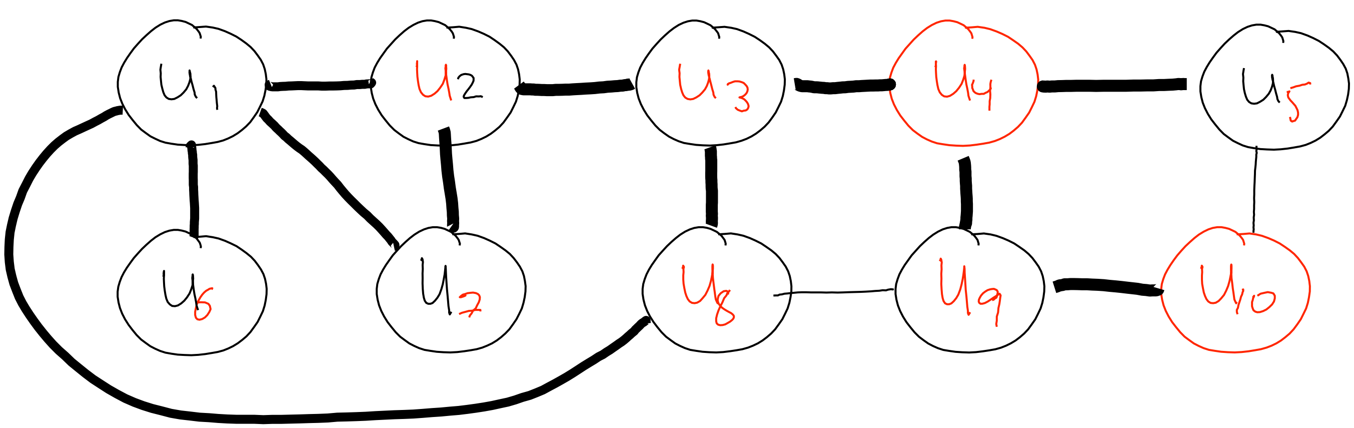
In the graph below, we have

$$P(u_3 \mid u_1, u_2, u_4, \ldots, u_{10}) = P(u_3 \mid u_2, u_4, u_8)$$

$$P(u_1 \mid u_2, \ldots, u_{10}) = P(u_1 \mid u_2, u_6, u_7, u_8)$$

- **The Markov Random field satisfies**

$$p(u_i \mid \{u_j\}_{j \in \mathcal{V} \setminus i}) = p(u_i \mid \{u_j\}_{j \in \mathcal{N}_i}) \,. \qquad (1)$$



$$\mathcal{N}_3 = \{2, 4, 8\}$$

$$\mathcal{N}_1 = \{2, 6, 7, 8\}$$

Markov Random Fields: conditional independence

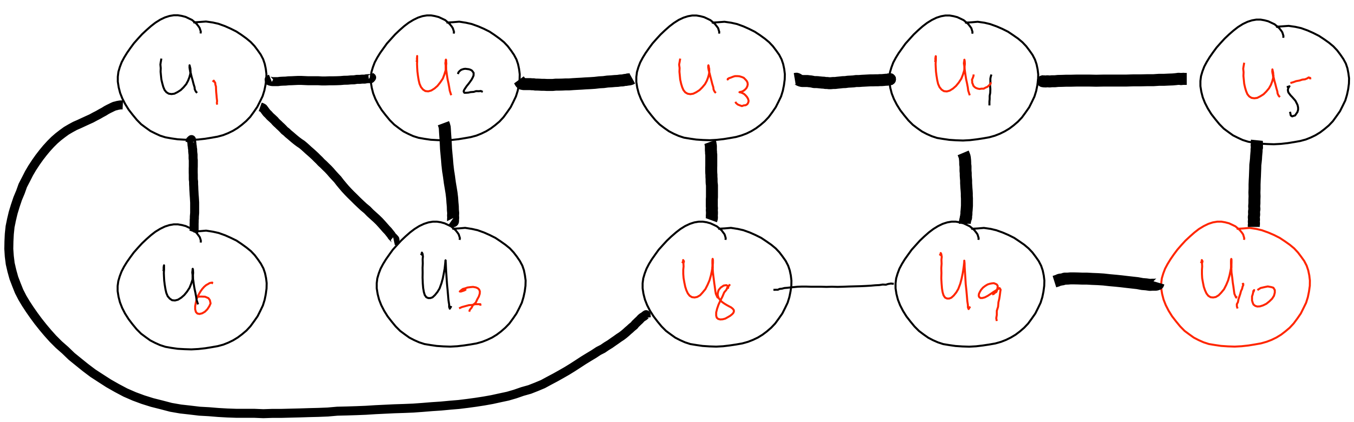
For a given set of labels Λ

and a configuration $\omega = (l_1, l_2, \ldots, l_{10})$ $l_i \in \Lambda$

$$p(u_3 \mid u_1, u_2, u_4 \ldots, u_{10}) = \frac{P(u_3 = l_3, u_1 = l_1, u_{10} = l_{10})}{\sum_{l \in \Lambda} P(u_3 = l, u_1 = l_1, \ldots u_{10} = l_{10})}$$

- The Markov Random field satisfies

$$p(u_i \mid \{u_j\}_{j \in \mathcal{V} \setminus i}) = p(u_i \mid \{u_j\}_{j \in \mathcal{N}_i}) \, . \quad (1)$$



$\mathcal{N}_3 = \{2, 4, 8\}$

$\mathcal{N}_1 = \{2, 6, 7, 8\}$

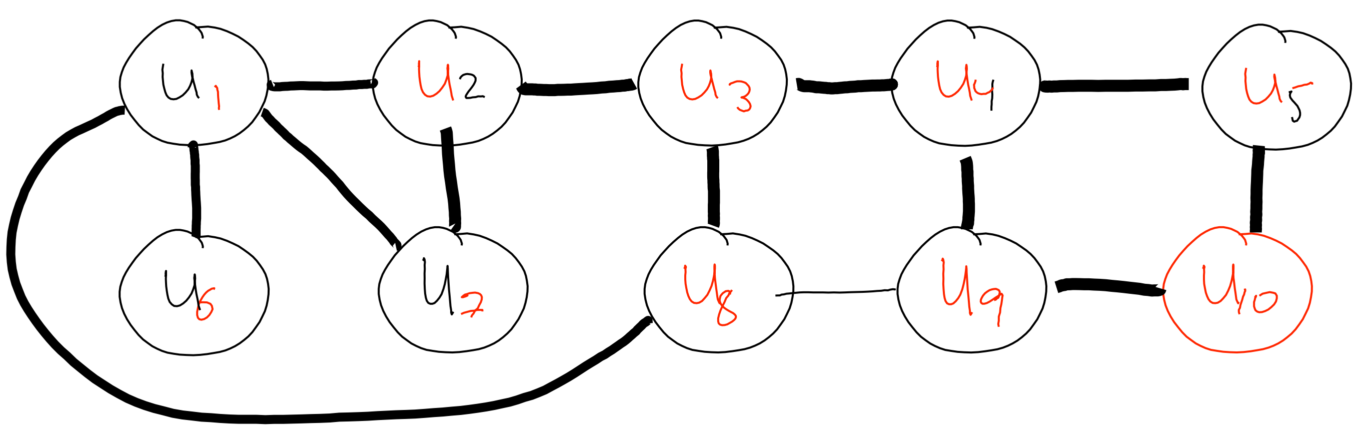
Markov Random Fields: conditional independence

a function $p(u)$ that models the joint distribution

$$P(u_3 \mid u_1, u_2, u_4 .., u_{10}) = \frac{\overbrace{P(u_3 = l_3, u_1 = l_1, u_{10} = l_{10})}}{\sum_{l \in \Lambda} P(u_3 = l, u_1 = l_1, .. u_{10} = l_{10})}$$

- **The Markov Random field satisfies**

$$p(u_i \mid \{u_j\}_{j \in \mathcal{V} \setminus i}) = p(u_i \mid \{u_j\}_{j \in \mathcal{N}_i}) . \qquad (1)$$



$\mathcal{N}_3 = \{2, 4, 8\}$

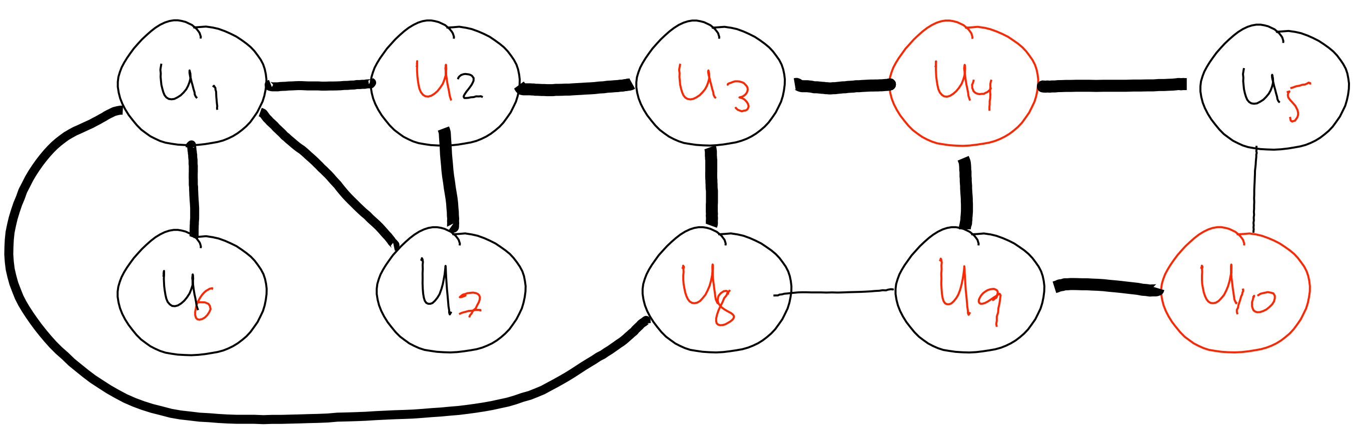
$\mathcal{N}_1 = \{2, 6, 7, 8\}$

Markov Random Fields: key questions

- What functions $p(w)$ yield conditional PDFs that satisfy the conditional independence constraints?
- What is the general form of $p(w)$?

- The Markov Random field satisfies

$$p(u_i \,|\, \{u_j\}_{j \in \mathcal{V} \backslash i}) = p(u_i \,|\, \{u_j\}_{j \in \mathcal{N}_i}) \,. \qquad (1)$$



$N_3 = \{2, 4, 8\}$

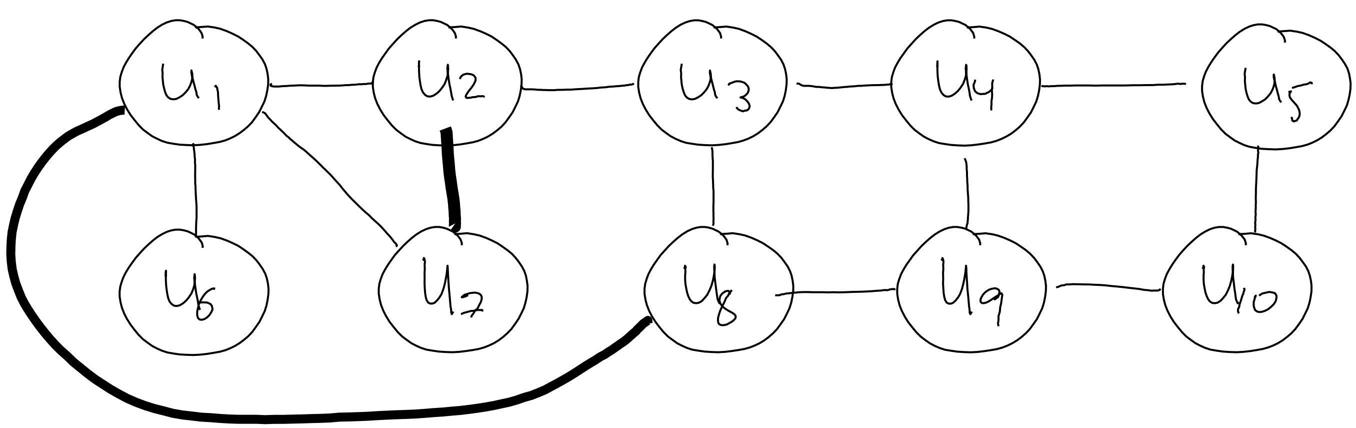
$N_1 = \{2, 6, 7, 8\}$

joint distribution of an MRF: Gibbs distribution

Theorem: U is an MRF over the graph G if and only if $p(\omega) = \text{Prob}(U=\omega)$ is a Gibbs distribution wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of G



clique = fully connected subgraph of G

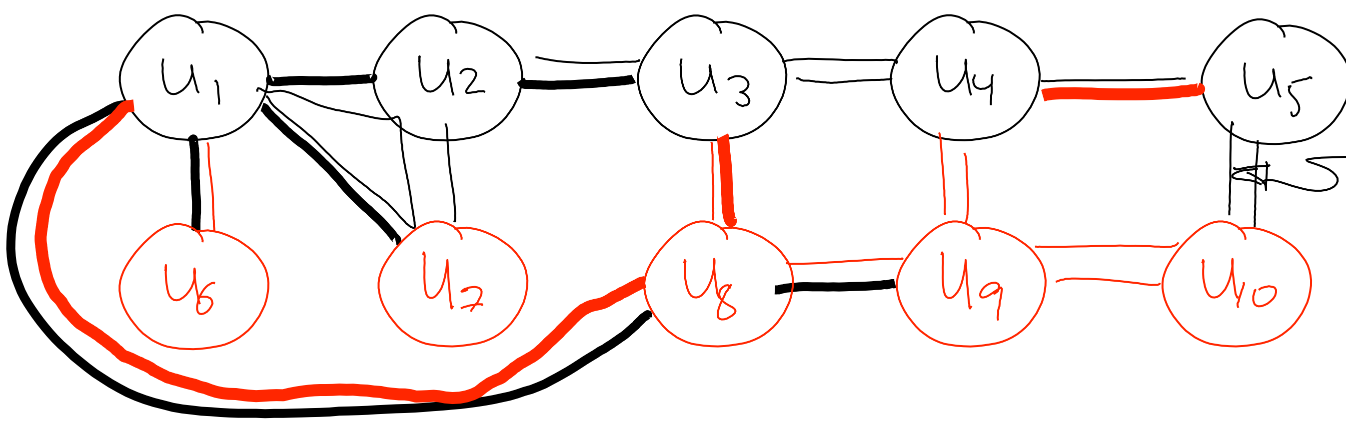
joint distribution of an MRF: Gibbs distribution

Theorem. U is an MRF over the graph G if and only if $p(\omega) = \text{Prob}(U=\omega)$ is a Gibbs distribution wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of G

$C = \{\{1,6\}, \{1,2,7\}, \{1,8\}, \{2,3\}, \{3,8\}, \{3,4\}, \ldots\}$



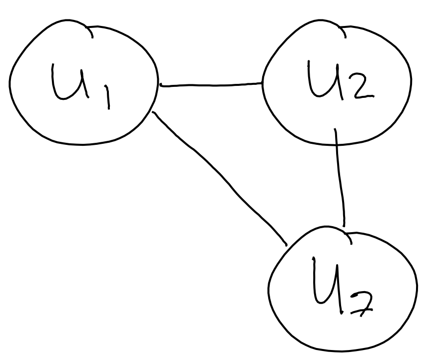
clique = fully connected subgraph of G

joint distribution of an MRF: Gibbs distribution

Theorem. U is an MRF over the graph G if and only if $p(\omega) = \text{Prob}(U = \omega)$ is a Gibbs distribution wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of G
2. Clique potential $0 < \phi_c(\overline{U}_c, \theta_c) < 1$



maximal clique

random variables in clique

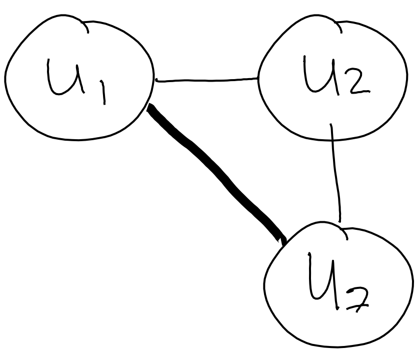
parameters

joint distribution of an MRF: Gibbs distribution

Theorem. $\mathcal{U}$ is an MRF over the graph G if and only if $p(\omega) = \text{Prob}(\mathcal{U}=\omega)$ is a Gibbs distribution wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of G
2. Clique potential $0 < \Phi_c(\bar{u}_c, \theta_c) < 1$



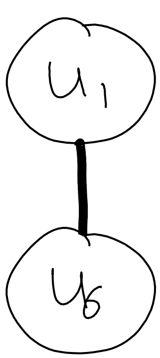
$$\Phi_{12}\left((u_1, u_2, u_7); \theta_{127}\right) > 0$$

joint distribution of an MRF: Gibbs distribution

Theorem: U is an MRF over the graph G if and only if $p(\omega) = \mathrm{Prob}(U=\omega)$ is a Gibbs distribution wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of G
2. Clique potential $0 < \Phi_c(\overline{u}_c, \vartheta_c) \le 1$



$$\phi_{16}\left((u_1, u_6) ; \vartheta_{16}\right) > 0$$

joint distribution of an MRF: Gibbs distribution

Theorem. U is an MRF over the graph G if and only if $p(\omega) = \text{Prob}(U = \omega)$ is a Gibbs distribution wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of G
2. Clique potential $0 < \Phi_c(\bar{u}_c, \theta_c) < 1$
3. $$\underbrace{p(u_1, u_2, \ldots, u_N)}_{\text{joint probability}} = \underbrace{\frac{1}{Z}}_{\text{normalizing constant}} \underbrace{\prod_{c \in C} \Phi_c(\bar{u}_c, \theta_c)}_{\text{product over all clique potentials}}$$

joint distribution of an MRF: Gibbs distribution

Theorem. U is an MRF over the graph G if and only if $p(\omega) = \mathrm{Prob}(U=\omega)$ is a Gibbs distribution wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of G
2. Clique potential $0 < \Phi_c(\overline{u}_c, \theta_c) < 1$
3. $$p(u_1, u_2, \dots, u_N) = \frac{1}{Z} \underbrace{\prod_{c \in C} e^{-\phi_c(\overline{u}_c; \theta_c)}}_{\text{product over all clique potentials}}$$

joint probability

normalizing constant

joint distribution of an MRF: Gibbs distribution

<u>Theorem</u>: U is an MRF over the graph G if and only if $p(\omega) = \text{Prob}(U = \omega)$ is a <u>Gibbs distribution</u> wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of <u>maximal cliques</u> of G
2. Clique potential $0 \le \psi_c(\bar{u}_c, \theta_c)$
3. $$p(u_1, u_2, \dots, u_N) = \frac{1}{Z} e^{-\sum_{c \in C} \psi_c(\bar{u}_c, \theta_c)}$$

joint probability

normalizing constant

product over all clique potentials

joint distribution of an MRF: Gibbs distribution

Theorem: U is an MRF over the graph G if and only if $p(\omega) = \text{Prob}(U=\omega)$ is a Gibbs distribution wrt G

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of G
2. Clique potential $0 \leq \psi_c(\bar{u}_c, \theta_c)$
3. $p(u_1, u_2, \ldots, u_N) = \frac{1}{Z} e^{-E(u_1, \ldots, u_N; \theta)}$

Energy function $E = \sum_{c \in C} \psi_c(\bar{u}_c; \theta_c)$

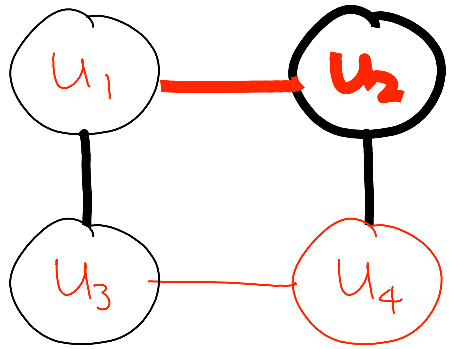
joint distribution of an MRF: Gibbs distribution

Theorem: $\mathcal{U}$ is an MRF over the graph $\mathcal{G}$ if and only if $p(\omega) = \text{Prob}(\mathcal{U} = \omega)$ is a Gibbs distribution wrt $\mathcal{G}$

Gibbs distribution defined in 4 steps:

1. C is the set of maximal cliques of $\mathcal{G}$
2. Clique potential $0 \leq \psi_c(\bar{u}_c, \vartheta_c)$
3. Energy function $E = \sum_{c \in C} \psi_c(\bar{u}_c; \vartheta_c)$
4. Partition function $Z = \sum_{u_1, \ldots, u_N} e^{-E(u_1, \ldots, u_N; \vartheta)}$

MRF example



Cliques: $\{1,2\}, \{1,3\}, \{2,4\}, \{3,4\}$

$$P(u_1,u_2,u_3,u_4) = \frac{1}{Z}\,\phi_{12}(u_1,u_2)\cdot\phi_{13}(u_1,u_3)\cdot\phi_{24}(u_2,u_4)\,\phi_{34}(u_3,u_4)$$

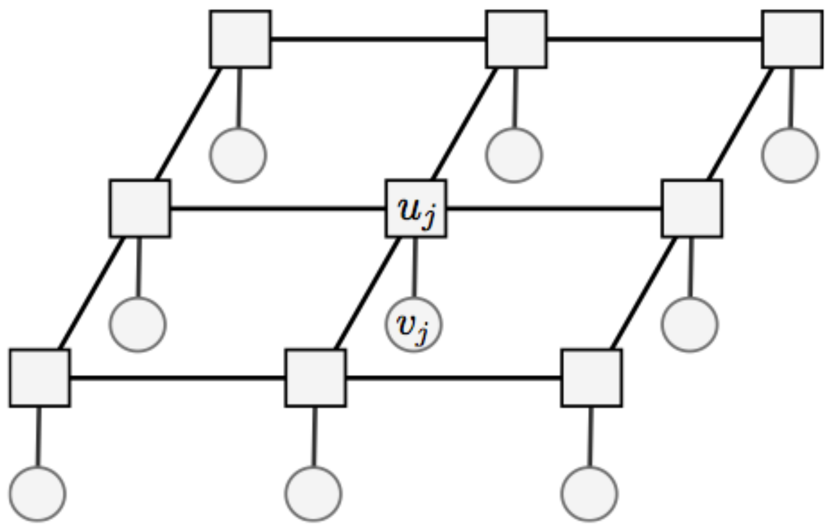
$$E(u_1,u_2,u_3,u_4) = \psi_{12}(u_1,u_2)+\psi_{13}(u_1,u_3) + \psi_{24}(u_2,u_4)+\psi_{34}(u_3,u_4)$$

$$P(u_1,u_2,u_3,u_4) = \frac{1}{Z}\exp\left(-E(u_1,u_2,u_3,u_4)\right)$$

example: image denoising

Consider image restoration: Given a noisy image v, perhaps with missing pixels, recover an image u that is both smooth and close to v.

