Markov Random Fields

Goal: Introduce basic properties of Markov Random Field (MRF) models and related energy minimization problems in image analysis.

Outline:

1. MRFs and Energy Minimization
2. Quadratic Potentials (Gaussian MRFs)
3. Non-Convex Problems (Robust Regularization)
4. Discrete MRFs (Ising and Potts Models)
5. Gibbs Sampling, ICM, Simulated Annealing
6. Min-Cut/Max-Flow, Expansion Moves

Additional Readings:

- S. Prince, *Computer Vision: Models, Learning and Inference*. See Chapter 12, Models for Grids. `http://computervisionmodels.blogspot.com/`.

Energy Minimization and MRFs

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:

$$E(u) \;=\; E_{data}(u) \,+\, E_{smoothness}(u)\;.$$

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.

Denoising: Given a noisy image $\hat{I}(x, y)$, where some measurements may be missing, recover the original image $I(x, y)$, which is typically assumed to be smooth.

Stereo Disparity: Given two images of a scene, find the binocular disparity at each pixel, $d(x, y)$. The disparities are expected to be piecewise smooth since most surfaces are smooth.

Surface Reconstruction: Given a sparse set of depth measurements and/or normals, recover a smooth surface $z(x, y)$ consistent with the measurements.

Segmentation: Assign labels to pixels in an image, e.g., to segment foreground from background.

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

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

  $\mathcal{N}_i$ is often called the Markov blanket of node i.

Bayesian filtering (see the tracking notes) used a special class of MRFs for which the graph was a chain. The joint distribution over the RVs of a first-order Markov chain can be factored into a product of conditional distributions. This permits efficient inference (remember the recursive form of the filtering distribution). Similar properties hold for tree-structured MRFs, but not for graphs with cycles.

The key to MRFs is that, through local connections, information can propagate a long way through the graph. This *communication* is important if we want to express models in which knowing the value of one node tells us something important about the values of other, possibly distant, nodes in the graph.

Markov Random Fields (cont)

The distribution over an MRF (i.e., over RVs $u = (u_1, ..., u_N)$) that satisfies (1) can be expressed as the product of (positive) potential functions defined on maximal cliques of $\mathcal{G}$ *[Hammersley-Clifford Thm]*.

Such distributions are often expressed in terms of an *energy function* E, and clique potentials Ψ_c:

$$p(u) \;=\; \frac{1}{Z}\exp(-E(u,\theta))\;, \quad \text{where } E(u,\theta) \;=\; \sum_{c\in\mathcal{C}} \Psi_c(\bar{u}_c,\theta_c)\;. \quad (2)$$

Here,

- $\mathcal{C}$ is the set of maximal cliques of the graph (i.e., maximal subgraphs of $\mathcal{G}$ that are fully connected),
- The *clique potential* Ψ_c, $c \in \mathcal{C}$, is a non-negative function defined on the RVs in clique $\bar{u}_c$, parameterized by θ_c.
- Z, the *partition function*, ensures the distribution sums to 1:

  $$Z = \sum_{u_1 ... u_N} \prod_{c\in\mathcal{C}} \exp(-\Psi_c(\bar{u}_c,\theta_c))$$

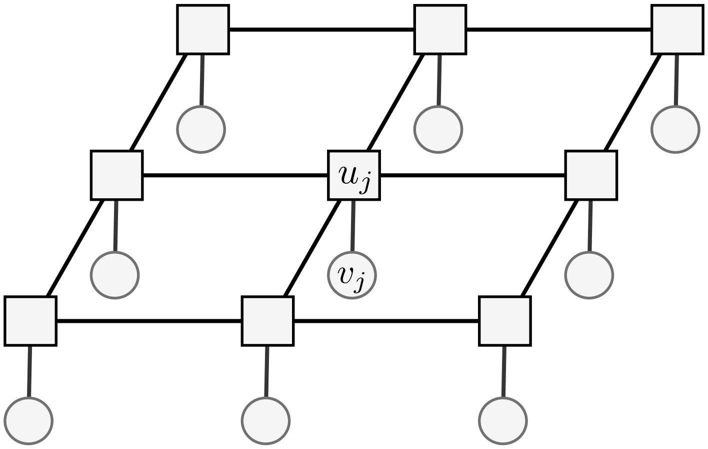
  The partition function is important for learning as it's a function of the parameters $\theta = \{\theta_c\}_{c\in\mathcal{C}}$. But often it's not critical for inference.

Inference with MRFs is challenging as useful factorizations of the joint distribution, like those for chains and trees are not available. For all but a few special cases, MAP estimation is NP-hard.

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.



Accordingly, the energy function is given by

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

- 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).

Quadratic Potentials in 1D

Let v be the sum of a smooth 1D signal u and IID Gaussian noise e:

$$v \;=\; u + e \,, \tag{4}$$

where $u = (u_1, ..., u_N)$, $v = (v_1, ..., v_N)$, and $e = (e_1, ..., e_N)$.

With Gaussian IID noise, the negative log likelihood provides a quadratic *data term*. If we let the *smoothness term* be quadratic as well, then up to a constant, the log posterior is

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

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

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

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 (cont)

We therefore have N linear equations in the N unknowns:

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

Sparse matrix techniques or simple iterative methods can be used to solve (7). In perhaps the simplest scheme, *Jacobi* update iterations are given by:

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

Jacobi iteration converges when the matrix is diagonally dominant (i.e., on each row, the magnitude of the diagonal entry must be larger than the sum of magnitudes of all off-diagonal entries). Other iterative methods convergence more quickly (e.g., *Gauss-Seidel*, *Multigrid*, ...)