Let each pixel be a node in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with 4-connected neighbourhoods. The maximal cliques are pairs of nodes.



smoothness prior

Accordingly, the energy function is given by

consistency of u_j with measurement v_j

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j) \quad (3)$$

example: image denoising

smoothness prior

Accordingly, the energy function is given by

consistency of u_j with measurement v_j

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j) \qquad (3)$$

- Unary (clique) potentials D stem from the measurement model, penalizing the discrepancy between the data v and the solution u. This models assumes conditional independence of observations. The unary potentials are pixel log likelihoods.
- Interaction (clique) potentials V provide a definition of smoothness, penalizing changes in u between pixels and their neighbours.

Goal: Find the image u that minimizes $E(u)$ (and thereby maximizes $p(u|v)$ since, up to a constant, E is equal to the negative log posterior).

Topic 16:

Markov Random Fields (MRFs)

- MRFs & energy minimization
- continuous MRFs
  - quadratic potentials (Gaussian MRFs)
  - robust/non-convex potentials (robust regularization)
- discrete MRFs
  - binary MRFs: Ising model & graph cuts
  - multi-valued MRFs: Potts model & expansion moves
- applications: grabcut, texture synthesis, PEARL

quadratic potentials in 1D

consider measurement vector

$$v = u + e$$

smooth 1D signal $(u_1, u_2, \ldots, u_N)$

iid Gaussian noise $(e_1, e_2, \ldots, e_N)$

Energy

$$E(u) = \underbrace{\sum_{n=1}^{N} (u_n - v_n)^2}_{\text{negative log-likelihood for noise model}} + \lambda \underbrace{\sum_{n=1}^{N-1} (u_{n+1} - u_n)^2}_{\text{quadratic smoothness prior}}$$

$\lambda > 0$ trades smoothness and data fidelity

quadratic potentials in 1D

$$E(u) \;=\; \sum_{n=1}^{N}(u_n - v_n)^2 + \lambda \sum_{n=1}^{N-1}(u_{n+1} - u_n)^2\,. \qquad (5)$$

A good solution u^* should be close to v, and adjacent nodes on the grid should have similar values. The constant, $\lambda > 0$, controls the tradeoff between smoothness and data fit.

To find the optimal u^*, we take derivatives of $E(u)$ with respect to u_n:

$$\frac{\partial\, E(u)}{\partial\, u_n} \;=\; 2\,(u_n \;-\; v_n) \;+\; 2\lambda\,(-u_{n-1} + 2u_n - u_{n+1})\;,$$

and therefore the necessary condition for the critical point is

$$u_n \;+\; \lambda\,(-u_{n-1} + 2u_n - u_{n+1}) \;=\; v_n\;. \qquad (6)$$

Equation (6) does not hold at endpoints, $n = 1$ and $n = N$, as they have only one neighbor. For endpoints we obtain different equations:

$$u_1 + \lambda\,(u_1 - u_2) \;=\; v_1$$
$$u_N + \lambda\,(u_N - u_{N-1}) \;=\; v_N$$

quadratic potentials in 1D: optimization

We therefore have N linear equations in the N unknowns:

$$\begin{pmatrix} 1+\lambda & -\lambda & 0 & 0 & \dots & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & \dots & 0 \\ 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & \dots & & 0 & -\lambda & 1+\lambda \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_N \end{pmatrix} \qquad (7)$$

quadratic potentials in 1D: optimization

We therefore have N linear equations in the N unknowns:

$$\begin{pmatrix} 1+\lambda & -\lambda & 0 & 0 & \dots & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & \dots & 0 \\ 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & \dots & & 0 & -\lambda & 1+\lambda \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_N \end{pmatrix} \qquad (7)$$

$$u_n^{(t+1)} = \begin{cases} \frac{1}{1+2\lambda}\left(v_n + \lambda u_{n-1}^{(t)} + \lambda u_{n+1}^{(t)}\right) & \text{for } 1<n<N \\ \frac{1}{1+\lambda}\left(v_1 + \lambda u_2^{(t)}\right) & \text{for } n=1 \\ \frac{1}{1+\lambda}\left(v_N + \lambda u_{N-1}^{(t)}\right) & \text{for } n=N \end{cases} \qquad (8)$$

MRF optimization as a smoothing filter

We therefore have N linear equations in the N unknowns:

$$\begin{pmatrix} 1+\lambda & -\lambda & 0 & 0 & \dots & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & \dots & 0 \\ 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & \dots & & 0 & -\lambda & 1+\lambda \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_N \end{pmatrix} \qquad (7)$$

If we neglect boundary conditions, e.g., assuming the signal is much longer than the filter support, then (7) can be approximated by convolution:

$$u * (\delta + \lambda g) = v\,. \qquad (9)$$

where $g = [-1, 2, -1]$. Note that in (9) we convolve the noiseless signal u with the filter to obtain the noisy input, v.

MRF optimization as a smoothing filter

$$
\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} 1+\lambda & -\lambda & 0 & 0 & \dots & 0 \\ -\lambda & 1+2\lambda & -\lambda & 0 & \dots & 0 \\ 0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & \dots & & 0 & -\lambda & 1+\lambda \end{pmatrix}^{-1} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_N \end{pmatrix}
$$

We can also define the corresponding inverse filter h; i.e., $u = h * v$.

MRF optimization as a smoothing filter (cont)

To explore the IIR interpretation in more detail, and derive the convolution kernel h, we consider Fourier analysis of (9). Taking the Fourier transform (DTFT) of both sides of (9), we obtain

$$\hat{U}(\omega)\,[1+\lambda(2-e^{-i\omega}-e^{i\omega})] \;=\; \hat{V}(\omega)\,, \tag{10}$$

where $\hat{U}$ and $\hat{V}$ are the Fourier transforms of u and v, respectively. Using the identity $2\cos(\omega) = e^{-i\omega}+e^{i\omega}$, we can simplify (10) to

$$\hat{U}(\omega)\,[1+2\lambda(1-\cos(\omega))] \;=\; \hat{V}(\omega)\,. \tag{11}$$

From (11) we find

$$\hat{U}(\omega) \;=\; \hat{H}(\omega)\,\hat{V}(\omega)\,, \tag{12}$$

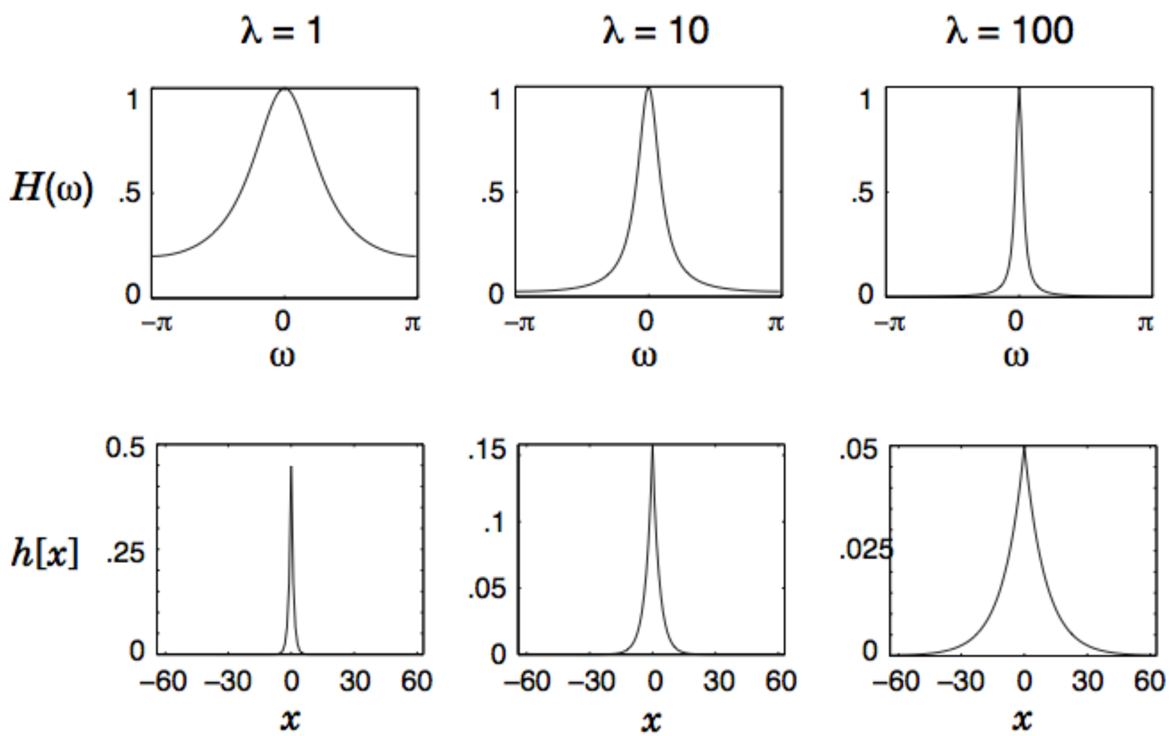
where

$$\hat{H}(\omega) \;=\; \frac{1}{1+2\lambda(1-\cos(\omega))}\,. \tag{13}$$

Therefore, by the convolution theorem, we see that $u[x]$ is simply the convolution of the input signal $v[x]$ with a filter kernel $h[x]$. Moreover, the Fourier transform of the filter kernel is $\hat{H}(\omega)$, as given in (13).

effect of pairwise term (λ)

Here are examples of h and its amplitude spectra $\hat{H}(\omega)$, for different values of λ (remember, larger λ means more emphasis on smoothness).



The effective support of the feedforward smoothing filter h can be very large, even with the same, small support of the feedback computation.

missing measurements

We therefore have N linear equations in the N unknowns:

if vn is missing

$$
\begin{pmatrix}
1+\lambda & -\lambda & 0 & 0 & \dots & 0 \\
-\lambda & 1+2\lambda & -\lambda & 0 & \dots & 0 \\
0 & -\lambda & 1+2\lambda & -\lambda & \dots & 0 \\
0 & \cdots & -1 & 2 & -1 \cdots & 0 \\
 & & \ddots & \ddots & \ddots & \\
 & & & & & \\
0 & \dots & & 0 & -\lambda & 1+\lambda
\end{pmatrix}
\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \\ \vdots \\ \\ u_N \end{pmatrix}
=
\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \\ \vdots \\ \\ v_N \end{pmatrix}
\tag{7}
$$

if vn is missing

$$E(u) = \sum_{n=1}^{N} (\cancel{u_n - v_n})^2 + \lambda \sum_{n=1}^{N-1} (u_{n+1} - u_n)^2 \,.$$

$$\frac{\partial E(u)}{\partial u_n} = 2\,(\cancel{u_n - v_n}) \; + \; 2\lambda\,(-u_{n-1} + 2u_n - u_{n+1}) \,,$$

$$\cancel{u_n} \; + \; \lambda\,(-u_{n-1} + 2u_n - u_{n+1}) \; = \; v_n$$

image smoothing in 2D

For 2D images, the analogous energy we want to minimize becomes

$$E(u) = \sum_{n,m \in P} (u[n,m] - v[n,m])^2 \quad \leftarrow \text{data fidelity term}$$

$$+ \lambda \sum_{\text{all } n,m} (u[n+1,m] - u[n,m])^2 + (u[n,m+1] - u[n,m])^2 \quad (20)$$

$\leftarrow$ smoothness term

where P is a subset of pixels where the measurements v are available.

image smoothing in 2D

Taking derivatives with respect to $u[n,m]$ and setting them equal to zero yields a linear system of equations that has the same form as (9). The only difference is that the linear filter g is now 2D: e.g.,

$$g = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} .$$

One can again solve for u iteratively, where, ignoring the edge pixels for simplicity, we have

$$u^{(t+1)}[n,m] = \begin{cases} \frac{1}{1+4\lambda}(v[n,m] + \lambda\, s^{(t)}[n,m]) & \text{for } n,m \in P\,, \\ \frac{1}{4}\, s^{(t)}[n,m] & \text{otherwise}\,, \end{cases} \tag{21}$$

where $s[n,m]$ is the sum of the 4 neighbors of pixel $[n,m]$, i.e., $u[n-1,m] + u[n+1,m] + u[n,m-1] + u[n,m+1]$.

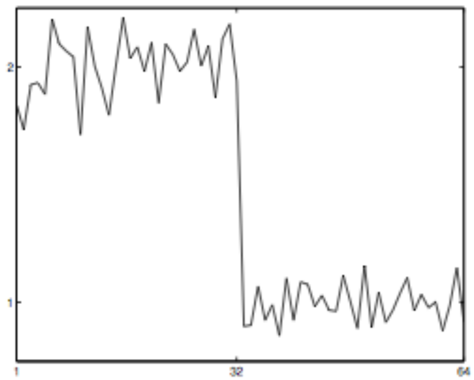
Topic 16:

Markov Random Fields (MRFs)

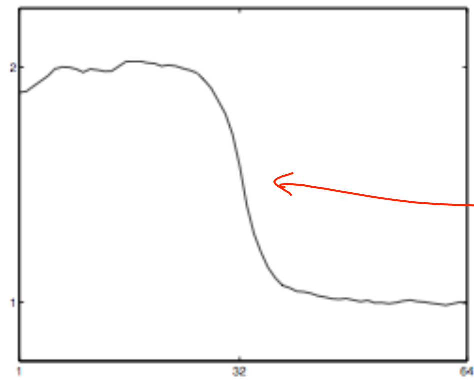
- MRFs & energy minimization
- continuous MRFs
  - quadratic potentials (Gaussian MRFs)
  - robust/non-convex potentials (robust regularization)
- discrete MRFs
  - binary MRFs: Ising model & graph cuts
  - multi-valued MRFs: Potts model & expansion moves
- applications: grabcut, texture synthesis, PEARL

robust potentials

Smoothing a noisy step edge:



Noisy step



LS smoother

step edge has been blurred from smoothing

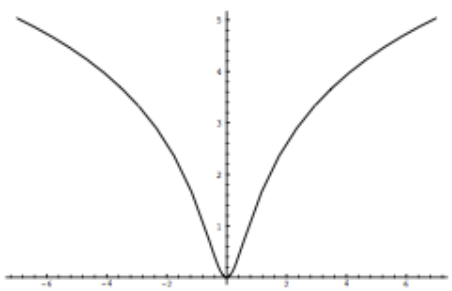
robust potentials

Instead of quadratic potentials, we could use a robust error function ρ:

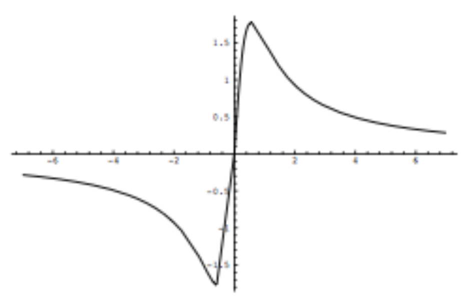
$$E(u) = \sum_{n=1}^{N} \rho(u_n - v_n,\, \sigma_d) + \lambda \sum_{n=1}^{N-1} \rho(u_{n+1} - u_n,\, \sigma_s)\,, \tag{22}$$

where σ_d and σ_s are scale parameters. For example, the *Lorentzian* error function is given by

$$\rho(z,\sigma) = \log\left(1 + \frac{1}{2}\left(\frac{z}{\sigma}\right)^2\right), \quad \rho'(z,\sigma) = \frac{2z}{2\sigma^2 + z^2}. \tag{23}$$



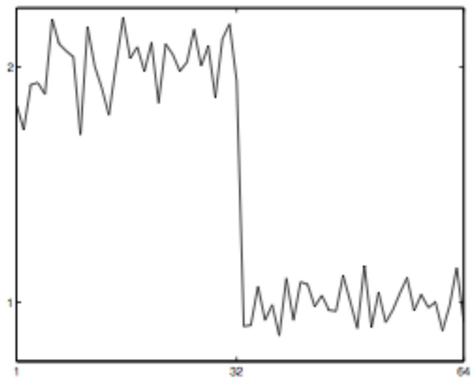
Error function



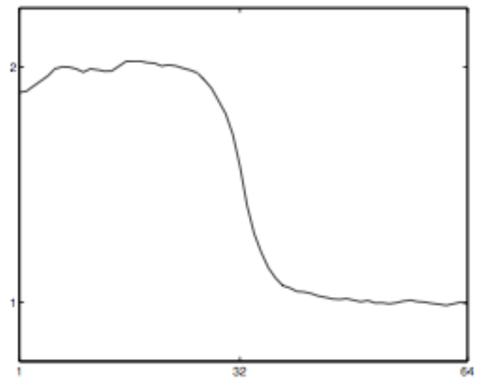
Influence function

robust potentials

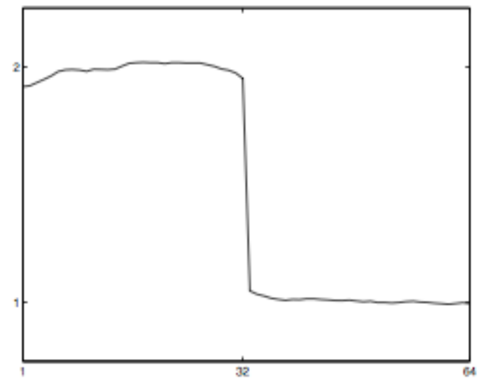
Smoothing a noisy step edge:



Noisy step



LS smoother



Lorentzian smoother

Original image

Output of robust smoothing

robust potentials

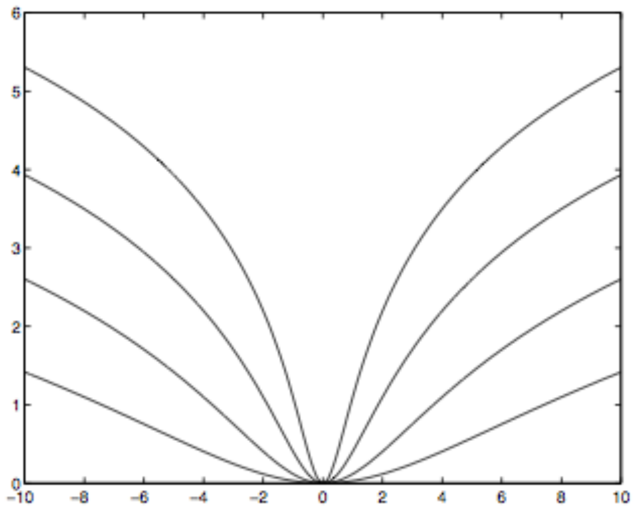
Edges

Original image

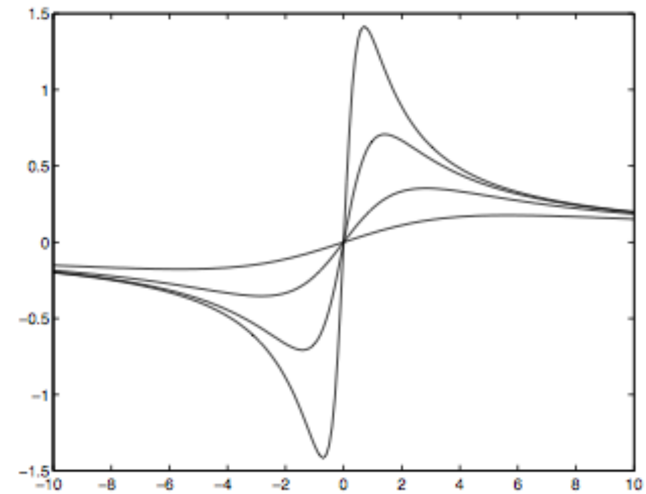
Output of robust smoothing

graduated non-convexity

For example, the plots below depicts Lorentzian error and influence functions for four values of σ (i.e., for $\sigma = 0.5, 1, 2,$ and 4). For larger σ, the error functions become more like a quadratic, and the influence functions become more linear. I.e., the nonconvex Lorentzian error function becomes a simple (convex) quadratic when σ is very large.