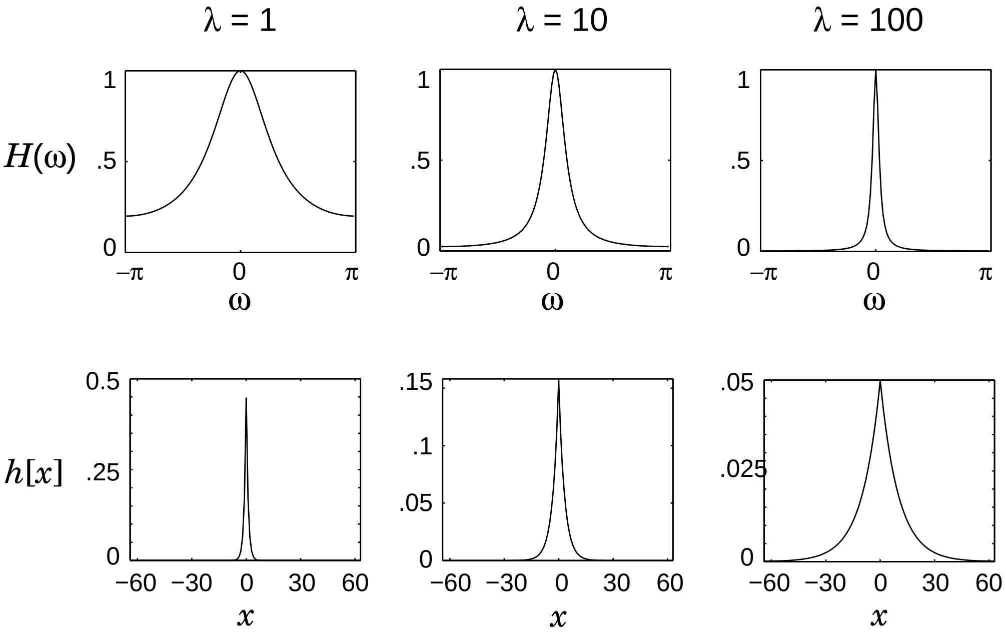
Interpretation as IIR Filter

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

$$u * (\delta + \lambda g) = v \, . \tag{9}$$

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

We can also define the corresponding inverse filter h; i.e., $u = h * v$. 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. For many feedback iterations the support of h is infinite (IIR filters). (But as λ increases the matrix becomes less diagonally dominant.)

Details of the IIR Interpretation

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).

In other words, the matrix equation (7) is approximately equal to a convolution of the measurements $v[x]$ with an appropriate linear filter $h[x]$. If we adopt a different smoothness constraint, i.e., a different high-pass filter $g[x]$, then from (9), the equivalent feedforward linear filter has frequency response:

$$\hat{H}(\omega) = \frac{1}{1+\lambda\,\hat{G}(\omega)}\;, \tag{14}$$

where $\hat{G}(\omega)$ is the Fourier transform of $g[x]$. Increasing the value of λ makes the output smoother, i.e., the feedforward filter becomes more lowpass (see the figure on the previous page).

Another iterative approach is suggested by (14). For small enough values of λ such that $|\lambda\hat{G}(\omega)| < 1$, we can rewrite (14)

$$\hat{H}(\omega) = \frac{1}{1+\lambda\,\hat{G}(\omega)} = 1-\lambda\hat{G}(\omega)+(\lambda\,\hat{G}(\omega))^2-(\lambda\,\hat{G}(\omega))^3+\ldots \tag{15}$$

Since the desired output $\hat{U}(\omega)$ is given by $\hat{H}(\omega)\hat{V}(\omega)$, we can rewrite (9) as

$$u[x] = v[x]-\lambda g[x]*v[x]+\lambda g[x]*(\lambda g[x]*v[x])-\lambda g[x]*(\lambda g[x]*(\lambda g[x]*v[x]))+\ldots \tag{16}$$

That is, $-\lambda g[x]$ is used as a recursive linear filter. The response $u[x]$ can therefore be computed as the limit (as $t\to\infty$) of the following iteration

$$\begin{aligned} u^{(0)}[x] &= v[x], \\ u^{(t+1)}[x] &= v[x]-\lambda g[x]*u^{(t)}[x], \quad \text{for } t\geq 0. \end{aligned}$$

Missing Measurements

The solution is easily extended to missing measurements (i.e., to handle interpolation and smoothing).

Suppose our measurements exist at a subset of positions, denoted P. Then we can write the energy function as

$$E(u) \;=\; \sum_{n \in P} (u_n - v_n)^2 + \lambda \sum_{\text{all}\, n} (u_{n+1} - u_n)^2 \,, \tag{17}$$

At locations n where no measurement exists, the derivative of E w.r.t. u yields the condition

$$-u_{n-1} + 2\, u_n - u_{n+1} \;=\; 0\,. \tag{18}$$

The solution is still a large matrix equation, as in (7). Rows of the matrix with measurements are unchanged. But those for which measurements are missing, have the form $\begin{pmatrix} 0 & \ldots & -1 & 2 & -1 & \ldots & 0 \end{pmatrix}$, with zeros substituted for the corresponding v_n on the right-hand side.

The Jacobi update equation in this case becomes

$$u_n^{(t+1)} = \begin{cases} \frac{1}{1+2\lambda}\,(v_n + \lambda u_{n-1}^{(t)} + \lambda u_{n+1}^{(t)}) & \text{for}\, n \in P\,, \\ \frac{1}{2}\,(u_{n-1}^{(t)} + u_{n+1}^{(t)}) & \text{otherwise} \end{cases} \tag{19}$$

The equations that govern the endpoints can be expressed in an analogous manner.

2D Image Smoothing

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

$$\begin{aligned} E(u) &= \sum_{n,m \in P} (u[n,m] - v[n,m])^2 \\ &\quad + \lambda \sum_{\text{all}\, n,m} (u[n{+}1,m] - u[n,m])^2 + (u[n,m{+}1] - u[n,m])^2 \end{aligned} \quad (20)$$

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

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

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

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

Problem: Linear filters are sensitive to outliers, and will not preserve image edges. They tend to oversmooth images at boundaries.

Robust Potentials

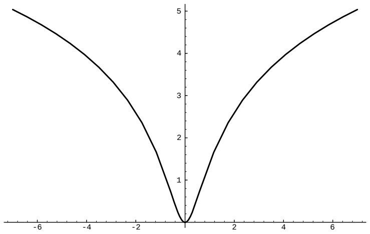
Quadratic potentials are not robust to *outliers* and hence they over-smooth edges. These effects will propagate throughout the graph.

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

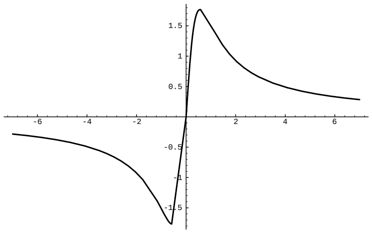
$$E(u) \;=\; \sum_{n=1}^{N} \rho(u_n - v_n,\, \sigma_d) + \lambda \sum_{n=1}^{N-1} \rho(u_{n+1} - u_n,\, \sigma_s)\,, \qquad (22)$$

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

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

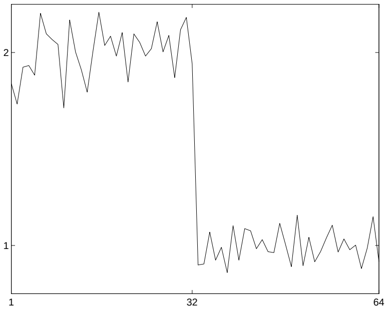


Error function

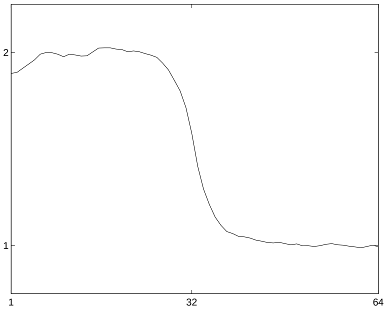


Influence function

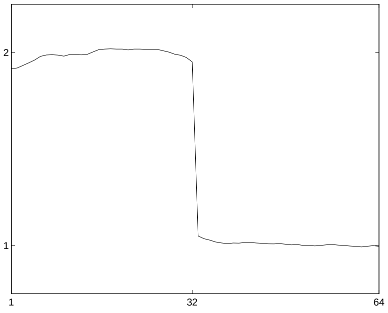
Smoothing a noisy step edge:



Noisy step



LS smoother



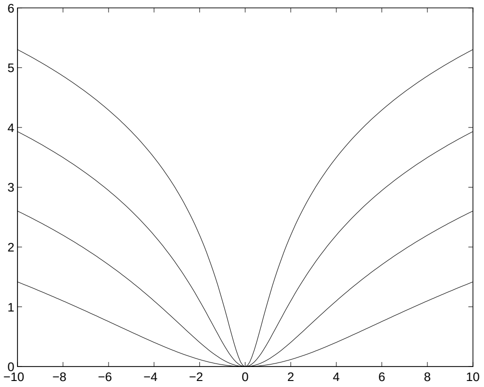
Lorentzian smoother

Unfortunately, the problem is no longer convex. Optimization is tough.

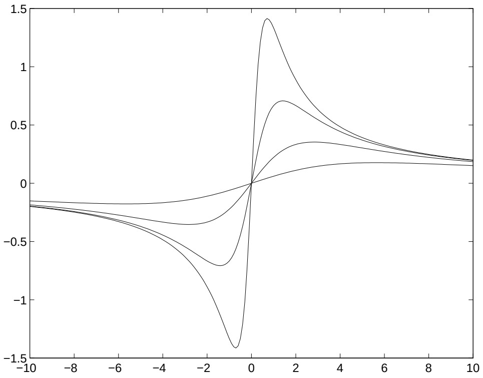
Graduated Non-Convexity

Robust formulations produce nonconvex optimization problems. To help avoid poor local minima, one can choose a robust ρ-function with a scale parameter, and adjust the scale parameter to construct a convex approximation. This approximation is readily minimized. Then successively better approximations of the true objective function are constructed by slowly adjusting the scale parameter back to its original value. This process is called *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.

The graduated non-convexity algorithm begins with the convex (quadratic) approximation so the initial estimate contains no outliers. Outliers are gradually introduced by lowering the value of σ and repeating the minimization. While this approach works well in practice, it is not guaranteed to converge to the global minimum since, as with least squares, the solution to the initial convex approximation may be arbitrarily bad.

Measurements beyond some threshold, τ, can be considered outliers. The point where the influence of outliers first begins to decrease occurs when the second derivative of the ρ-function is zero. For the Lorentzian, the second derivative,

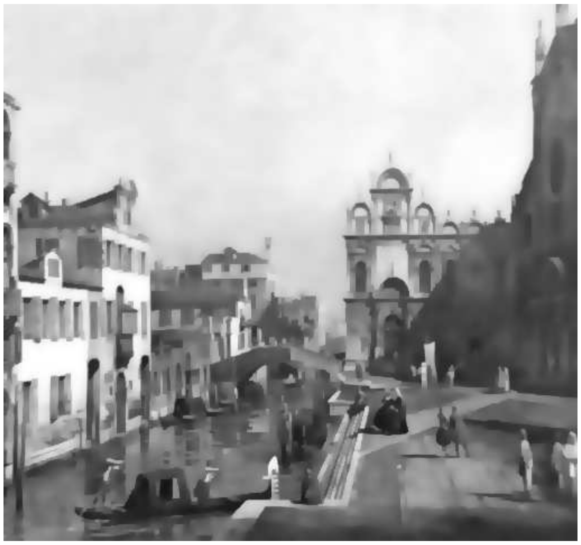
$$\rho''(z) = \frac{2(2\sigma^2 - z^2)}{(2\sigma^2 + z^2)^2},$$

is zero when $z = \pm\sqrt{2}\,\sigma$. If the maximum expected (absolute) residual is τ, then choosing $\sigma = \tau/\sqrt{2}$ will result in a convex optimization problem. A similar treatment applies to other robust ρ-functions. Note that this also gives a simple test of whether or not a particular residual is treated as an outlier. In the case of the Lorentzian, a residual is an outlier if $|z| \geq \sqrt{2}\,\sigma$.