Lorentzian error functions



Lorentzian influence functions

Example of Graduated Non-Convexity: To compute the Lorentzian-smoothed version of the noisy step edge on p. 12, we initially set $\sigma_s = 10$ and then gradually reduced it to 0.1.

Topic 16:

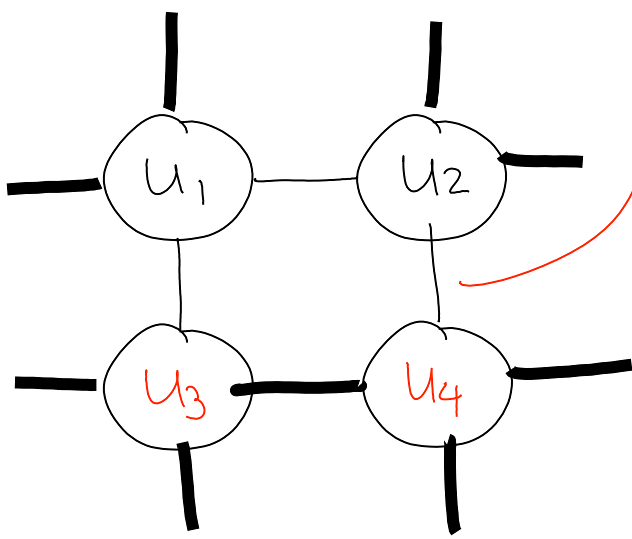
Markov Random Fields (MRFs)

- MRFs & energy minimization
- continuous MRFs
  - quadratic potentials (Gaussian MRFs)
  - robust/non-convex potentials (robust regularization)
- discrete MRFs
  - binary MRFs: Ising model & graph cuts
  - multi-valued MRFs: Potts model & expansion moves
- applications: grabcut, texture synthesis, PEARL

binary MRFs

Ising model: A binary MRF, $u_i \in \{0, 1\}$, with 4-connected neighbourhoods, and interaction clique potentials given by

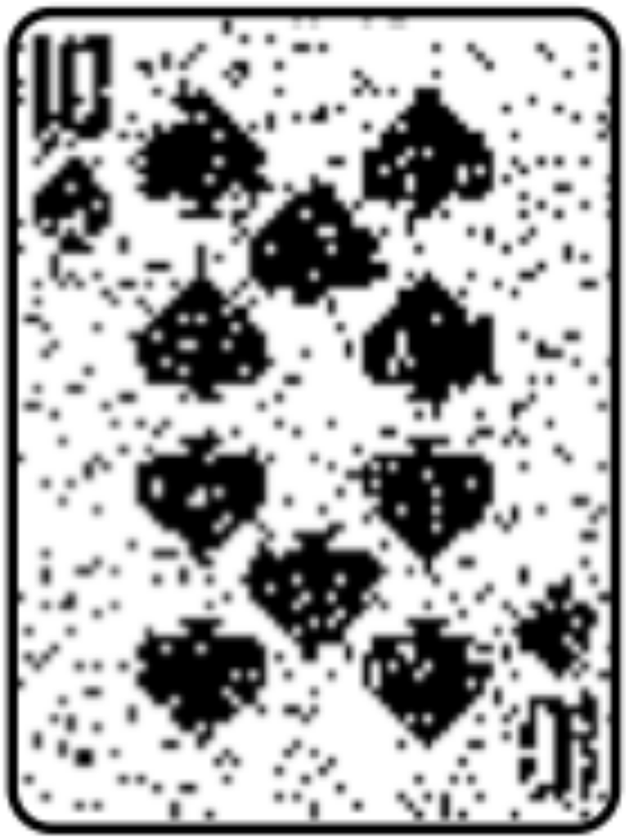
$$V(u_i, u_j) = \beta \, |u_i - u_j| \, , \quad \beta > 0 \, . \qquad (25)$$



$$\begin{cases} 0 & \text{if } u_2 = u_4 \\ \beta & \text{if } u_2 \neq u_4 \end{cases}$$

example: binary image denoising

starting from a noiseless binary image, randomly flip bits with probability θ



MRF energy for binary image denoising

Binary Image Denoising: Let u be a binary image, and let v be a noisy version of u; i.e., with probability θ we randomly flip bits in u. With 4-connected neighbourhoods, the energy function is

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j) \tag{26}$$

Use the Ising model for the interaction potentials (25). The unary potentials are simply the negative log data likelihood, i.e.,

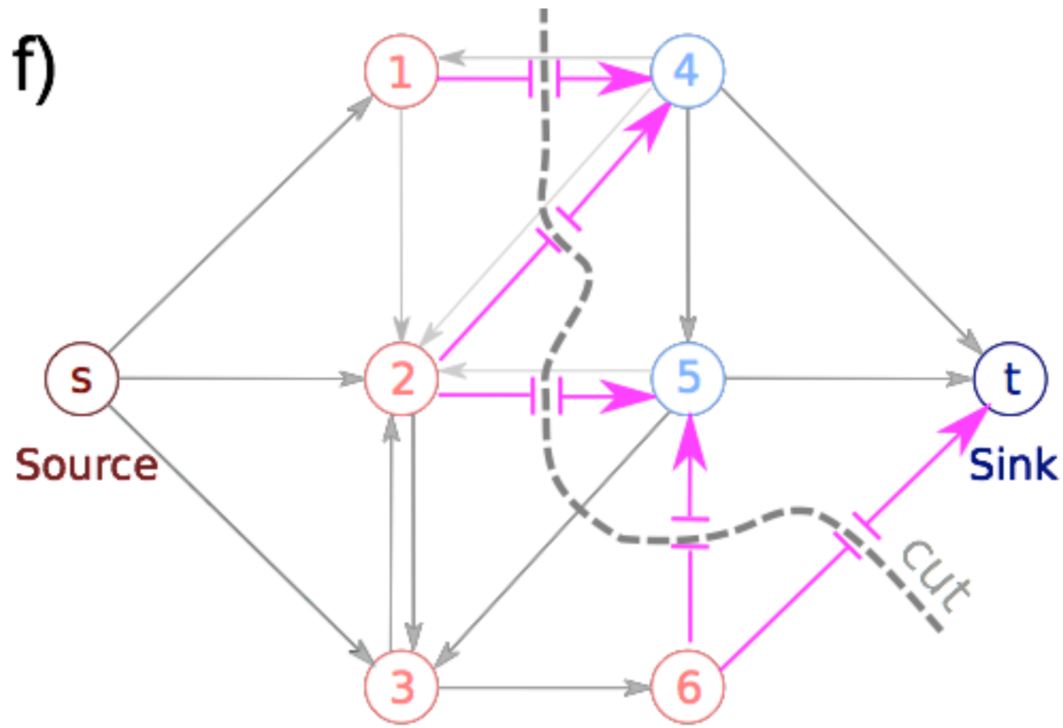
$$D(u_j) = \begin{cases} -\log(1-\theta) & \text{for } u_j = v_j \\ -\log\theta & \text{for } u_j \neq v_j \end{cases} \tag{27}$$

Goal: Find the signal u that minimizes the energy $E(u)$.

MRF optimization as a graph cut problem

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

We seek a graph & assigment of capacities such that min $E(u)$ = min-cut



aside: the max-flow problem

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

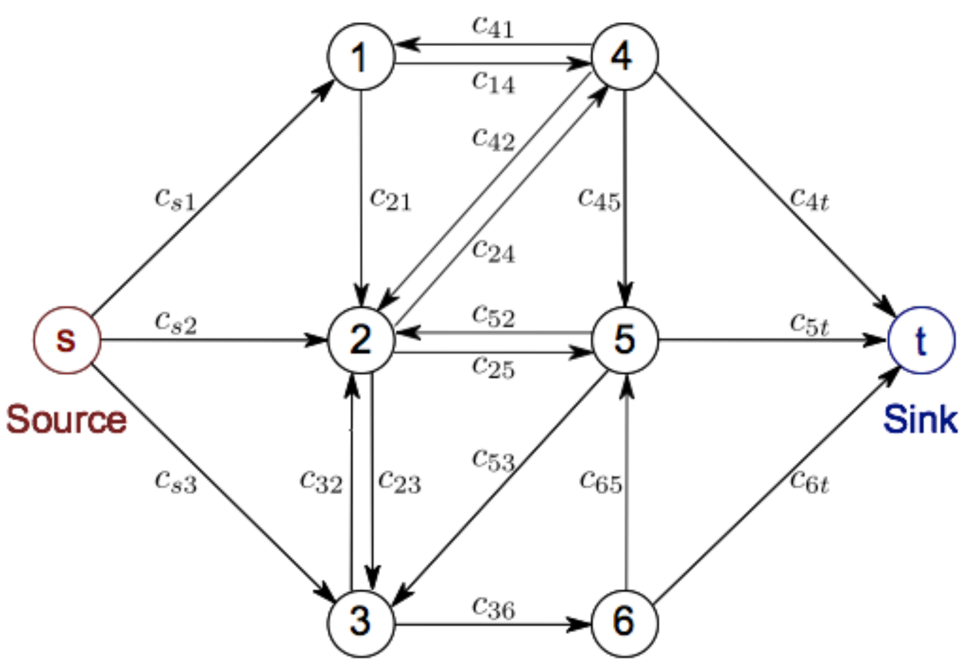
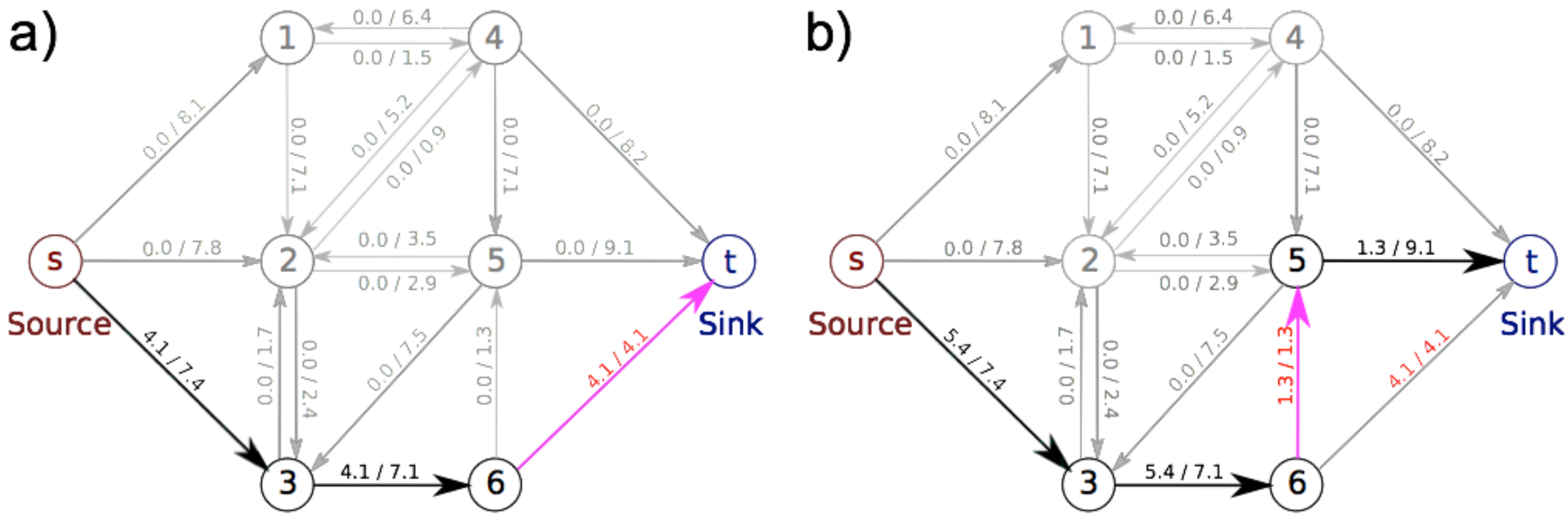


Figure 12.5 Max flow problem: we are given a network of vertices connected by directed edges, each of which has a non-negative capacity c_{mn}. There are two special vertices s and t termed the source and sink, respectively. In the max-flow problem, we seek to push as much 'flow' from source to sink while respecting the capacities of the edges.

aside: augmenting paths algorithm for max-flow

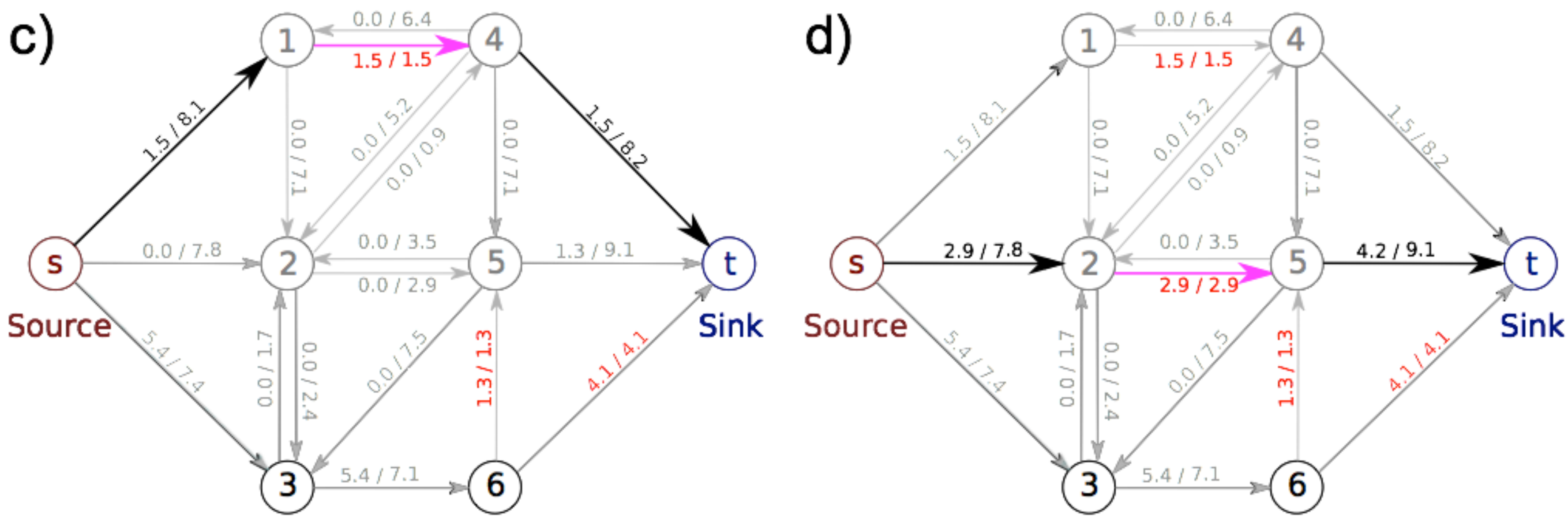


Idea: ① find a path from ⓢ to ⓣ with maximum spare capacity

② push enough flow to saturate it

③ repeat

aside: augmenting paths algorithm for max-flow

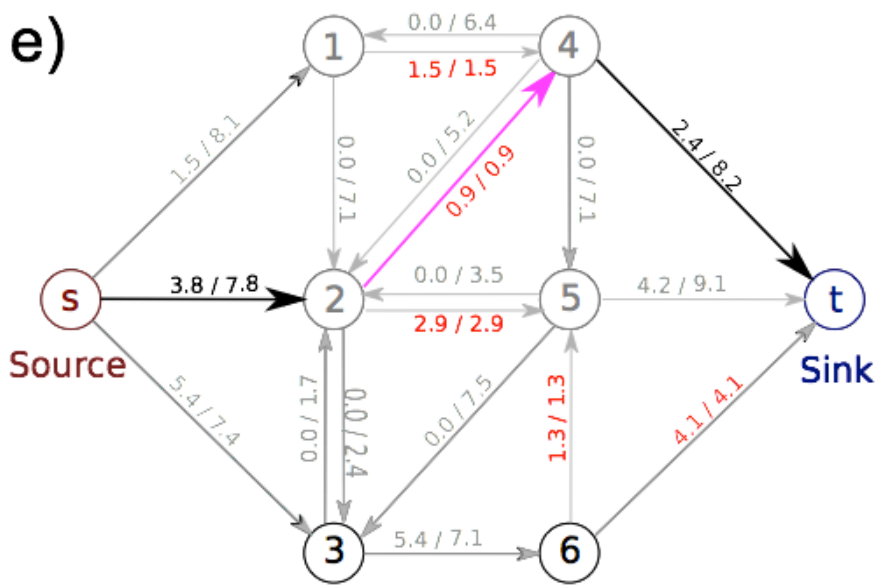


Idea: ① find a path from (s) to (t) with maximum spare capacity

② push enough flow to saturate it

③ repeat

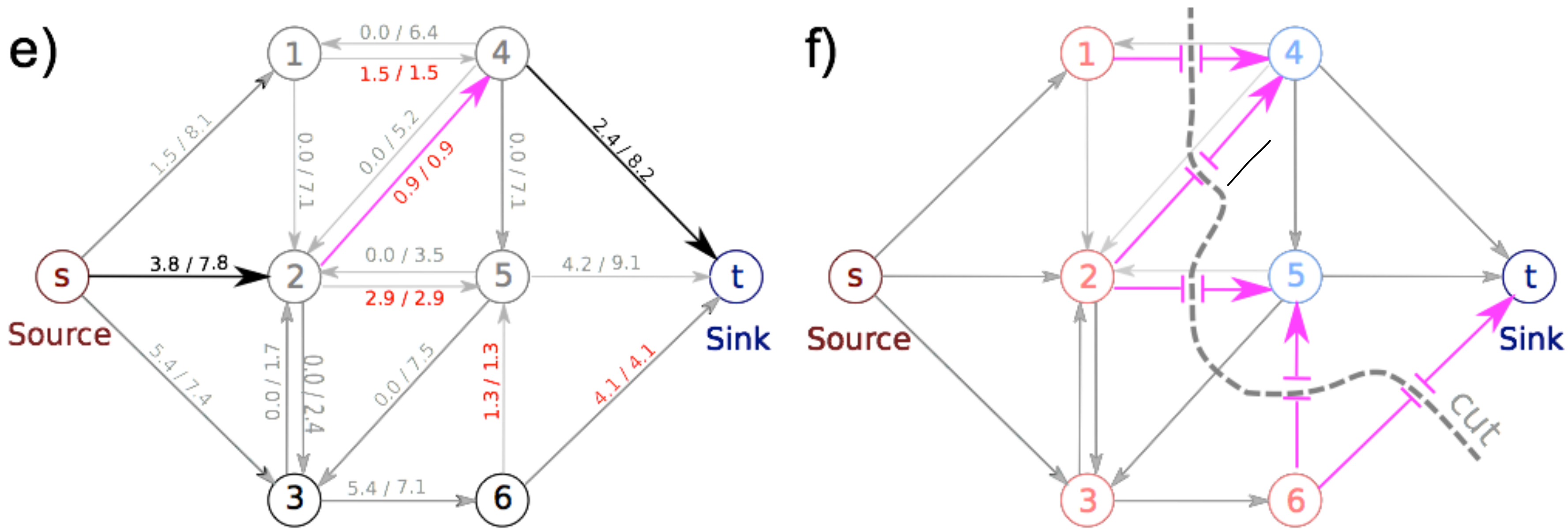
aside: augmenting paths algorithm for max-flow



Idea:

1. find a path from s to t with maximum spare capacity
2. push enough flow to saturate it
3. repeat

aside: equivalence of max-flow & min-cut

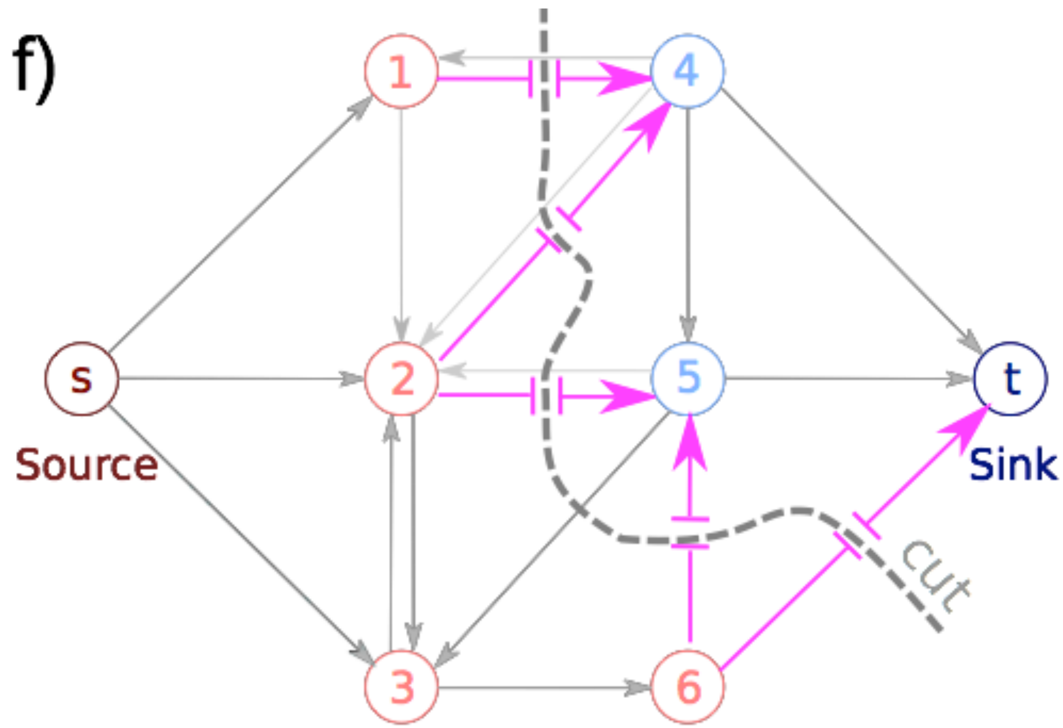


* When max flow reached, all paths from s to t have reached capacity
* a cut : set of edges that disconnects s, t
* max-flow = min sum of edge capacities in a cut

MRF optimization as a graph cut problem

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

We seek a graph & assigment of capacities such that min $E(u)$ = min-cut



potentials globally-optimizable using graph cuts

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V_{ij}(u_i, u_j)$$

(Kolmogorov & Zabih, 2002-4)

Energies with a pairwise potential $V_{ij}(u_i, u_j)$ can be globally optimized using graph cuts if and only if they satisfy

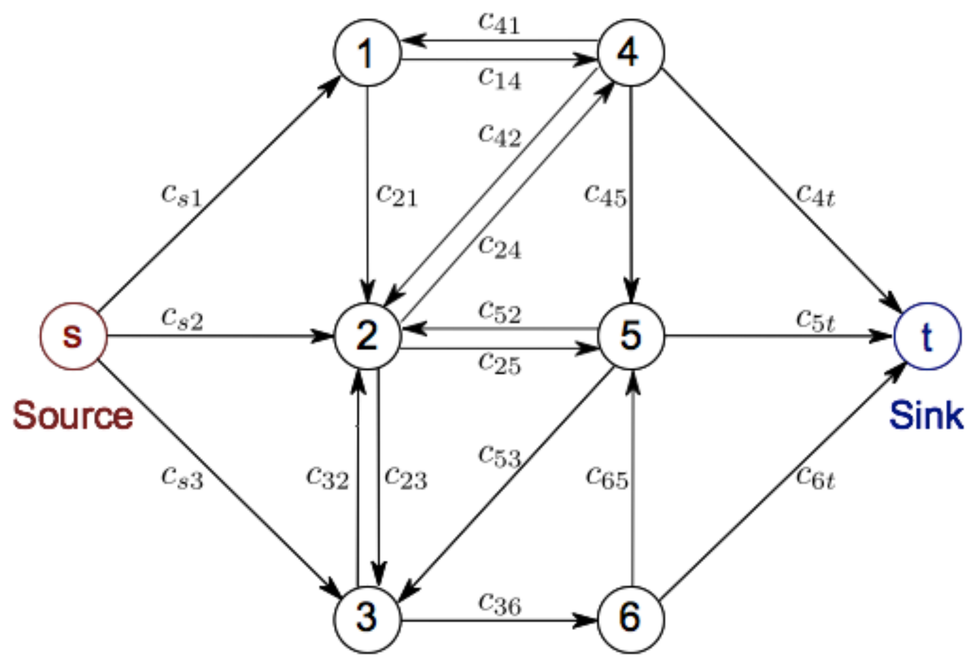
sub-modularity condition

$$\underbrace{V_{ij}(0,0) + V_{ij}(1,1)}_{\text{costs for being the same}} \leq \underbrace{V_{ij}(1,0) + V_{ij}(0,1)}_{\text{costs for being different}}$$

graph construction for MRF energy minimization

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

We seek a graph & assigment of capacities such that min $E(u)$ = min-cut

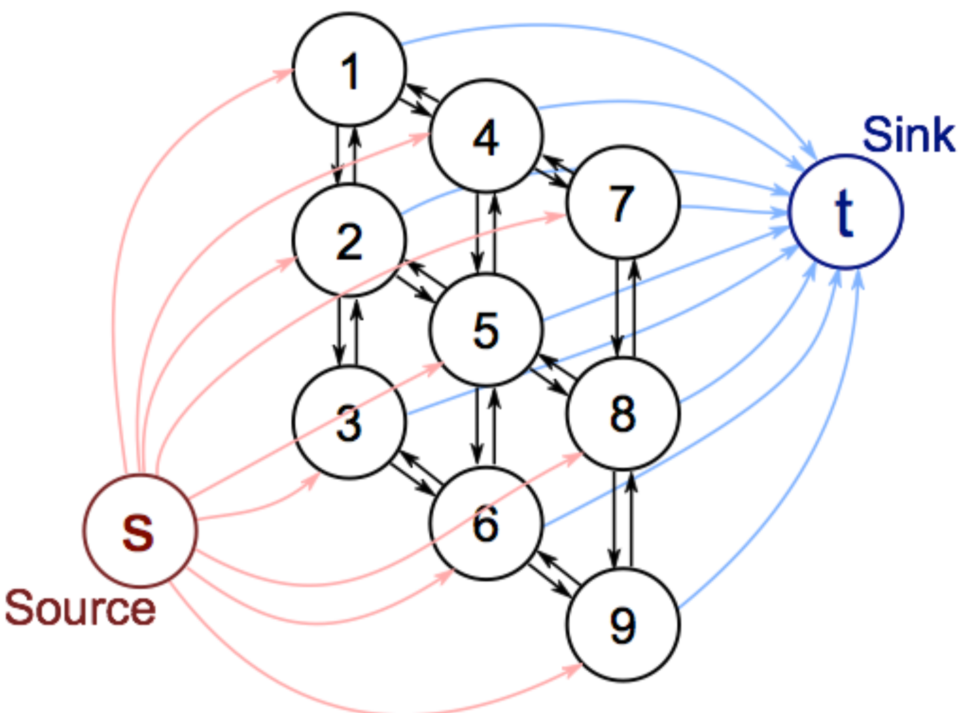


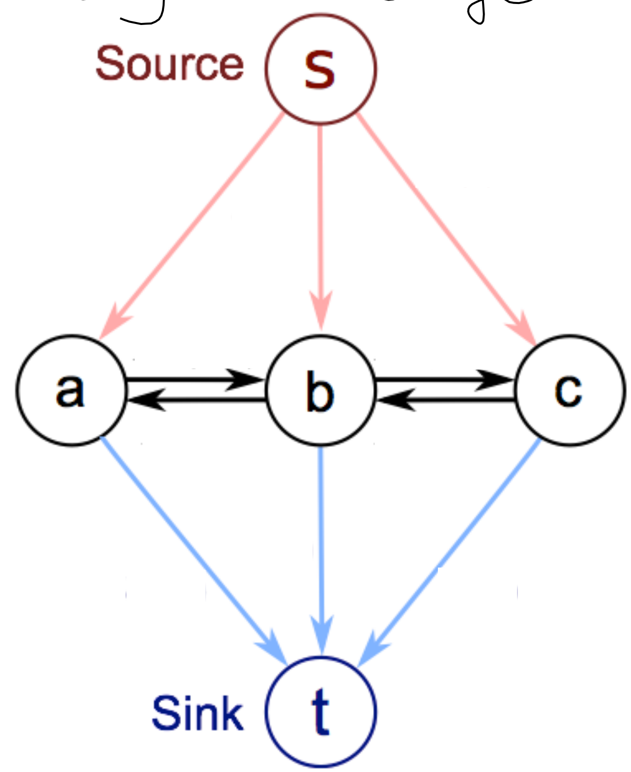
graph construction: vertices & edges

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

1. add (s) and (t) and edges to all MRF variables

2. add 2 directed edges for every MRF edge



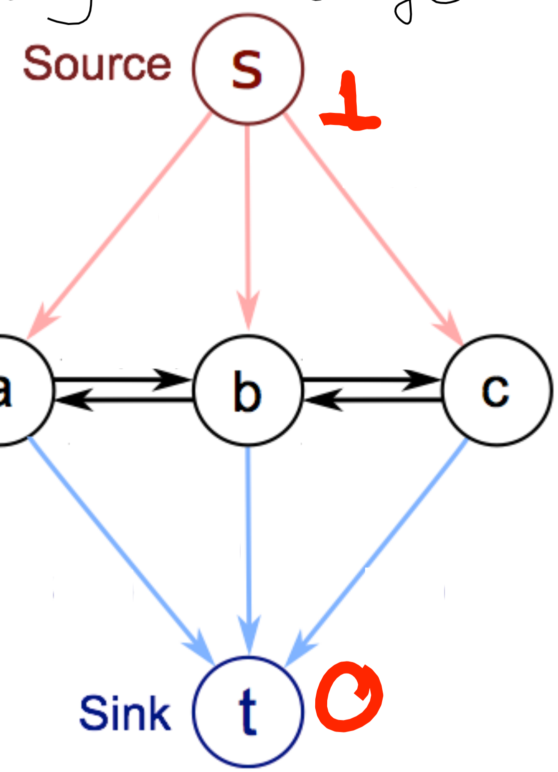


graph construction: vertices & edges

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

1. add s and t and edges to all MRF variables

2. add 2 directed edges for every MRF edge



3. after cut:
   - vars connected to s are set to 1
   - vars connected to t are set to 0

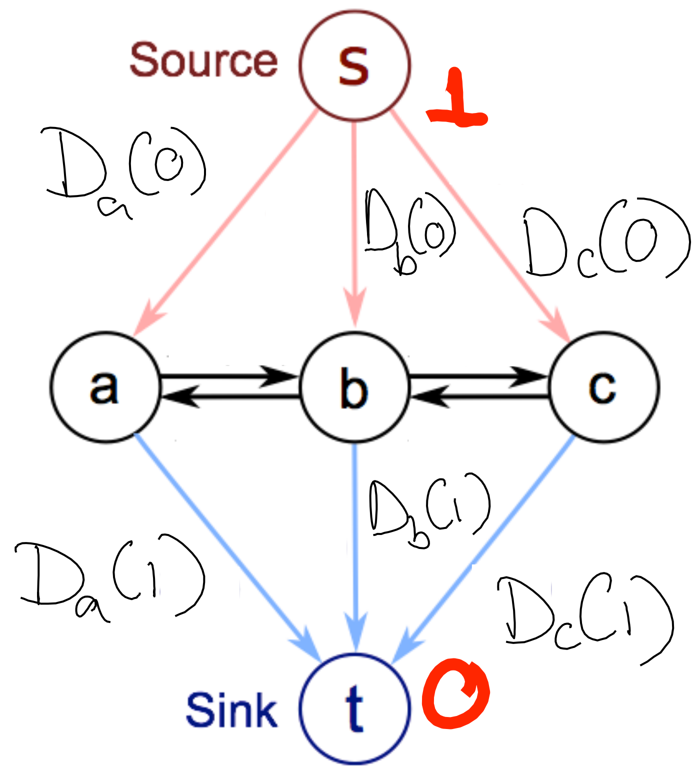
graph construction: capacities of edges to s, t

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

Case 1: 0-diagonal potential

b \ a	0	1
0	0	$V_{ab}(1,0)$
1	$V_{ab}(0,1)$	0

Edges to s, t have capacities given by the data term



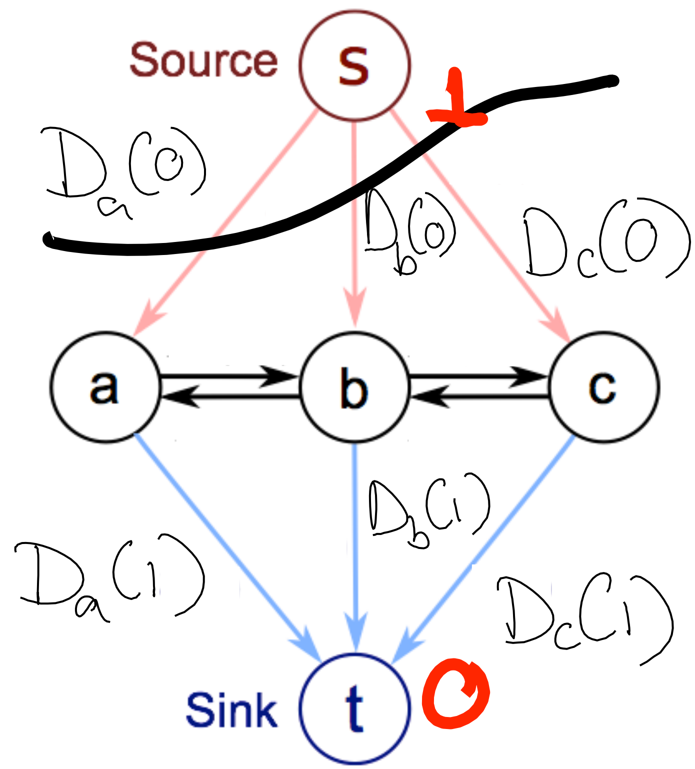
graph construction: capacities of edges to s, t

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

Case 1: 0-diagonal potential

b \ a	0	1
0	0	$V_{ab}(1,0)$
1	$V_{ab}(0,1)$	0

$$E(0,0,0) = D_a(0) + D_b(0) + D_c(0)$$



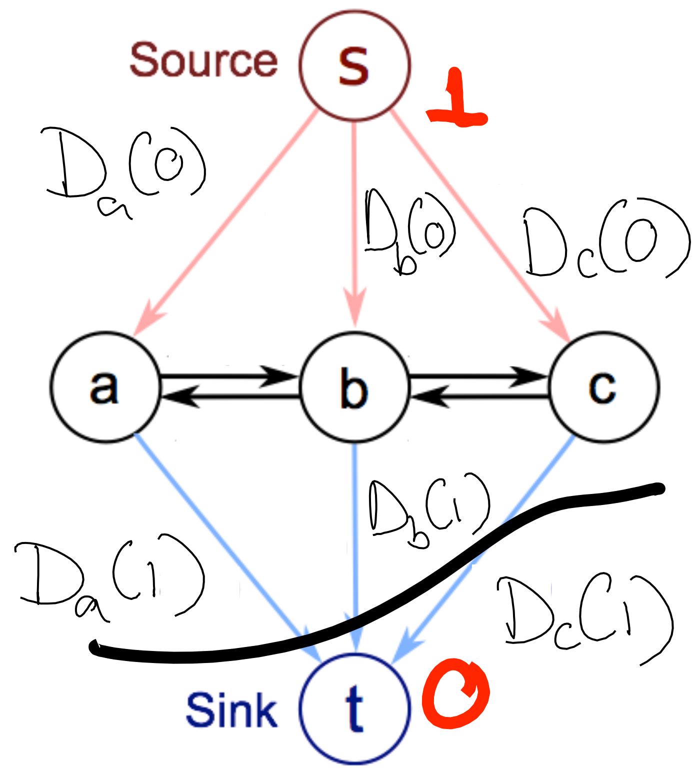
graph construction: capacities of edges to s, t

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

Case 1: 0-diagonal potential

b \ a	0	1
0	0	$V_{ab}(1,0)$
1	$V_{ab}(0,1)$	0

$$E(1,1,1) = D_a(1) + D_b(1) + D_c(1)$$



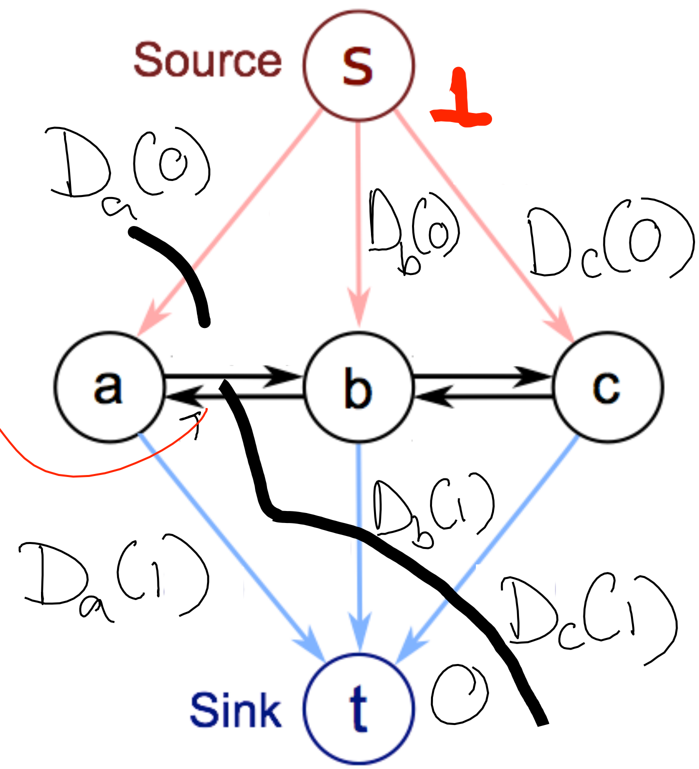
graph construction: capacities of internal edges

$$E(u) = \sum_{i\in\mathcal{V}} D(u_i) + \sum_{(i,j)\in\mathcal{E}} V(u_i,u_j)$$

Case 1: 0-diagonal potential

b \ a	0	1
0	0	$V_{ab}(1,0)$
1	$V_{ab}(0,1)$	0

capacity must be $V_{ab}(0,1)$



$$E(0,1,1) = D_a(0) + D_b(1) + D_c(1) + V_{ab}(0,1)$$

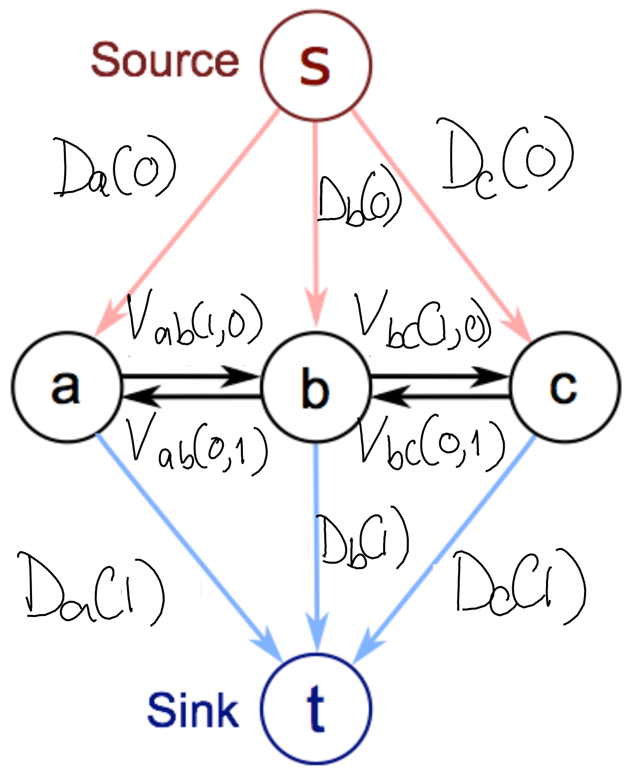
graph construction: capacities of internal edges

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

Case 1: 0-diagonal potential

b \ a	0	1
0	0	$V_{ab}(1,0)$
1	$V_{ab}(0,1)$	0

For 0-diagonal potentials, internal edge capacities = potentials



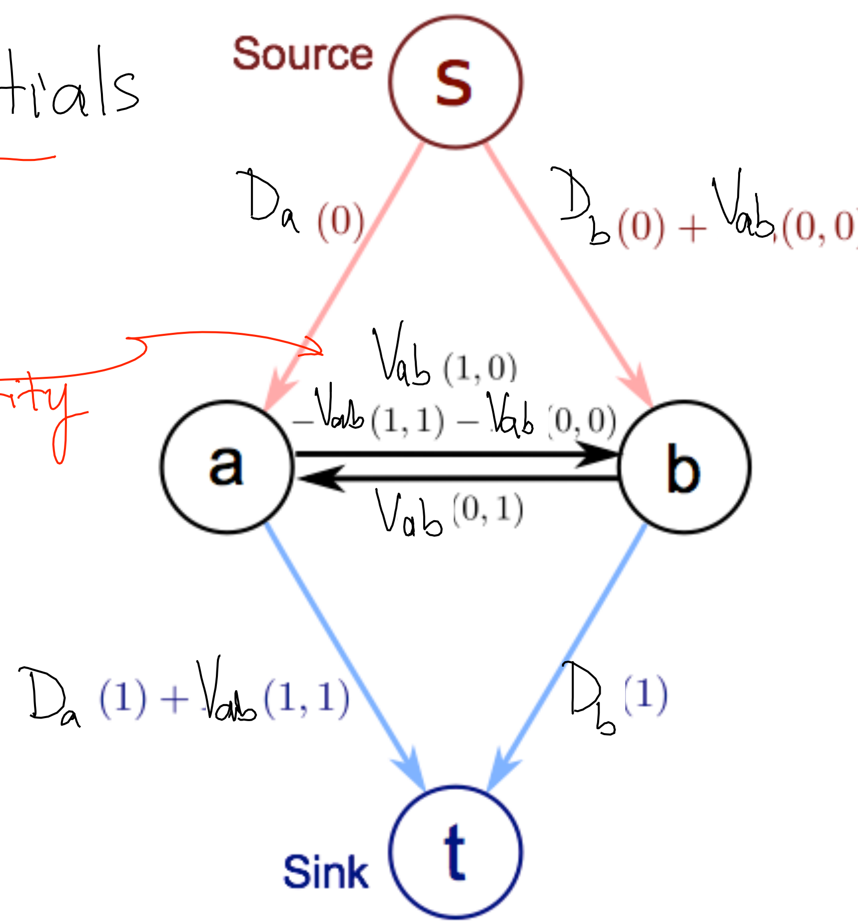
graph construction: non-zero-diagonal potentials