Robust Image Smoothing

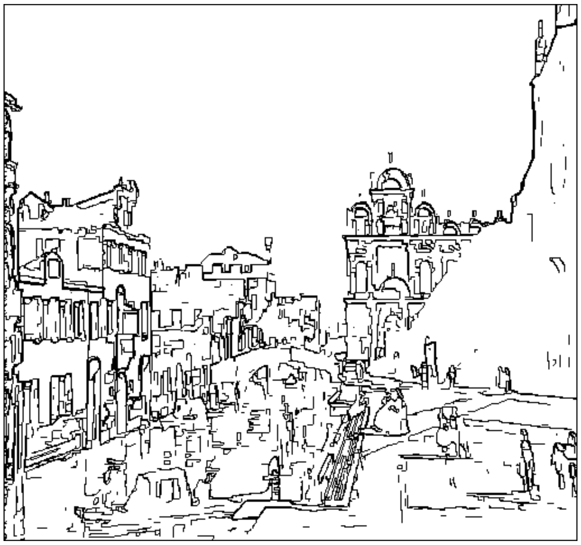
This smoother uses a quadratic data potential, and a Lorentzian smoothness potential to encourage an approximately piecewise constant result:

Original image



Output of robust smoothing

We can use the Lorentzian error function to detect spatial outliers.



Edges

Problem: Computational expense, local minima, and sensitivity to the initial guess.

Discrete Optimization

Quantizing the values that are assigned to the RVs in an MRF allows one to formulate inference using discrete optimization. To this end, let $\mathcal{L}$ be the finite set of labels that can be assigned to the MRF nodes.

Potts Model: $\mathcal{L} = \{1, ..., K\}$ with (robust) interaction potential

$$V(u_i, u_j) \;=\; \beta \,\min(|u_i - u_j|, 1)\;, \tag{24}$$

where $\beta > 0$ is the cost of an edge connecting nodes with different labels. The Potts Model, developed in statistical physics, has been used often for image processing problems.

Inference:

Gibbs Sampling: MCMC method for drawing samples from an MRF. One sweeps through the MRF updating one node at a time. At each step, a node is updated to be a random draw from its conditional distribution (i.e., holding all neighbouring nodes fixed).

Simulated Annealing: Draw samples from $p(u)^{1/T}$, the annealed MRF posterior, as T decreases. As $T \rightarrow 0$, only MAP states have significant probability mass. Provides global MAP estimate, but anneallng must take place in infinitesimal steps, and it uses Gibbs sampling as the inner loop (each time the temperature is reduced).

Iterated Conditional Modes: Greedy form of coordinate descent for approximate MAP estimation. One sweeps through the MRF nodes one at a time. For each node we assign the label that minimizes energy (for which all other nodes are held constant).

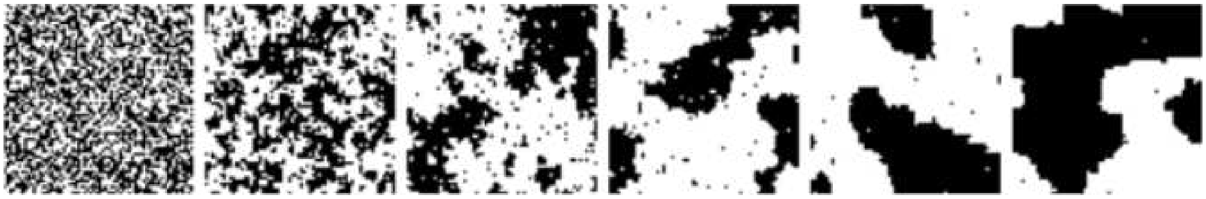
Binary MRFs

In 1989 it was shown that a 2-state version of the Potts model, known as the Ising model, could be solved. That is, the *global* MAP estimate can be found with a polynomial-time algorithm.

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

$$V(u_i, u_j) \; = \; \beta \, |u_i - u_j| \, , \quad \beta > 0 \, . \tag{25}$$

Random samples:



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.,

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

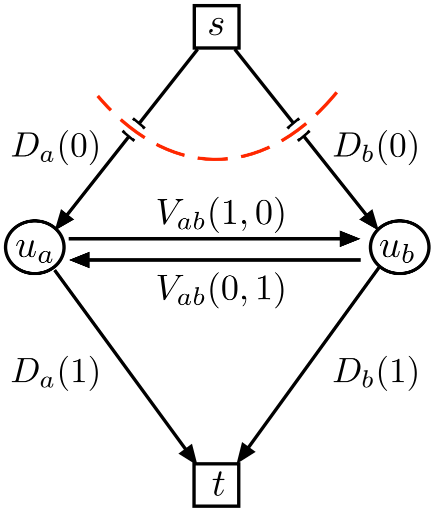
Goal: Find the signal u that minimizes the energy $E(u)$.

MAP Estimation via Max-Flow/Min-Cut

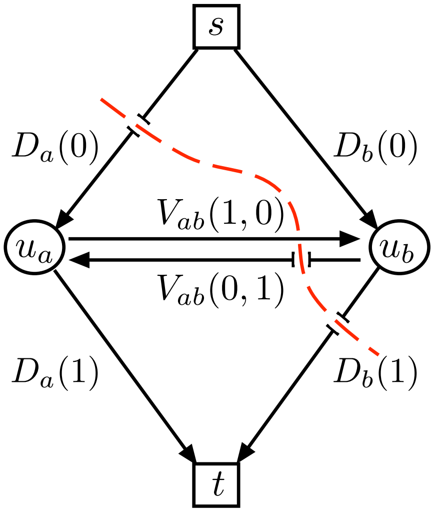
Construct a graph comprising the MRF nodes, a source s, a sink t and edge weights, such that an st-cut (separating t from s; see p. 18) specifies an MRF labeling whose cut *cost* equals the MRF energy.
The min-cut gives a unique, minimum energy labelling in $O(|\mathcal{E}|^2\mathcal{V})$.

Consider a graph with nodes (pixels) a and b, with RVs u_a and u_b:

- Let $D_a(u_a)$, $D_b(u_b)$, and $V_{ab}(u_a, u_b)$ be unary and interaction potentials, such that $V_{ab}(0,0) = V_{ab}(1,1) = 0$ (e.g., an Ising model).
- Given an st-cut, let all nodes with paths from s be labeled 1, and all nodes with paths to t be labeled 0.
- Labels are either equal or different; e.g., $(u_a, u_b) = (0,0)$ or $(0,1)$:



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



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

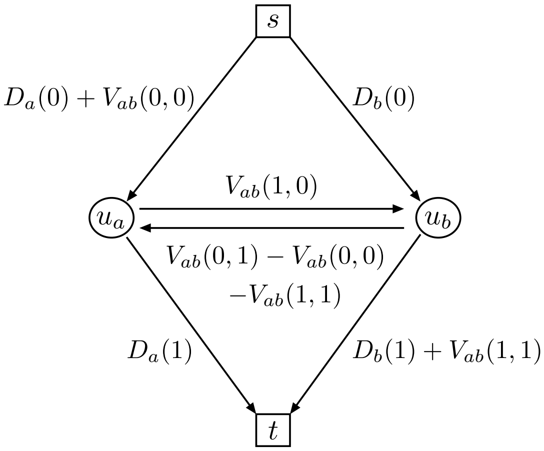
Kolmogorov and Zabih showed that graphs can be constructed for more general interaction potentials satisfying the *sub-modularity* condition:

$$V_{ab}(0,1) + V_{ab}(1,0) \ \geq \ V_{ab}(0,0) + V_{ab}(1,1)\ . \qquad (28)$$

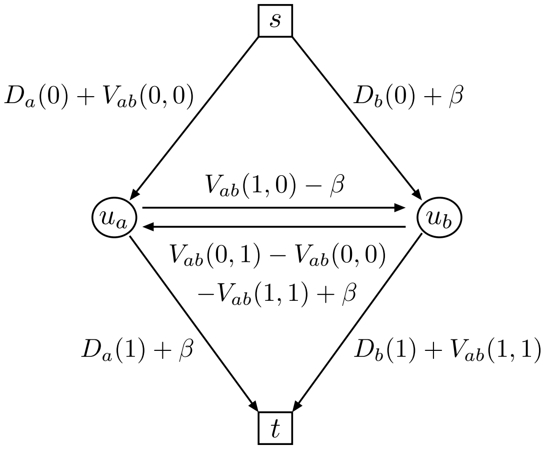
Max-Flow/Min-Cut and Graph Construction

The *Max-Flow* problem is to find the maximum flow from the source s to a sink t in a graph with positive edge weights (capacities). The flow is maximized when on every path from s to t there exists some edge that is at capacity. An st-cut is a graph cut comprising all directed edges from a node in a vertex set A to a node in its complement, $\bar{A}$, such that $s \in A$, $t \in \bar{A}$, and A is connected. The cost of a cut is simply defined as the sum of edge weights for the edges removed by the cut.

The *Min-Cut* problem is to find the st-cut with minimal cost. It is straightforward to show that the min-cut is a subset of the edges that reach capacity for the max-flow, and that the min-cut cost is equal to the max-flow. There are well-known polynomial algorithms for solving the Max-Flow problem. For more details, see `http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem`.



General interaction potentials: The graph construction on the previous page worked for interaction potentials with $V(0,0) = V(1,1) = 0$. One can construct graphs that do not require this constraint. The figure to the right shows a simple 2-node MRF where the cost of every (minimal) plausible cut separating s and t corresponds to a general interaction potential. This generalizes straightforwardly to MRFs with more than 2 nodes.



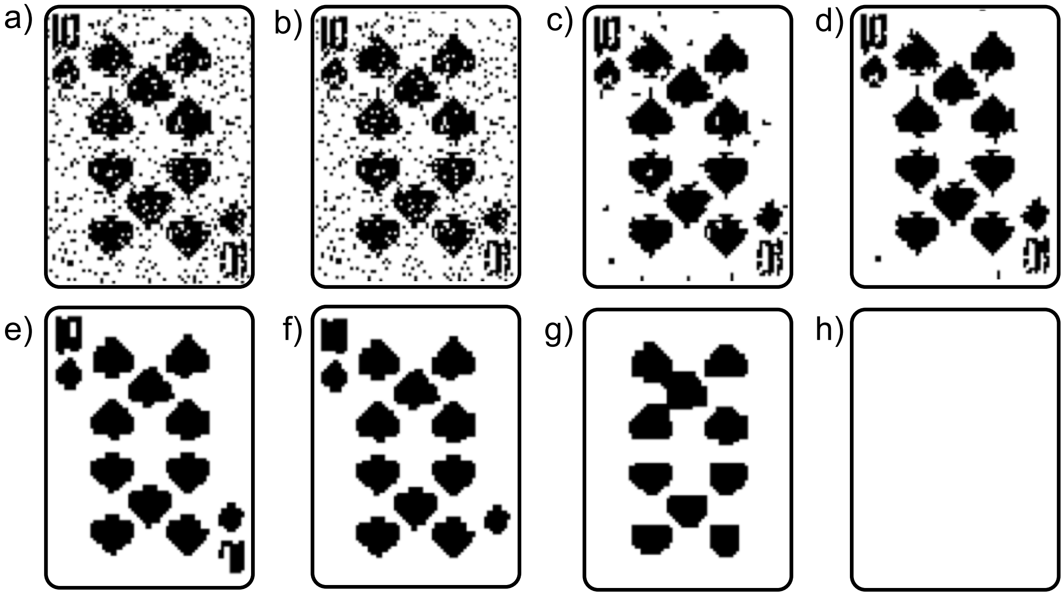
Positive capacities: But we need to avoid negative edge weights on the above graph. To this end we can add a fixed constant β to a selection of the edges so that all plausible (minimal) graph cuts separating s and t have their costs increased by β. While this changes the costs, it leaves the cuts with the minimum cost, hence the MAP labeling, unchanged. To ensure all edge weights are positive, we obtain the following two constraints:

$$\begin{aligned} V_{ab}(0,1) - V_{ab}(0,0) - V_{ab}(1,1) + \beta &\geq 0 \,, \\ V_{ab}(1,0) - \beta &\geq 0 \,. \end{aligned}$$

These yield the following *sub-modularity* condition:

$$V_{ab}(0,1) + V_{ab}(1,0) \;\geq\; V_{ab}(0,0) + V_{ab}(1,1) \,.$$

Binary Image Denoising (cont)



[Prince, 2011, Chapter 12]

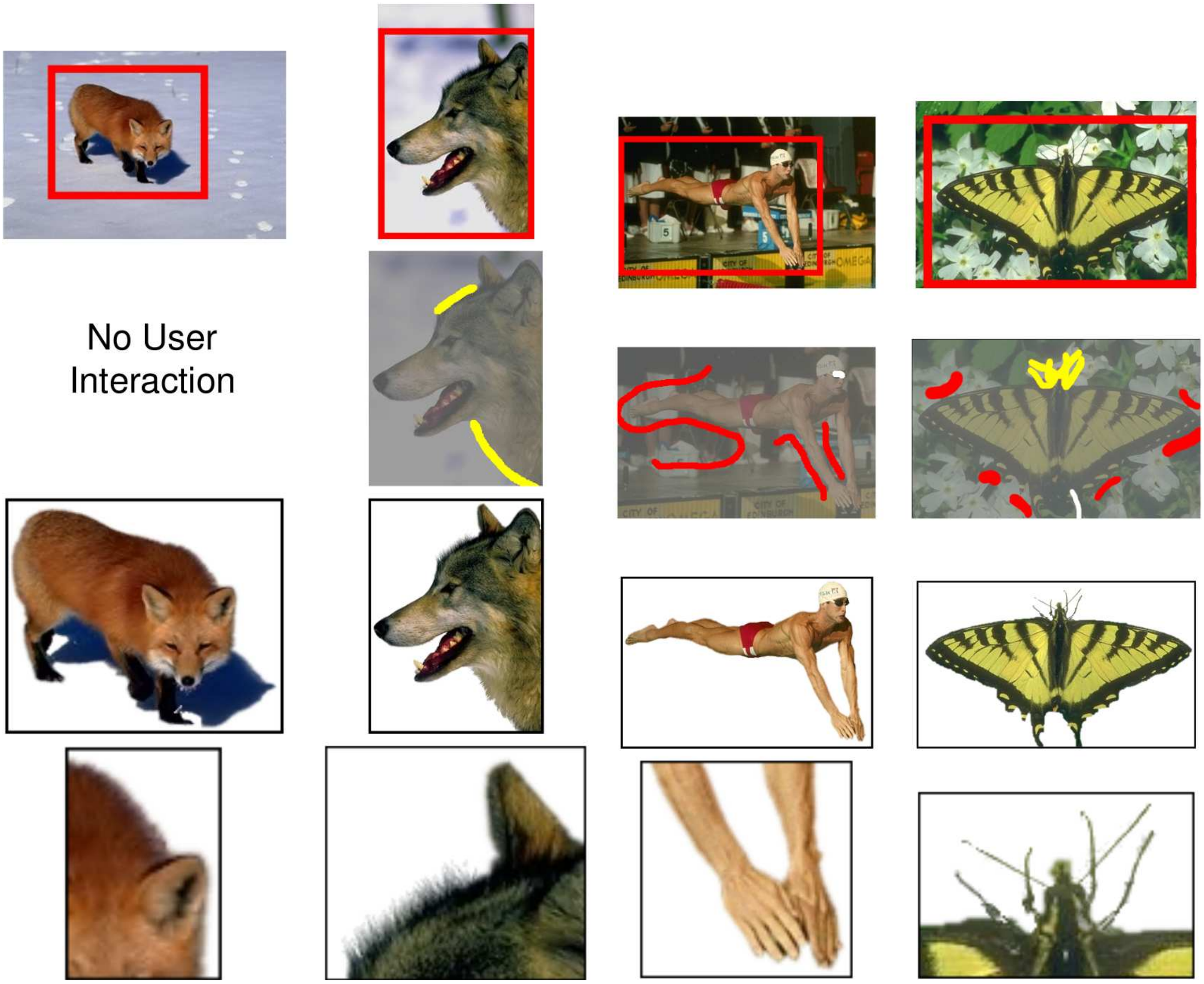
A binary MRF with unary and interaction potentials:

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

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

(a) The original image. (b-h) MAP estimates for increasing β, from on extreme, where the unary terms dominate, to the other, where the interaction potential dominates, yielding a uniform labeling.

Grabcut



[Rother, Kolmogorov, and Blake, 2004]

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.