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

Case 2: non 0-diagonal potentials

b \ a	0	1
0	$V_{ab}(0,0)$	$V_{ab}(1,0)$
1	$V_{ab}(0,1)$	$V_{ab}(1,1)$

requires sub-modularity



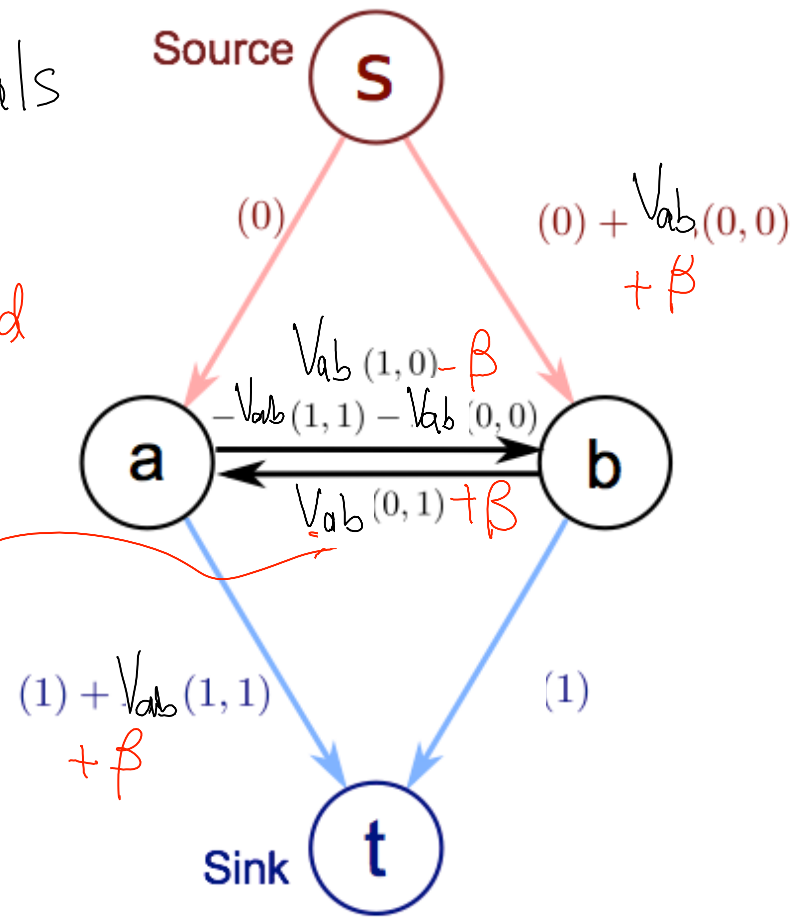
graph construction: non-zero-diagonal potentials

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

Case 2: non 0-diagonal potentials

b \ a	0	1
0	$V_{ab}(0,0)$	$V_{ab}(1,0)$
1	$V_{ab}(0,1)$	$V_{ab}(1,1)$

may need to add a constant β to ensure positivity for all capacities



(See Prince, 2011 for the details)

Topic 16:

Markov Random Fields (MRFs)

- MRFs & energy minimization
- continuous MRFs
  - quadratic potentials (Gaussian MRFs)
  - robust/non-convex potentials (robust regularization)
- discrete MRFs
  - binary MRFs: Ising model & graph cuts
  - multi-valued MRFs: Potts model & expansion moves
- applications: grabcut, texture synthesis, PEARL

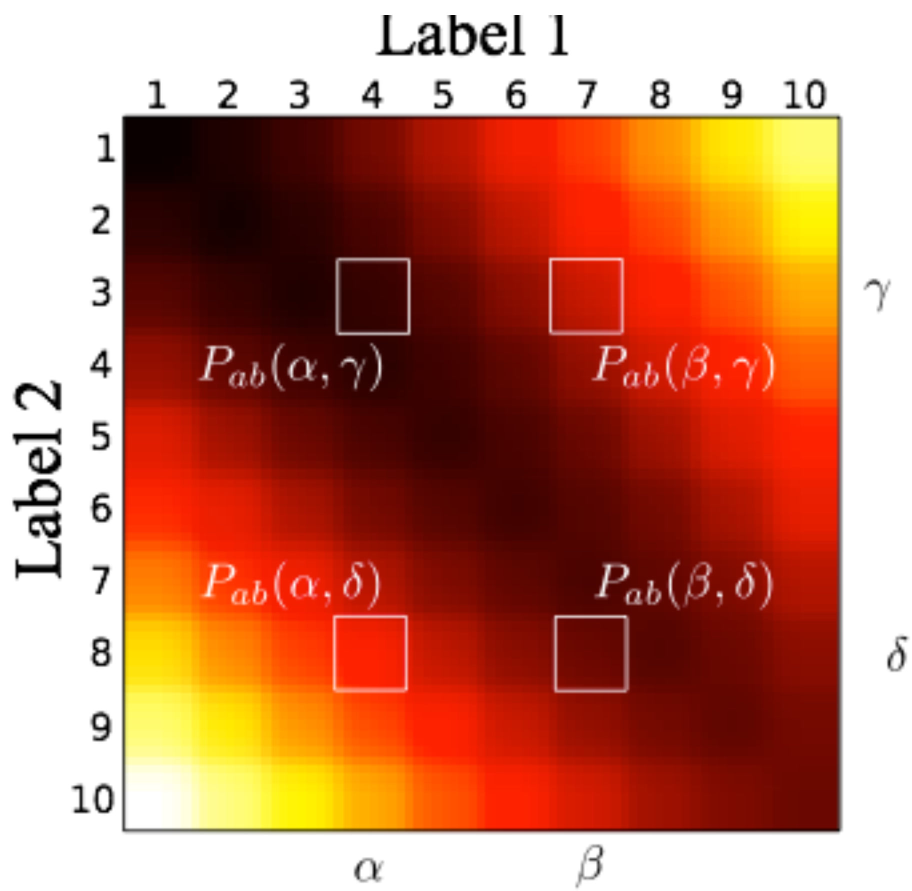
multi-label MRFs with sub-modular potentials

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V_{ij}(u_i, u_j)$$

condition for global optimization becomes

$$V_{ij}(\beta, \gamma) + V_{ij}(\alpha, \delta) \geq$$

$$V_{ij}(\alpha, \gamma) + V_{ij}(\beta, \delta)$$



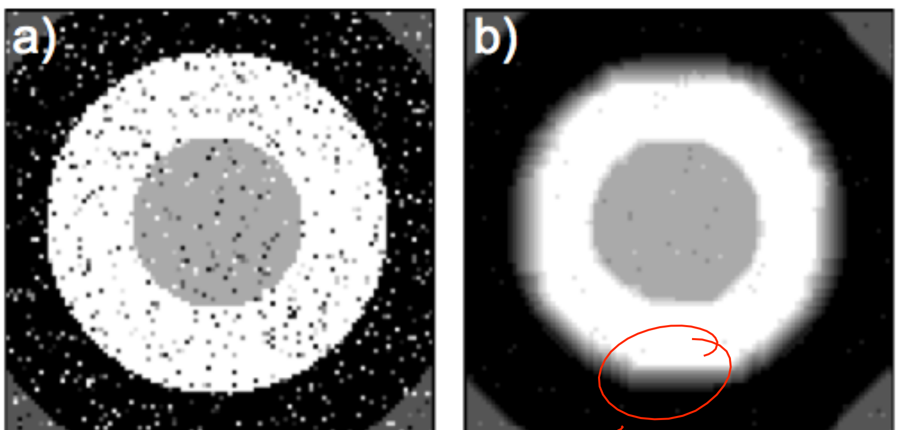
multi-label MRFs with sub-modular potentials

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

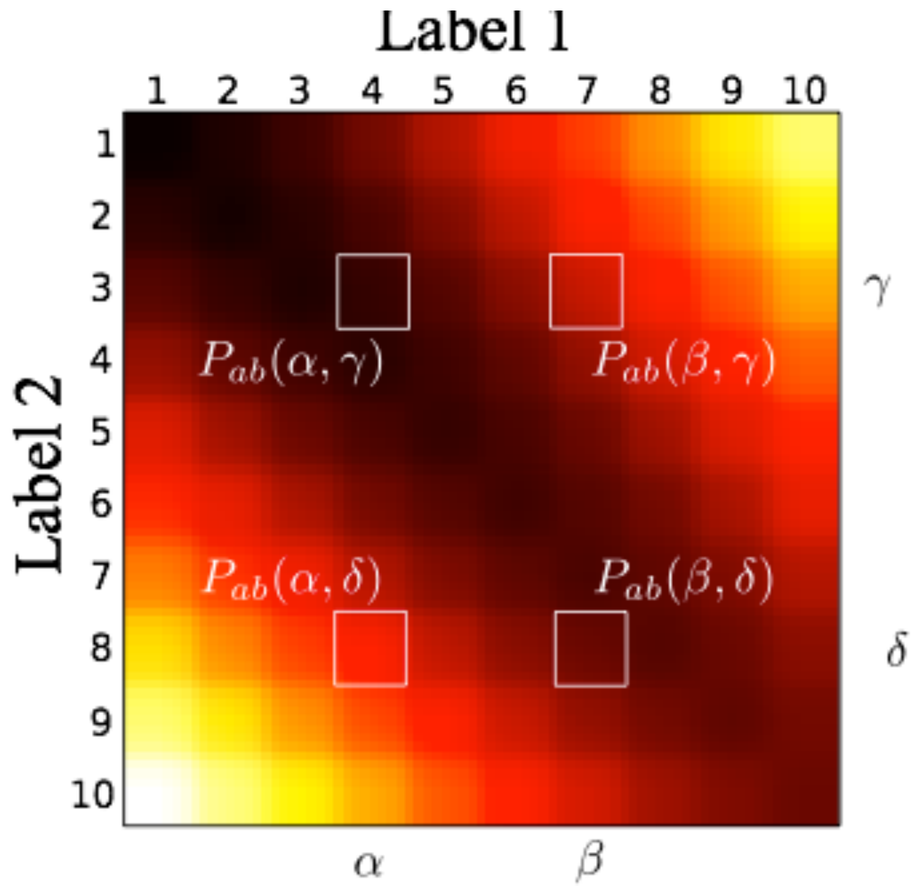
Example:

$$V_{ij}(a,b) = k(a-b)^2$$

constant



potential not robust $\Rightarrow$ blurred edges



multi-label MRFs with non-convex potentials

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

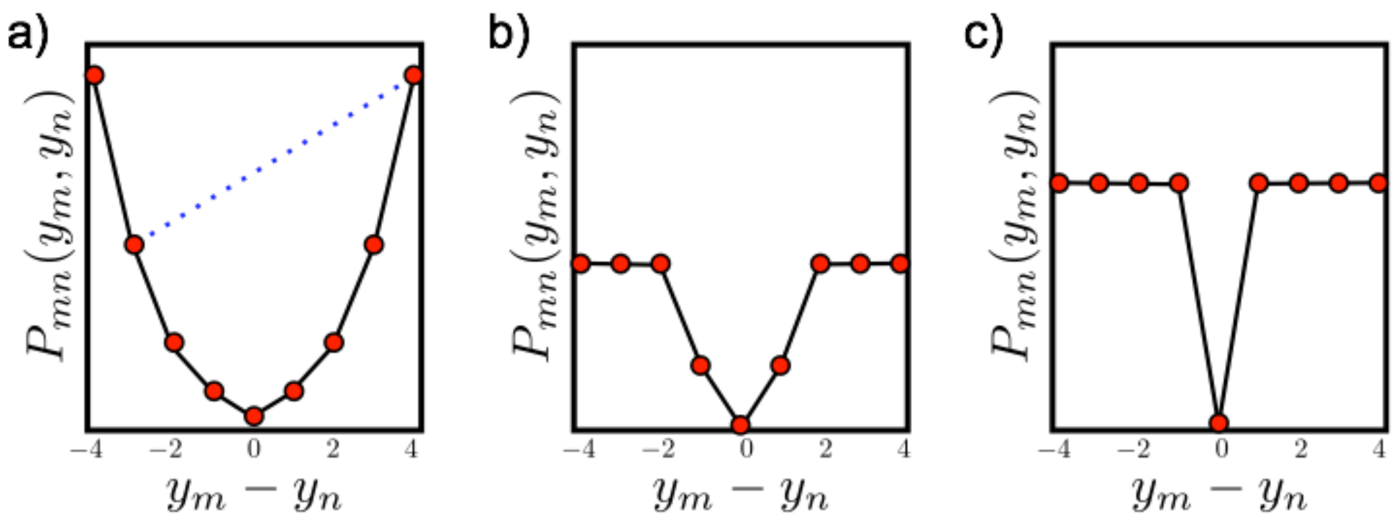


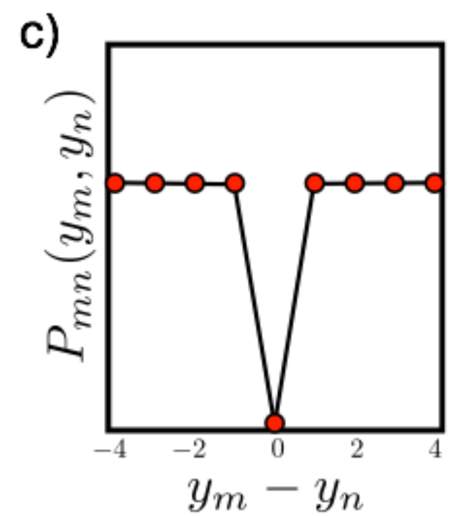
Figure 12.17 Convex vs. non-convex potentials. The method for MAP inference for multi-valued variables depends on whether the costs are a convex or non-convex function of the difference in labels. a) Quadratic function (convex), $P_{mn}(w_m, w_n) = \kappa(w_m - w_n)^2$. For convex functions, it is possible to draw a chord between any two points on the function without intersecting the function elsewhere (e.g., dotted blue line). b) Truncated quadratic function (non-convex), $P_{mn}(w_m, w_n) = \min(\kappa_1, \kappa_2(w_m - w_n)^2)$. c) Potts model (non-convex), $P_{mn}(w_m, w_n) = \kappa(1 - \delta(w_m - w_n))$.

multi-label MRFs: Potts model

$$E(u) = \sum_{i \in \mathcal{V}} D(u_i) + \sum_{(i,j) \in \mathcal{E}} V(u_i, u_j)$$

$$V_{ij}(a,b) = \begin{cases} 0 & \text{if } a = b \\ k & \text{if } a \neq b \end{cases}$$

ensures robustness

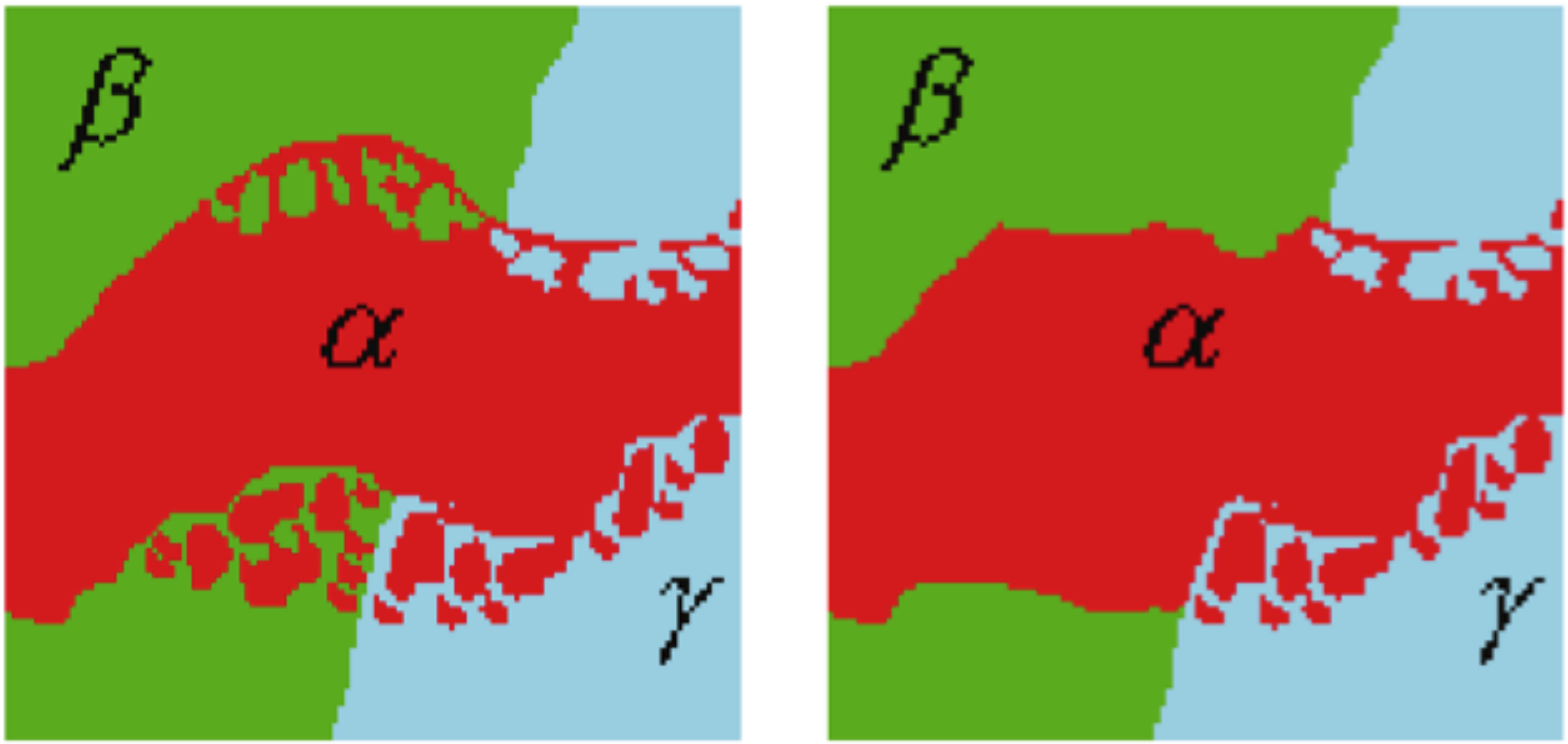


Unfortunately, the MRF cannot be globally optimized efficiently ($\Rightarrow$ use local optimization)

local binary optimization by single-pixel "moves"