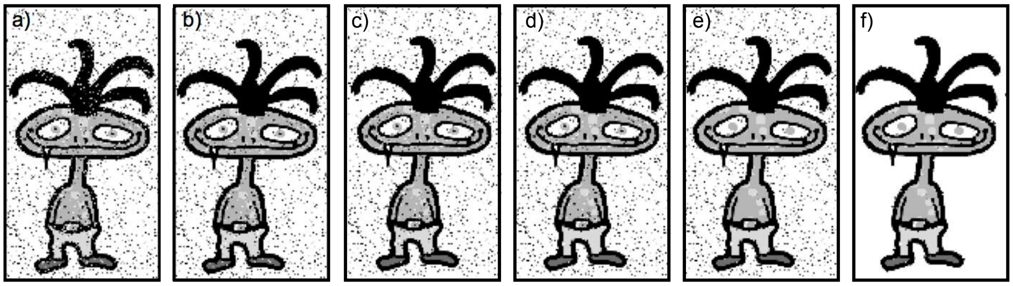
Multi-Valued MRFs

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).

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

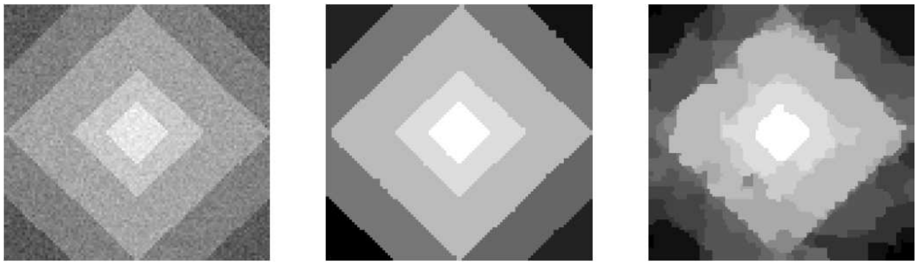


[Prince, 2011, Chapter 12]

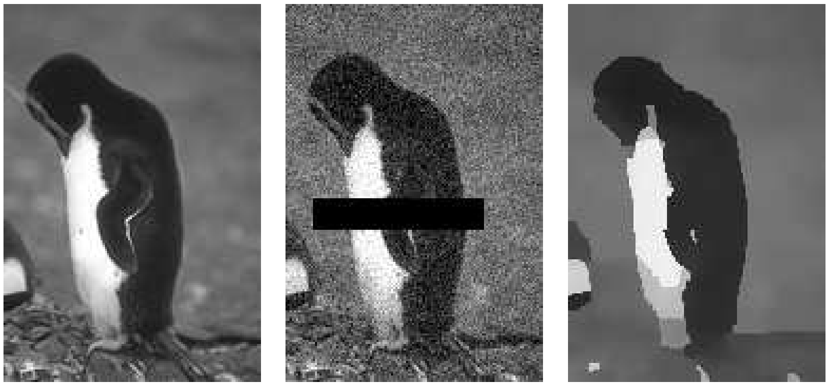
(a) The noisy image. (b) Step 1 cleans up the hair. Step 2 does nothing (the label has no image support). (c-f) Denoising of the boots, trousers, skin, and background.

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

Denoising with Expansion Moves

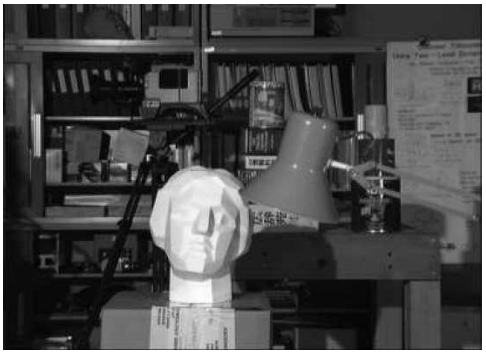


Left: Input with additive Gaussian noise, $\sigma = 10$. So the unary potentials are quadratic, $U(u_i) = (u_i - v_i)^2$. *Middle:* Expansion moves with robust truncated L1 cost, $V(u_i, u_j) = 80 \min(3, |u_i - u_j|)$. *Right:* $V(u_i, u_j) = 15\,|u_i - u_j|$. *[Boykov et al., 2001]*

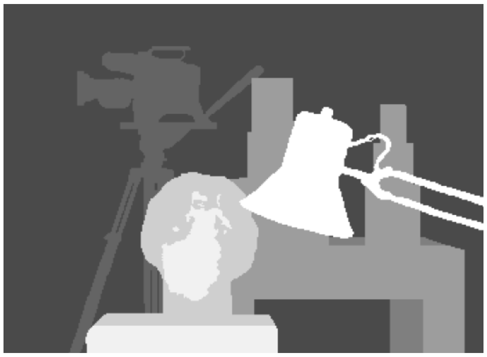


left: Original image. *middle:* Noise plus missing data. *right:* 256 labels, data log likelihood $D(u_i) = (u_i - v_i)^2$, Potts interaction potential.

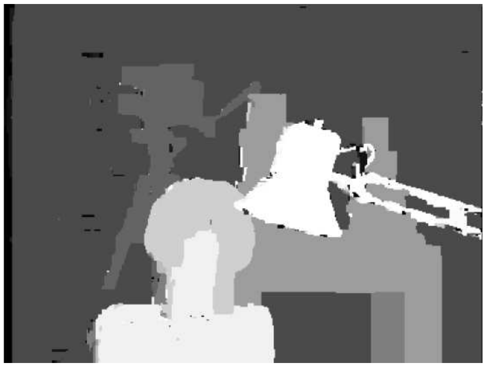
Stereo Matching With Expansion Moves



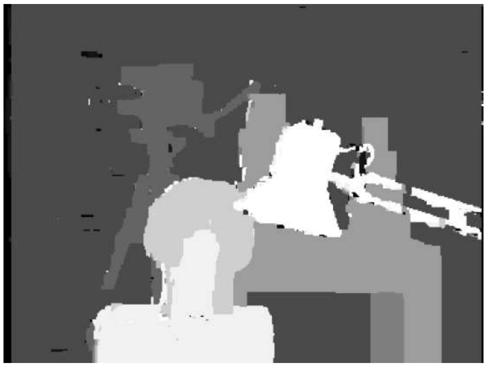
Image



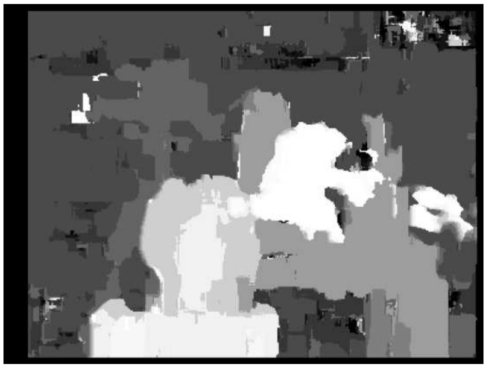
Ground truth



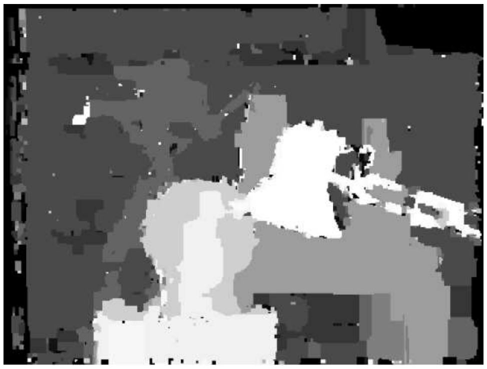
Swaps



Alpha Expansion



Cross-Correlation



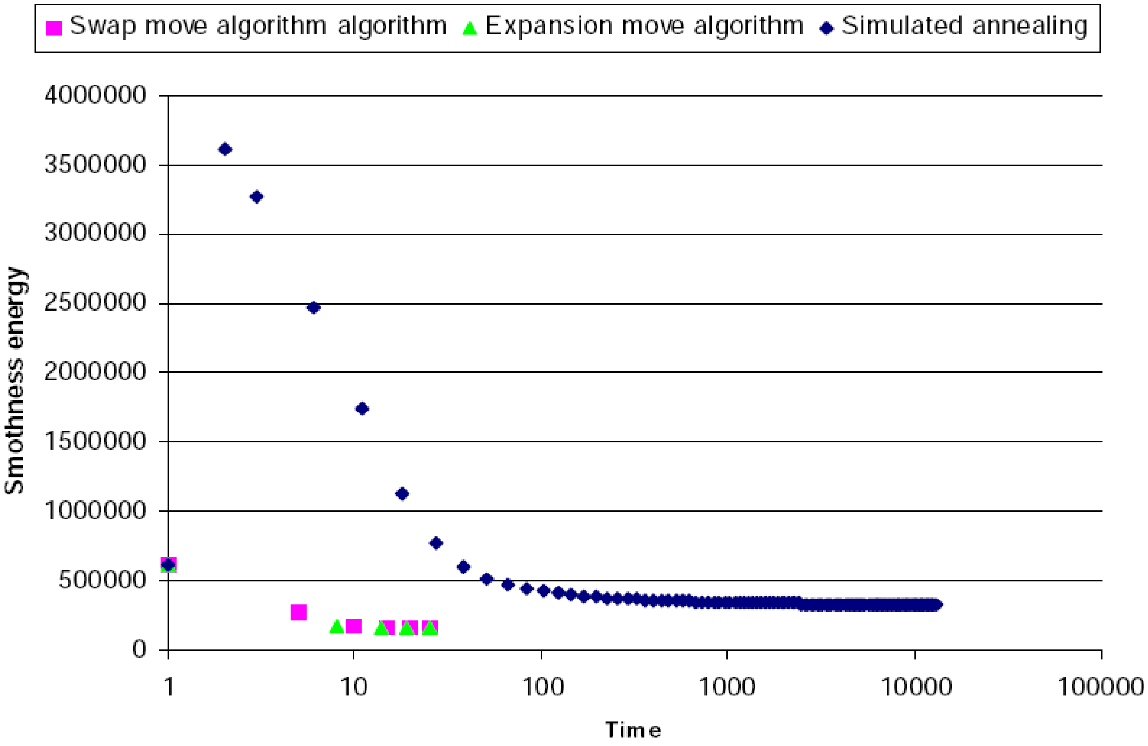
Simulated Annealling

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

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

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

Stereo Matching With Expansion Moves (cont)



Further Readings

Black, M., Sapiro, G., Marimont, D., and Heeger, D. (1998) Robust anisotropic diffusion, IEEE Trans. Image Proc. 7(3):421-432.

Blake, A., Rother, C., and Kohli, P. (2011), *Markov Random Fields for Vision and Image Processing*, MIT Press (to appear)

Boykov, Y. and Kolmogorov, V. (2004) An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans PAMI 26(9):1124-1137.

Boykov, Y., Veksler, O., and Zabih, R. (2001) Fast approximate energy minimization via graph cuts. IEEE Trans. PAMI, 23(11):1222-1239.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. PAMI 6:721–741.

Greig, D., Porteous, B., and Seheult, A. (1989) Exact maximum a posteriori estimation for binar images, J. Roy. Statist. Soc. B 51:271–279.

Kolmogorov, V. and Zabih, R. (2004) What Energy Functions can be Minimized via Graph Cuts? IEEE Trans. PAMI 26:147–159.

Rother C., Kolmogorov, V., and Blake, A. (2004) GrabCut: Interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23:309–314.