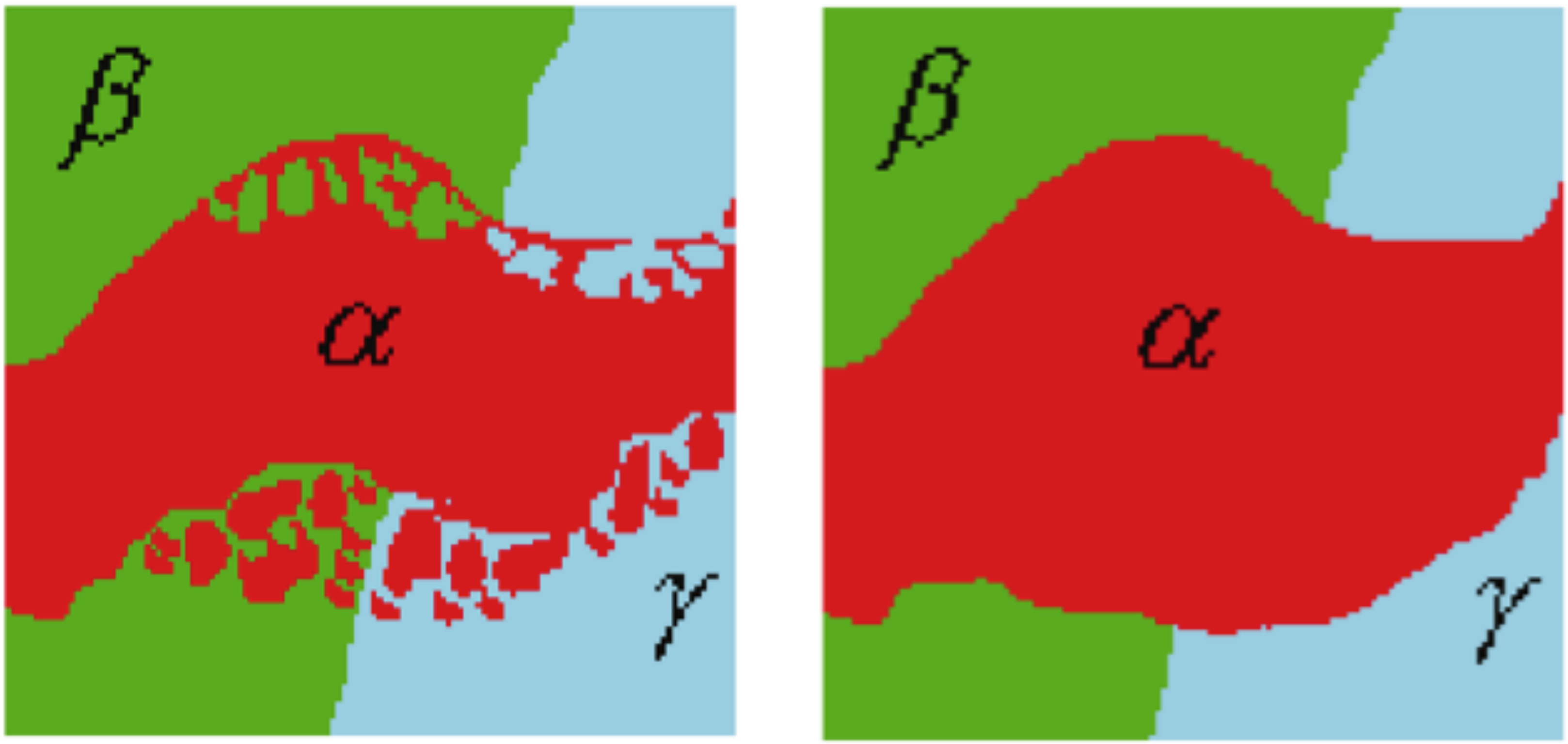
"move" = change the label for a subset of pixels (here: just 1 pixel)

local binary optimization by large moves: αβ-swap



αβ-swap: for all pixels with label α
decide if they should keep label
α or change it to β

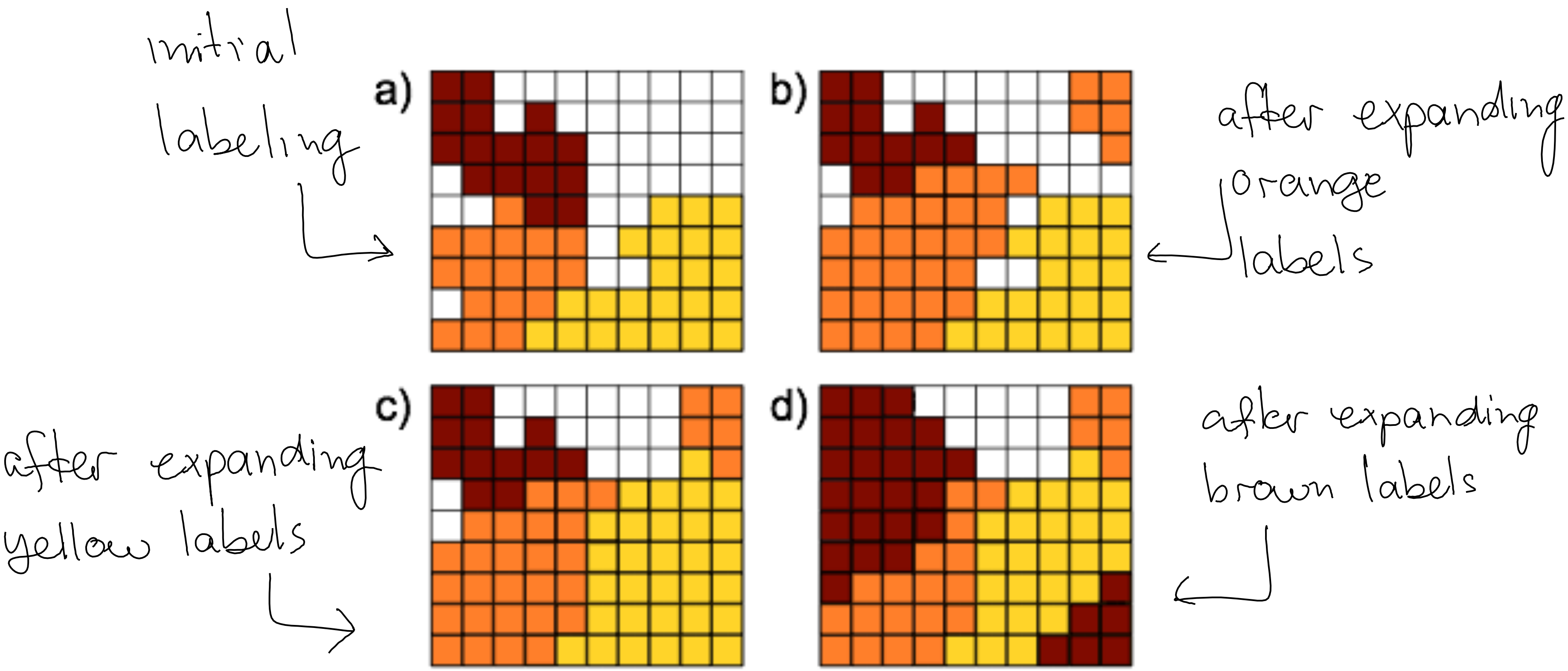
local optimization by large moves: α-expansion



α- expansion : for all pixels whose label is <u>not</u> α, decide if they should keep their current label or change it to α

local binary optimization by α-expansion

The general case for multi-valued MRFs remains NP-hard, **but** we can now use binary min-cut to find much better ”local” (greedy) moves. These local moves avoid bad local minima, and can be shown to come within a factor of 2 of the energy minimum.



local binary optimization by α-expansion

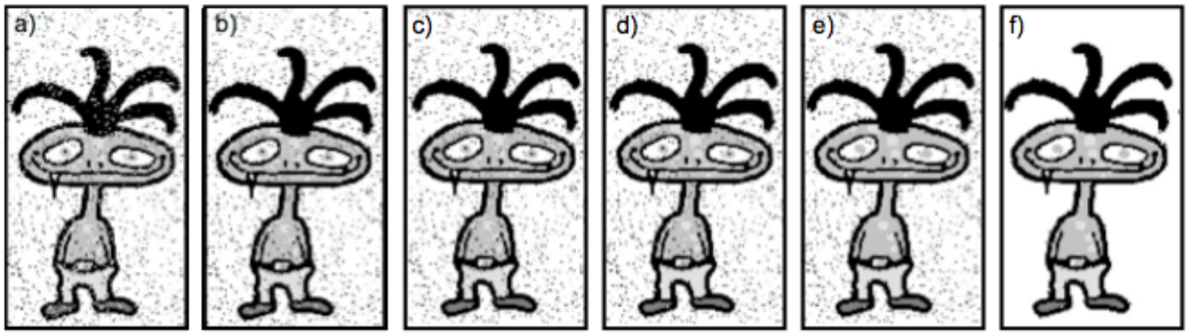
The general case for multi-valued MRFs remains NP-hard, **but** we can now use binary min-cut to find much better ”local” (greedy) moves. These local moves avoid bad local minima, and can be shown to come within a factor of 2 of the energy minimum.

α-**Expansions:** For a given label $\alpha \in \mathcal{L}$, let any RV whose current label is in $\mathcal{L} \backslash \alpha$ either switch to α, or remain the same. Given a labeling u, and the label α, construct a new graph such that the min-cut labeling $\hat{u}$ minimizes $E(\hat{u})$.

This can be shown to reduce the global energy with min-cut as long as the interaction potential is a metric (satisfying the triangle inequality).

α-expansion algorithm

Algorithm: Iteratively cycle through all labels, applying α-expansion moves until the energy stops decreasing.



[Prince, 2011, Chapter 12]

(a) The noisy image. (b) Step 1 cleans up the hair. Step 2 does nothing (the label has no image support). (c-f) Denoising of the boots, trousers, skin, and background.

See [Boykov, Veksler and Zabih, 2001] for the graph construction.

α-expansion: graph construction example

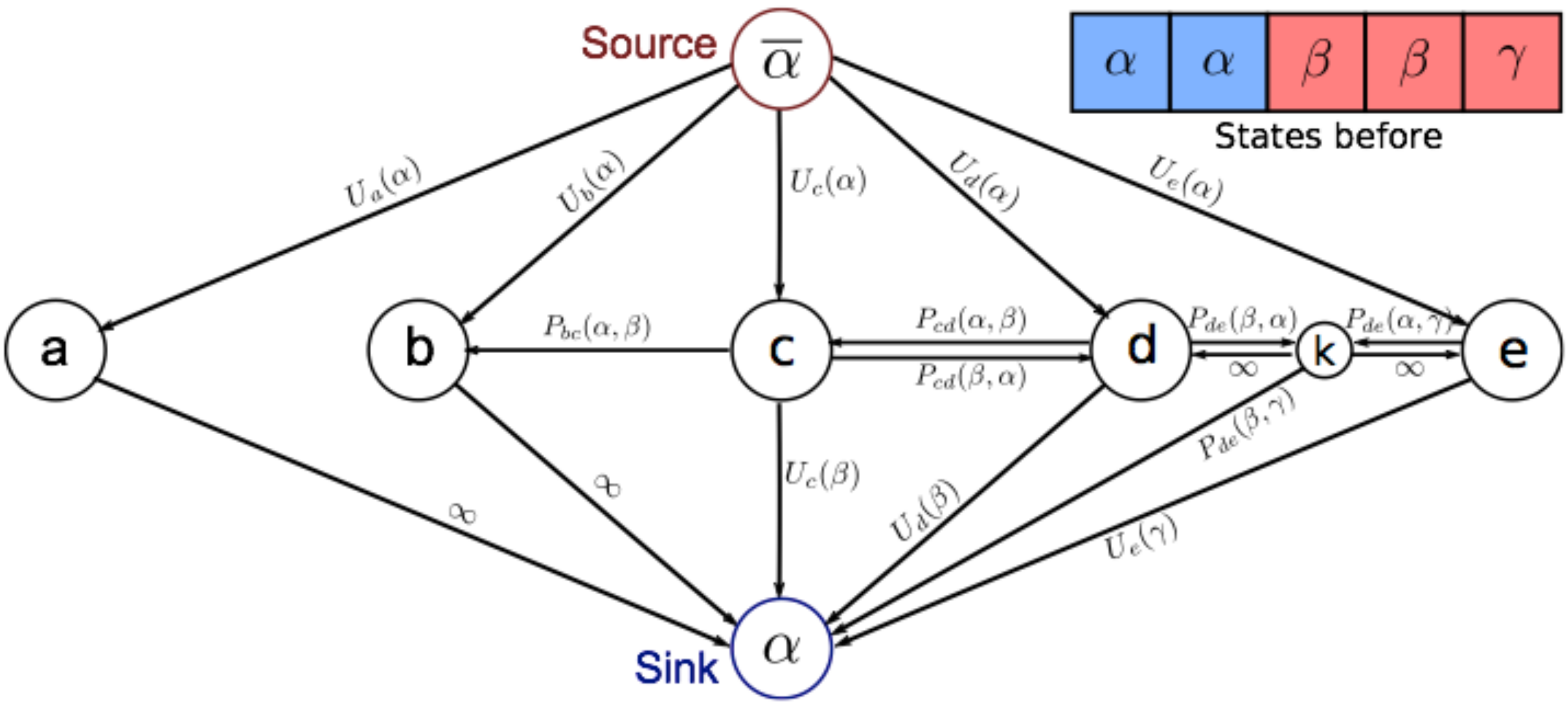


Figure 12.20 Alpha-expansion graph setup. Each pixel node (a,b,c,d,e) is connected to the source and the sink by edges with costs $U_\bullet(\overline{\alpha})$ and $U_\bullet(\alpha)$, respectively. In the minimum cut, exactly one of these links will be cut. The nodes and vertices describing the relationship between neighboring pixels depend on their current labels, which may be $\alpha-\alpha$ as for pixels a and b, $\alpha-\beta$ as for pixels b and c, $\beta-\beta$ as for pixels c and d or $\beta-\gamma$ as for pixels d and e. For the last case, an auxiliary node k must be added to the graph.

(see Prince, 2011)

denoising example

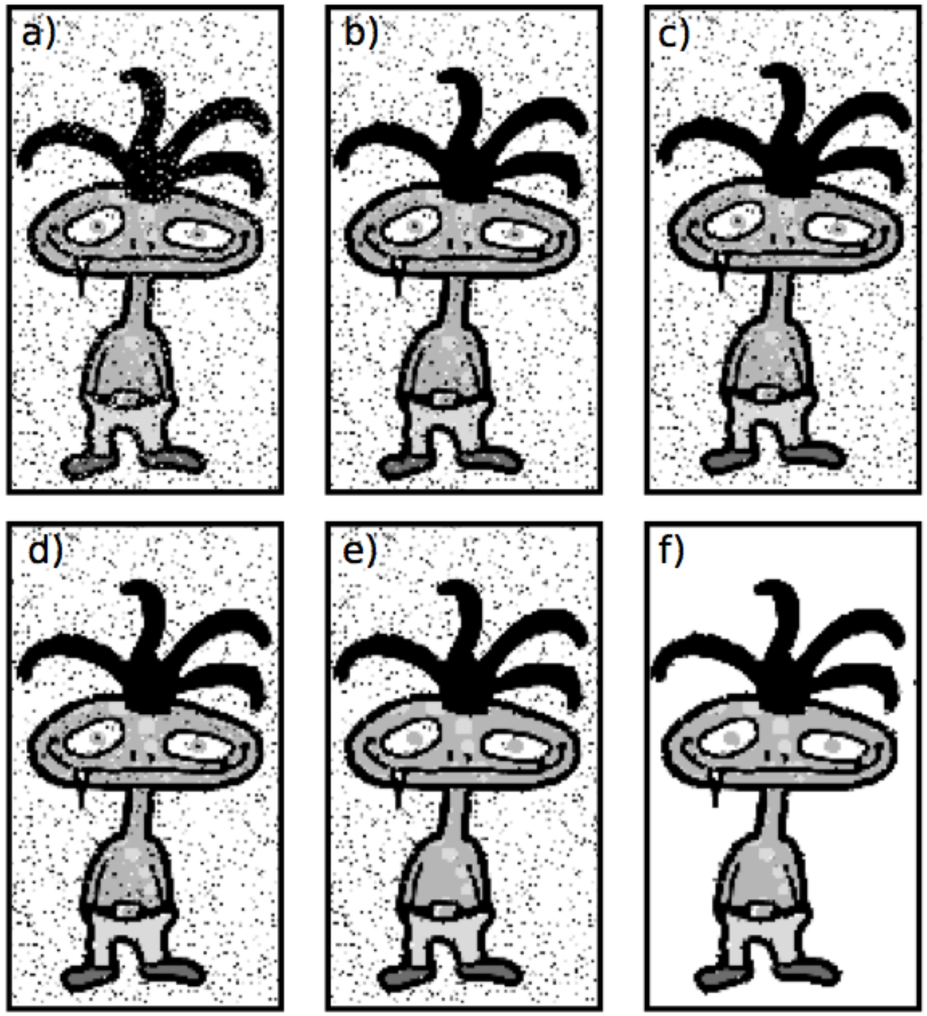
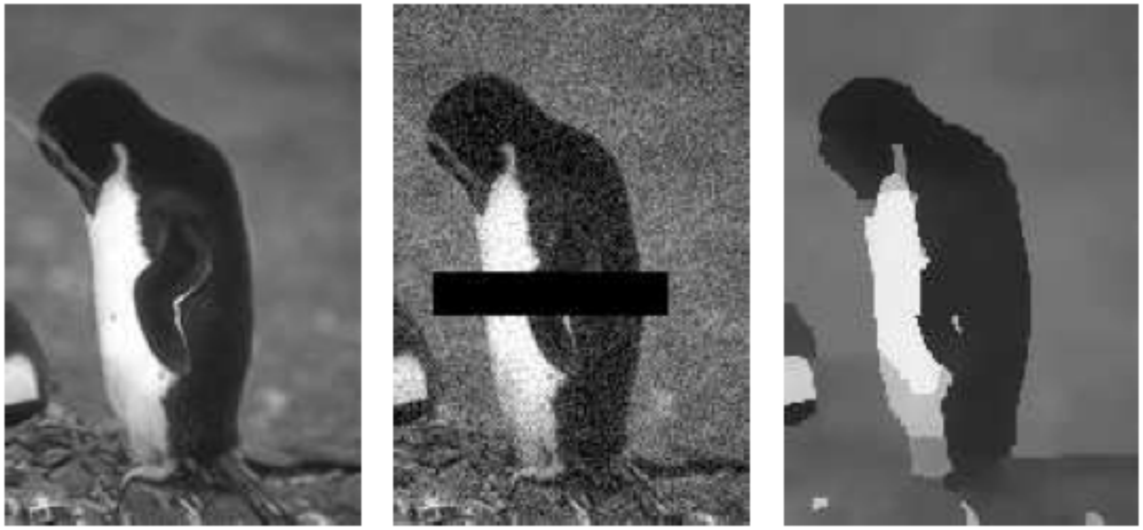


Figure 12.22 Alpha-expansion algorithm for denoising task. a) Observed noisy image. b) Label 1 (black) is expanded, removing noise from the hair. c-f) Subsequent iterations in which the labels corresponding to the boots, trousers, skin and background are expanded respectively.

denoising example

Left: Input with additive Gaussian noise, $\sigma = 10$. So the unary potentials are quadratic, $U(u_i) = (u_i - v_i)^2$. *Middle:* Expansion moves with robust truncated L1 cost, $V(u_i, u_j) = 80 \min(3, |u_i - u_j|)$. *Right:* $V(u_i, u_j) = 15\,|u_i - u_j|$. *[Boykov et al., 2001]*

denoising example



left: Original image. *middle:* Noise plus missing data. *right:* 256 labels, data log likelihood $D(u_i) = (u_i - v_i)^2$, Potts interaction potential.

stereo matching example

Image

Ground truth

Swaps

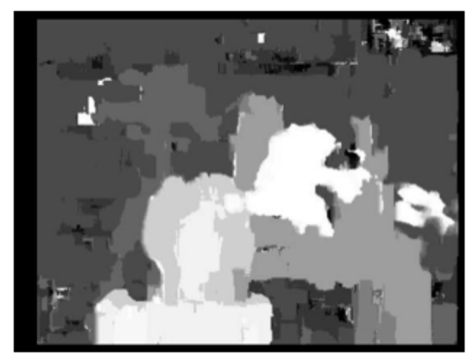
Alpha Expansion

$\mathcal{L} = \{0, 1, ..., 14\}$. Data log likelihood was a truncated quadratic, $D(d) = \min((I_1(x) - I_2(x-d))^2, 20)$. Potts interaction potentials

$$V_{ij}(d_i, d_j) = \begin{cases} 2K & \text{for } |I(x_i) - I(x_j)| \leq 5 \\ K & \text{for } |I(x_i) - I(x_j)| > 5 \end{cases} \quad \text{if } d_i \neq d_j$$

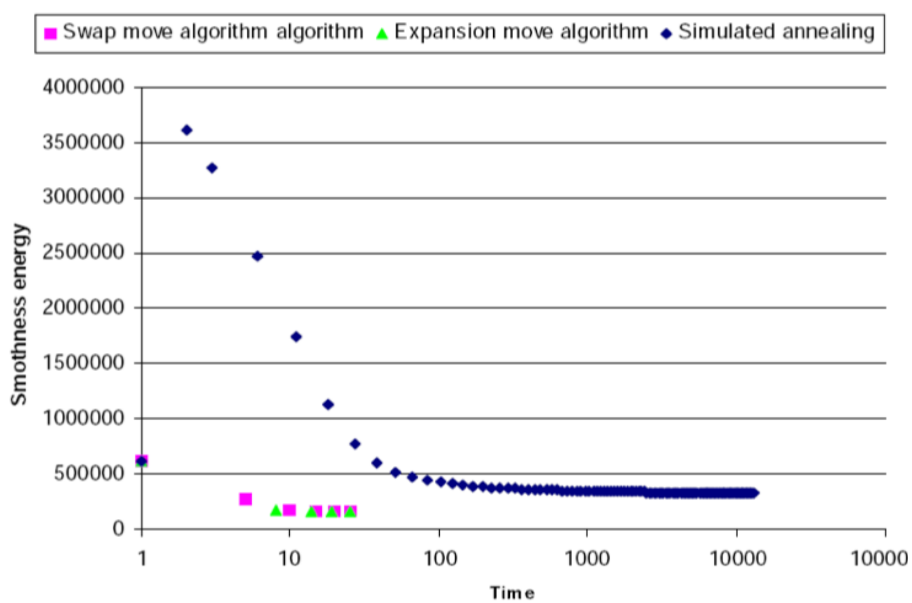
(see *[Boykov et al., 2001]*)

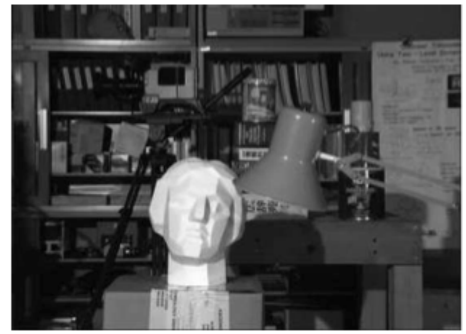
$V_{ij}(d_i, d_j) = 0$ if $d_i = d_j$



Cross-Correlation

stereo matching example



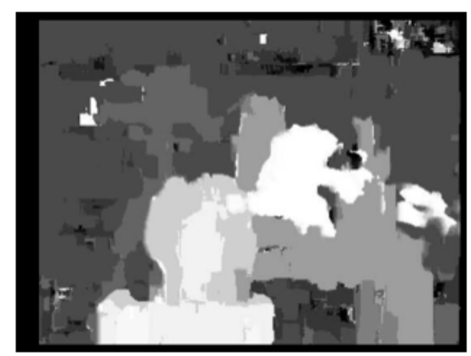


Image

Ground truth

Swaps

Alpha Expansion



Cross-Correlation

Topic 16:

Markov Random Fields (MRFs)

- MRFs & energy minimization
- continuous MRFs
  - quadratic potentials (Gaussian MRFs)
  - robust/non-convex potentials (robust regularization)
- discrete MRFs
  - binary MRFs: Ising model & graph cuts
  - multi-valued MRFs: Potts model & expansion moves
- applications: grabcut, texture synthesis, PEARL

references

- S. D. Prince, Computer vision: models, learning and inference, Cambridge University Press, 2012, www.computervisionmodels.com
- V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts?, IEEE PAMI, v. 26, 2004
- Y. Boykov, O. Veksler, R. Zabih, Fast Approximate Energy Mininization via Graph Cuts, IEEE PAMI, v. 23, 2001
- C. Rother, V. Kolmogorov, A. Blake, "GrabCut" – Interactive Foreground Extraction using Iterated Graph Cuts, ACM SIGGRAPH 2004
- V. Kwatra, A. Schodl, I. Essa, G. Turk, A. Bobick, Graphcut Textures: Image and Video Synthesis Using Graph Cuts, ACM SIGGRAPH 2003
- H. Isack, Y. Boykov, Energy-Based Geometric Multi-Model Fitting, IJCV, v.97, 2012

grabcut (Rother, Kolmogorov, Blake, 2004)

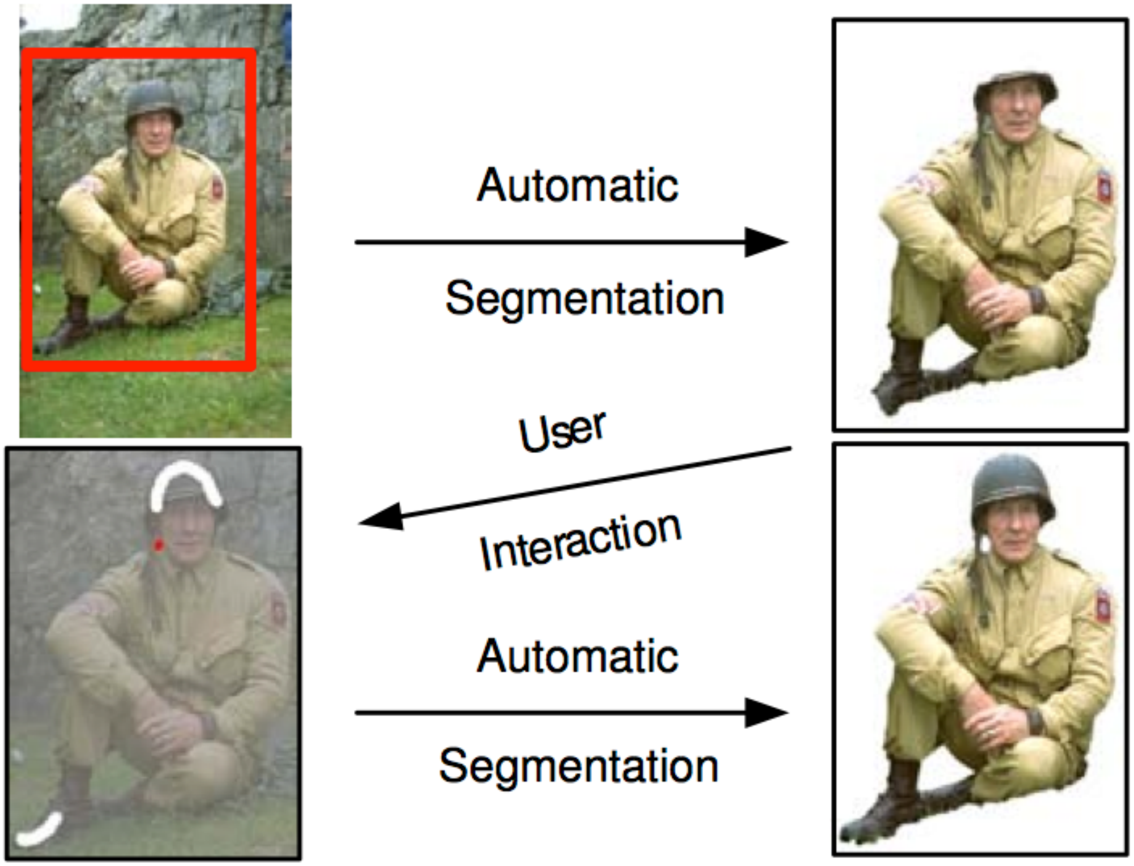
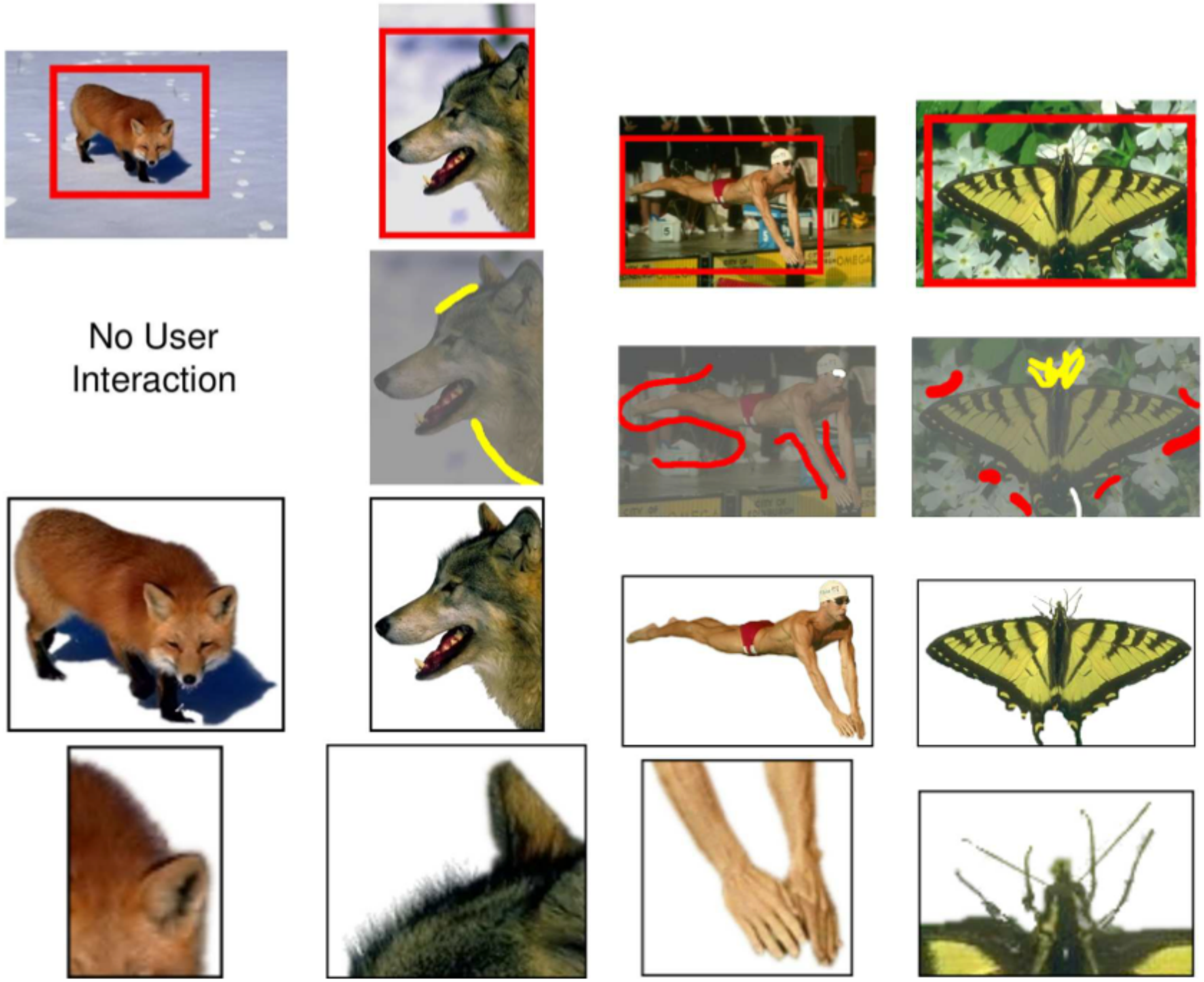
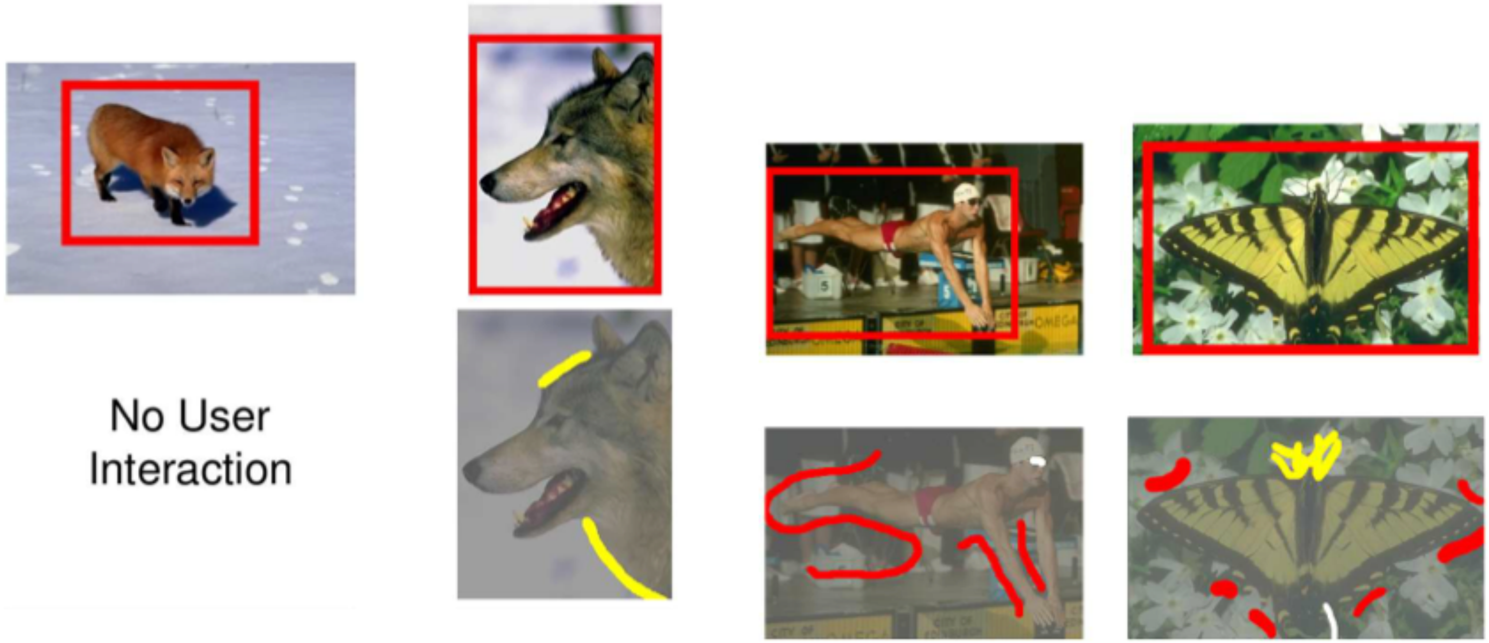


Figure 5: **User editing.** After the initial user interaction and segmentation (top row), further user edits (fig. 3) are necessary. Marking roughly with a foreground brush (white) and a background brush (red) is sufficient to obtain the desired result (bottom row).

grabcut (Rother, Kolmogorov, Blake, 2004)



grabcut: basic algorithm



1. User selects a bounding box. Pixels on box are are taken to be background, while those in the interior are taken to be foreground.
2. Gaussian mixture models (GMM) learned for foreground and background colors
3. Min-cut segmentation. Unary potentials given by GMM negative log likelihood. Interaction potentials much like the Ising model.
4. User then interacts, where necessary, by *painting* foreground (yellow) and/or background (red) pixels. Then return to step 2.

graphcut textures (Kwatra et al, 2003)

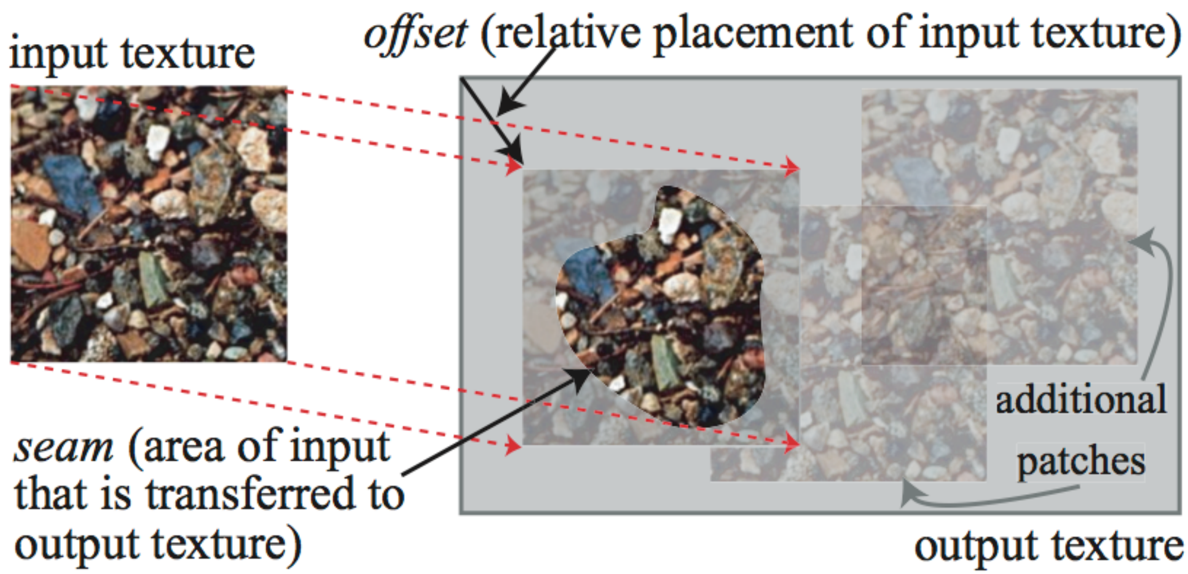


Figure 1: Image texture synthesis by placing small patches at various offsets followed by the computation of a seam that enforces visual smoothness between the existing pixels and the newly placed patch.

graphcut textures (Kwatra et al, 2003)

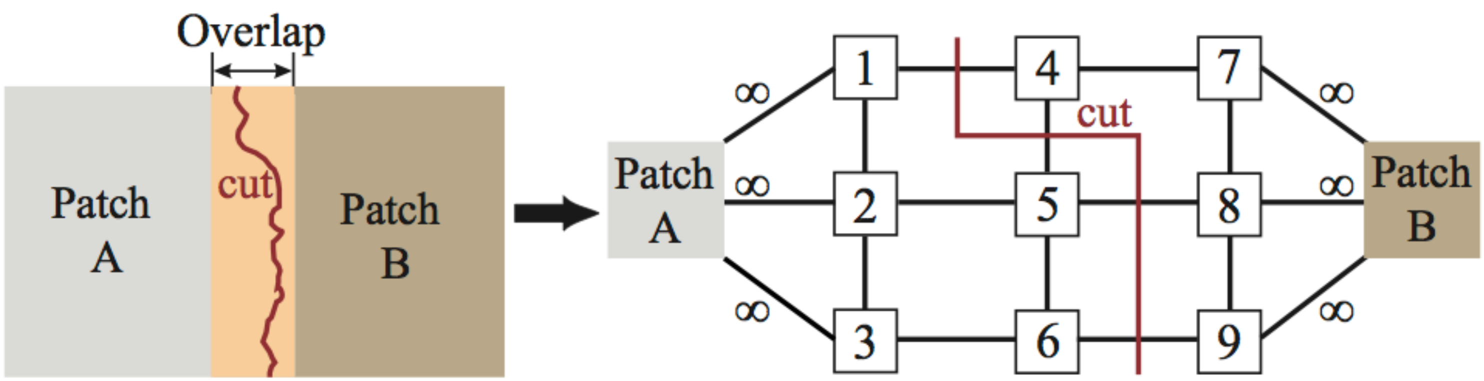


Figure 2: (Left) Schematic showing the overlapping region between two patches. (Right) Graph formulation of the seam finding problem, with the red line showing the minimum cost cut.

graphcut textures (Kwatra et al, 2003)

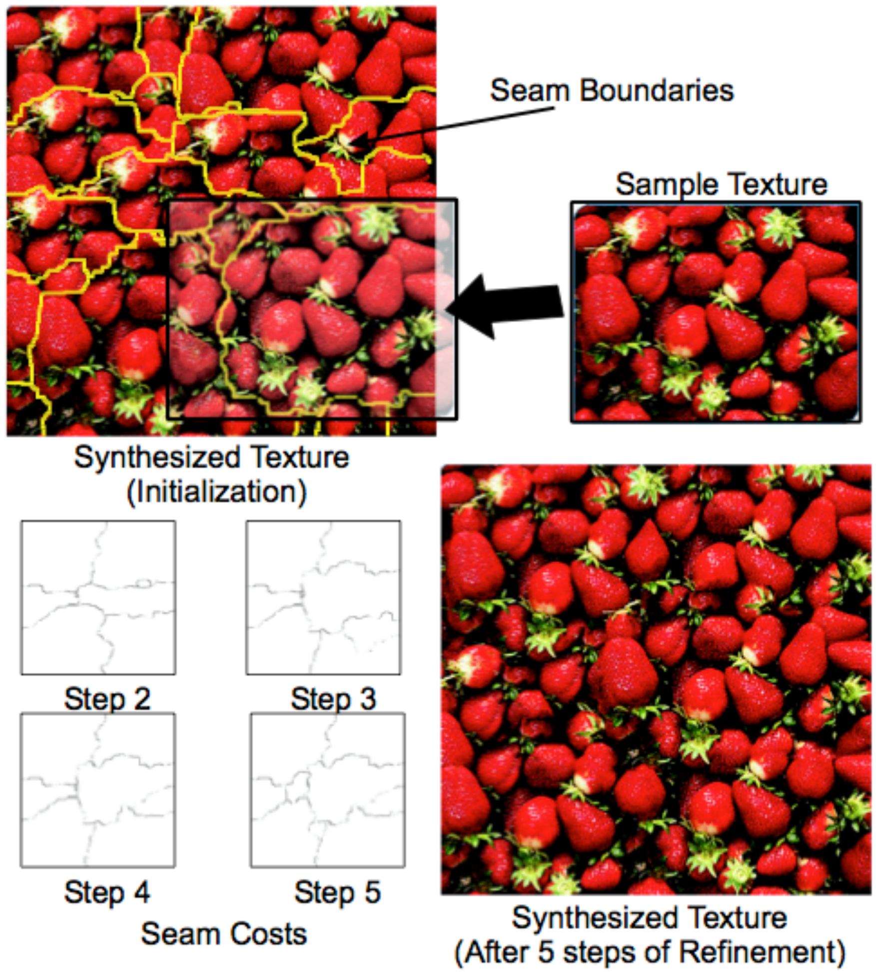


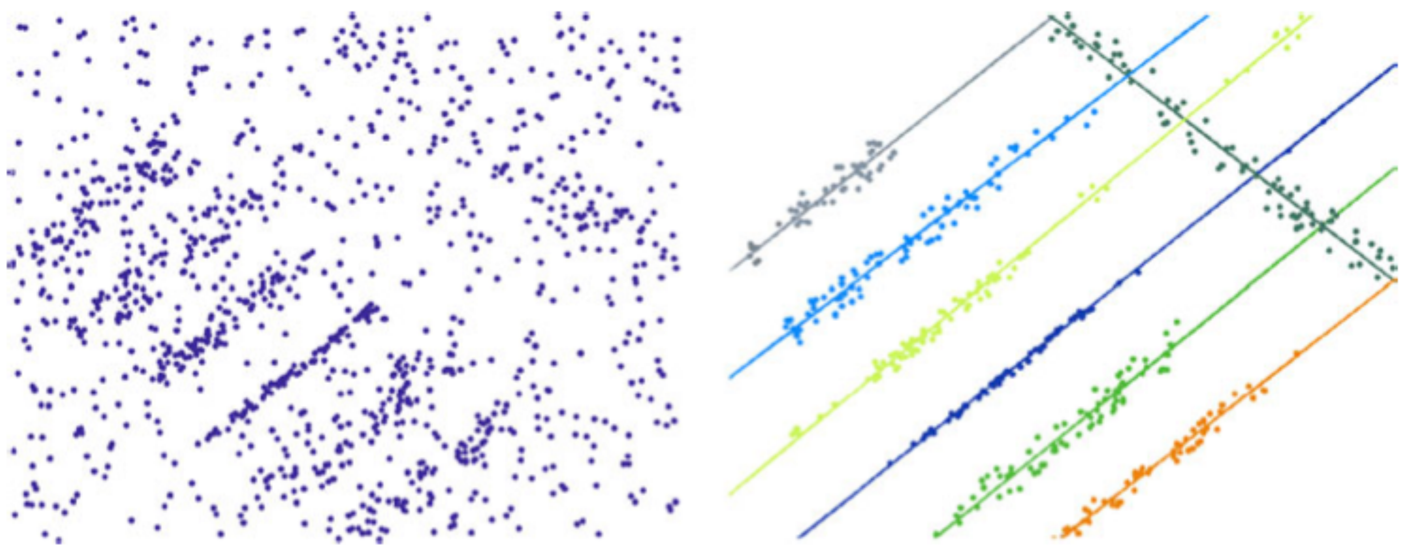
Figure 5: This figure illustrates the process of synthesizing a larger texture from an example input texture. Once the texture is initialized, we find new patch locations appropriately so as to refine the texture. Note the irregular patches and seams. Seam error measures that are used to guide the patch selection process are shown. This process is also shown in the video.

graphcut textures (Kwatra et al, 2003)

Figure 11: Examples of interactive blending of two source images. Shown are HUT and MOUNTAIN©Erskine Wood in the top row, and RAFT and RIVER©Tim Seaver in the bottom row. The results are shown in the third column. In the last column, the computed seams are also rendered on top of the results.

PEARL: incorporating “label costs”

$$E(f) = \overbrace{\sum_{p \in \mathcal{P}} D_p(f_p)}^{\text{data cost}} + \overbrace{\sum_{pq \in \mathcal{N}} V_{pq}(f_p, f_q)}^{\text{smooth cost}} + \overbrace{\sum_{L \subseteq \mathcal{L}} h_L \cdot \delta_L(f)}^{\text{label cost}} \qquad (\star)$$

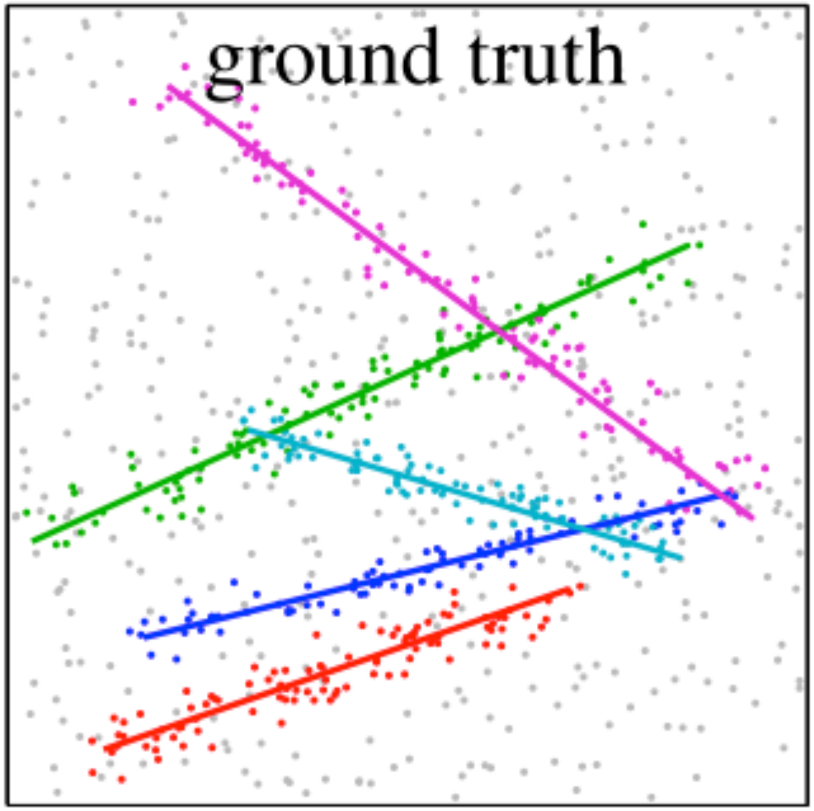


(a) models with different noise levels (b) PEARL result based on (6)

PEARL: steps

Algorithm 3: PEARL Algorithm (Isack and Boykov 2011)

1 **propose** initial models $\mathcal{L}_0$ (e.g. randomly sample data points)
2 run α-**expansion** to compute optimal labeling f w.r.t. $\mathcal{L}_t$
3 **re-estimate** model parameters to get $\mathcal{L}_{t+1}$; $t := t+1$; goto **2**



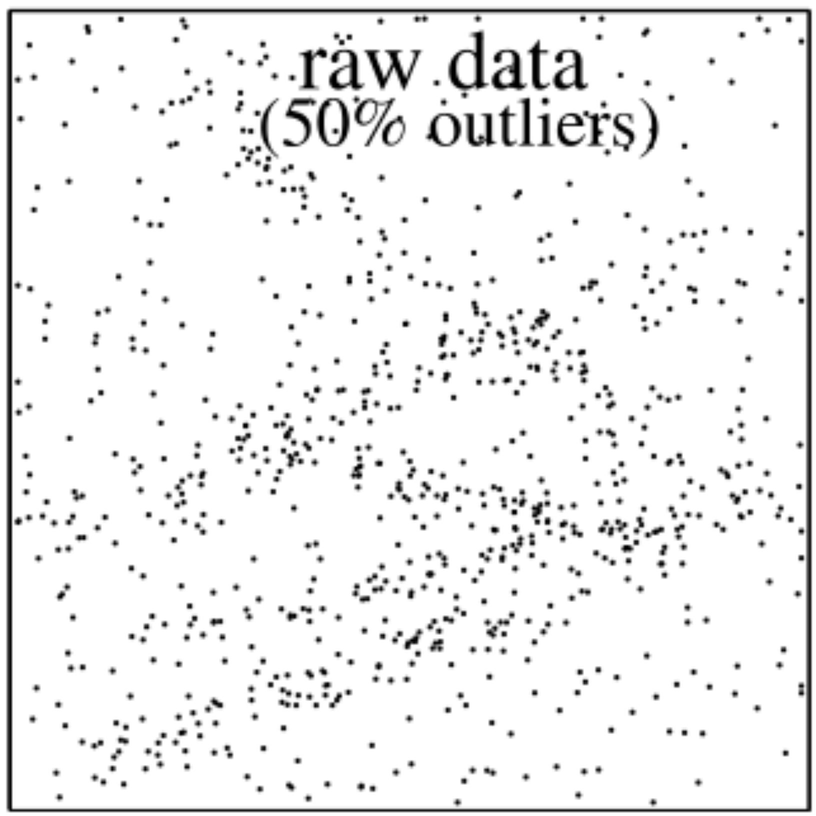
PEARL: steps

Algorithm 3: PEARL Algorithm (Isack and Boykov 2011)

1 **propose** initial models $\mathcal{L}_0$ (e.g. randomly sample data points)

2 run α-**expansion** to compute optimal labeling f w.r.t. $\mathcal{L}_t$

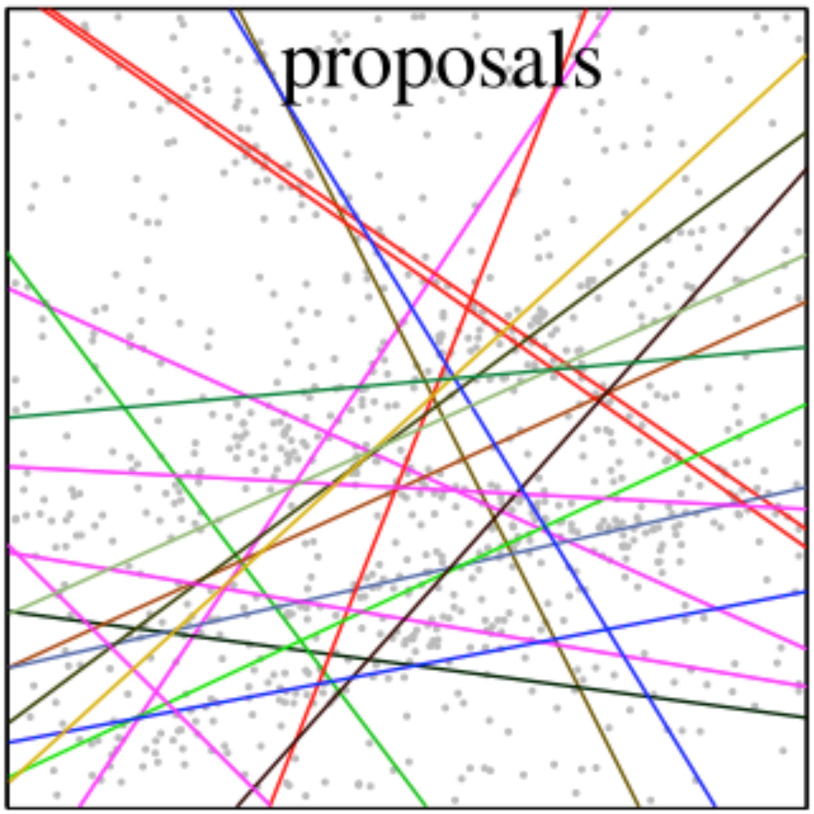
3 **re-estimate** model parameters to get $\mathcal{L}_{t+1}$; $t := t+1$; goto **2**



PEARL: steps

Algorithm 3: PEARL Algorithm (Isack and Boykov 2011)

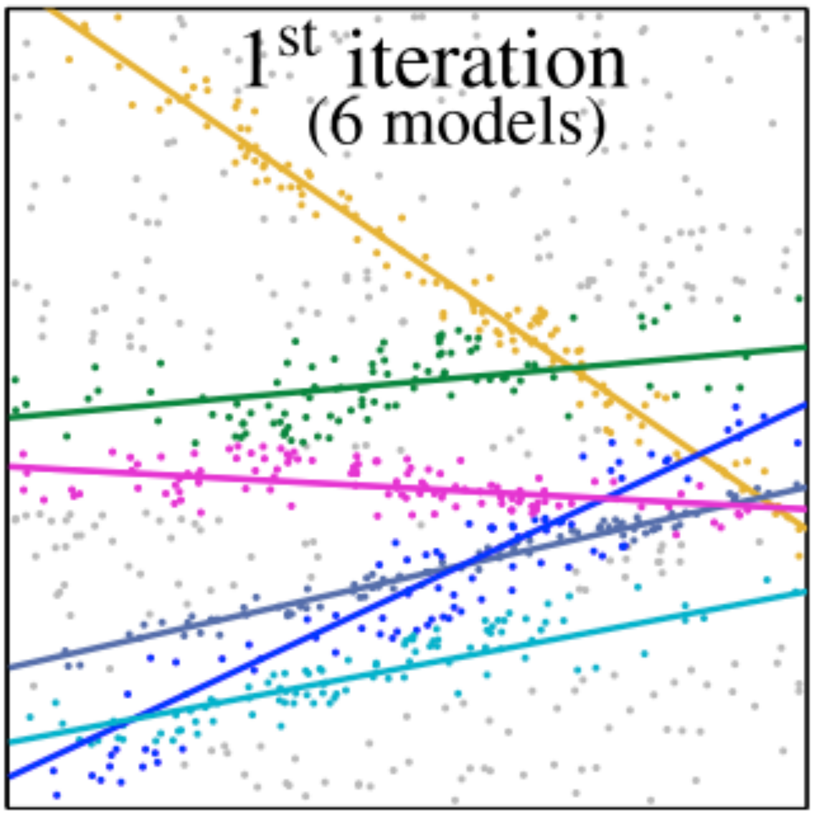
1 **propose** initial models $\mathcal{L}_0$ (e.g. randomly sample data points)
2 run α-**expansion** to compute optimal labeling f w.r.t. $\mathcal{L}_t$
3 **re-estimate** model parameters to get $\mathcal{L}_{t+1}$; $t := t+1$; goto **2**



PEARL: steps

Algorithm 3: PEARL Algorithm (Isack and Boykov 2011)

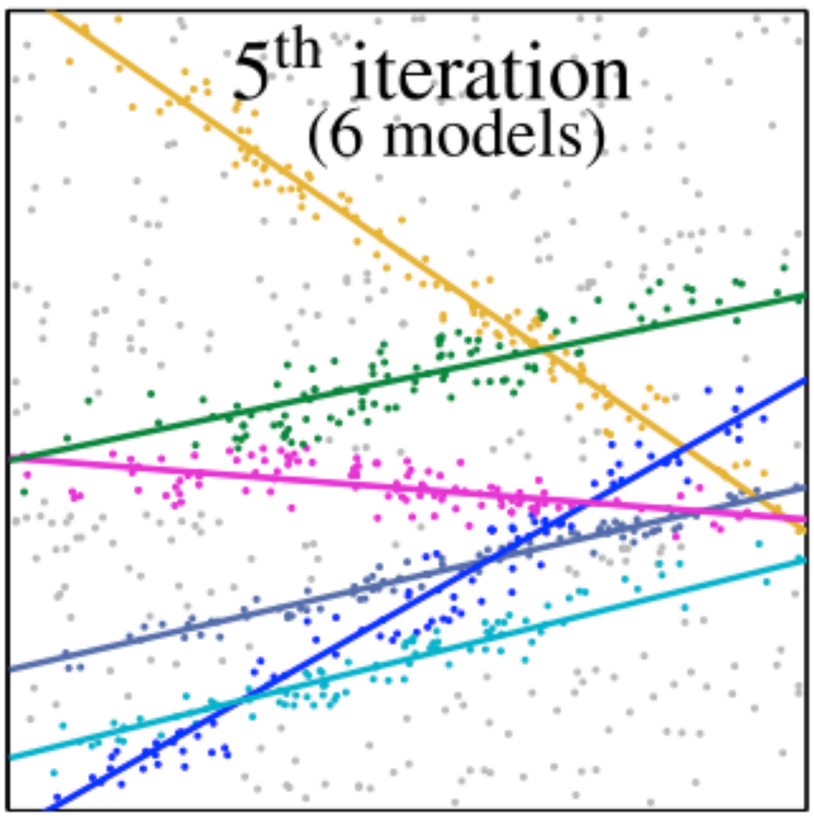
1 **propose** initial models $\mathcal{L}_0$ (e.g. randomly sample data points)
2 run α-**expansion** to compute optimal labeling f w.r.t. $\mathcal{L}_t$
3 **re-estimate** model parameters to get $\mathcal{L}_{t+1}$; $t := t+1$; goto **2**



PEARL: steps

Algorithm 3: PEARL Algorithm (Isack and Boykov 2011)

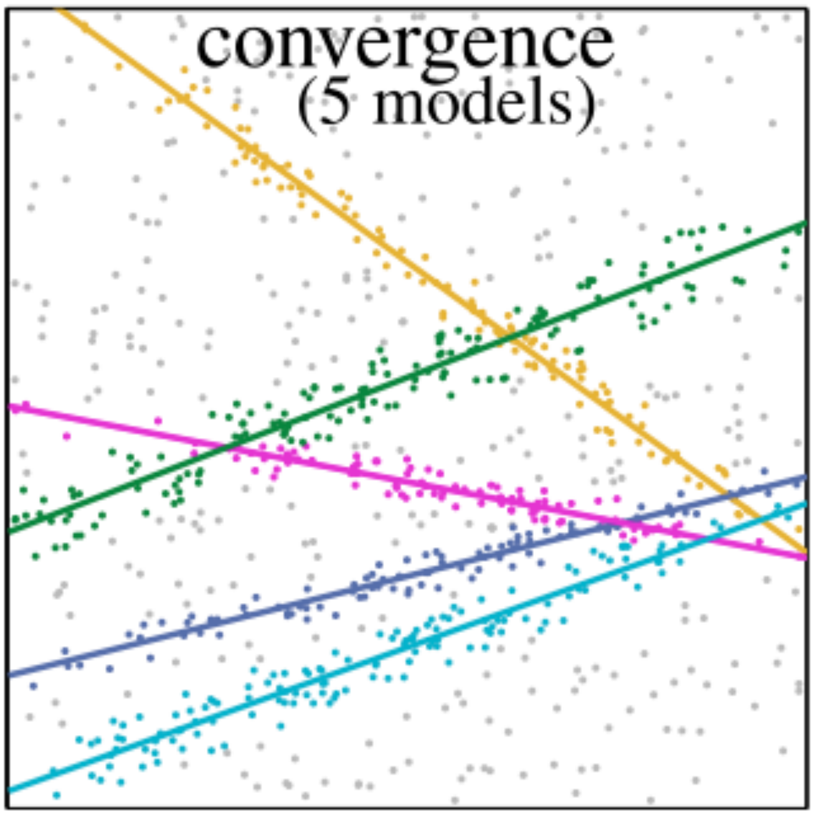
1 **propose** initial models $\mathcal{L}_0$ (e.g. randomly sample data points)
2 run α-**expansion** to compute optimal labeling f w.r.t. $\mathcal{L}_t$
3 **re-estimate** model parameters to get $\mathcal{L}_{t+1}$; $t := t+1$; goto 2



PEARL: steps

Algorithm 3: PEARL Algorithm (Isack and Boykov 2011)

1 **propose** initial models $\mathcal{L}_0$ (e.g. randomly sample data points)
2 run α-**expansion** to compute optimal labeling f w.r.t. $\mathcal{L}_t$
3 **re-estimate** model parameters to get $\mathcal{L}_{t+1}$; $t := t+1$; goto **2**



PEARL: steps

Algorithm 3: PEARL Algorithm (Isack and Boykov 2011)

1 **propose** initial models $\mathcal{L}_0$ (e.g. randomly sample data points)
2 run α-**expansion** to compute optimal labeling f w.r.t. $\mathcal{L}_t$
3 **re-estimate** model parameters to get $\mathcal{L}_{t+1}$; $t := t+1$; goto **2**

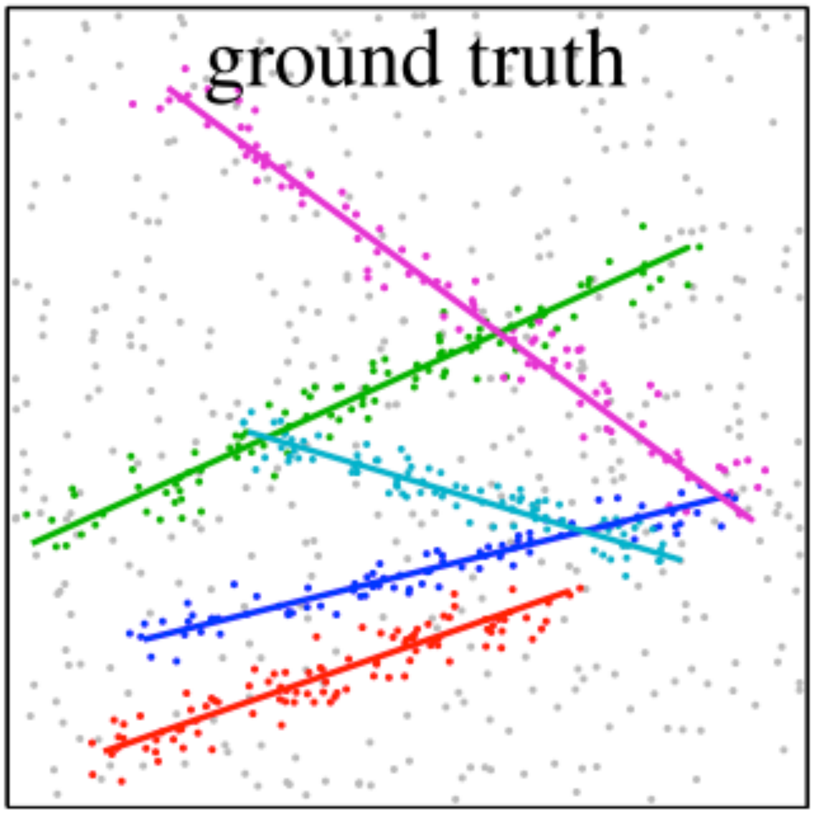


image segmentation (Delong & Boykov, 2012)

$$E(f, M) = \sum_{l \in \mathcal{L}} \underbrace{\sum_{p : f_p = l} - \log P(I_p \mid M_l)}_{\text{segment appearance}}$$

$$+ \lambda \underbrace{\sum_{pq \in \mathcal{N}} [f_p \neq f_q]}_{\text{segments' boundaries}} + \underbrace{\sum_{l \in \mathcal{L}} h_l \cdot \delta_l(f)}_{\text{segments' labels}} \qquad (37)$$