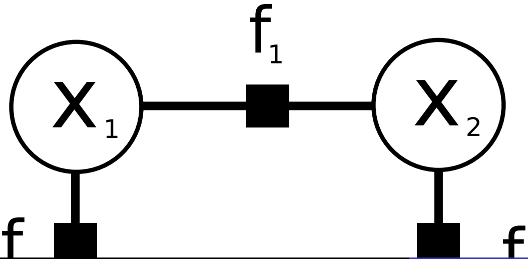
Bayesian Phase Unwrapping with Factor Graphs

6.556 Final Project

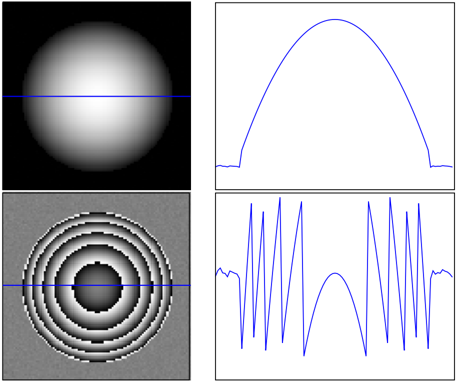
Eric Jonas

MIT Brain & Cognitive Sciences

May 12, 2009



Phase Unwrapping for MR

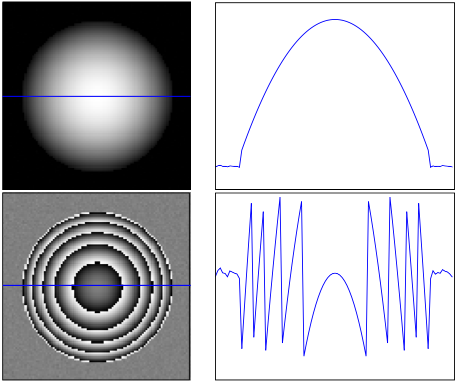


We think phase information varies smoothly except at boundaries

Right now most clinical applications only focus on magnitude

Phase provides additional information, such as field inhomogeneity and fluid velocity

Phase Unwrapping for MR



We think phase information varies smoothly except at boundaries

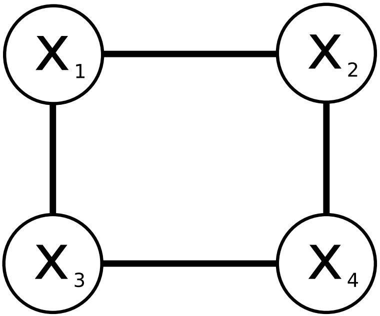
Right now most clinical applications only focus on magnitude

Phase provides additional information, such as field inhomogeneity and fluid velocity

Phase Wrapping

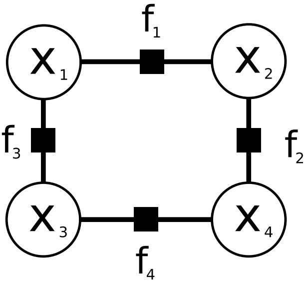
Our phase observations map $(-\infty, \infty)$ to $(0, 2\pi)$

Markov Random Fields



- You may know as "Undirected graphical model"
- Each X_i is a variable, we'll call it "state"
- Implicitly we know a series of joint distributions on the x_i represented by edges

Factor Graphs



- Factor Graphs [3] express the same concepts as MRFs but make the **factors** explicit.

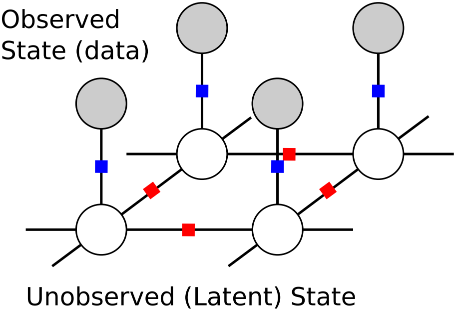
Factor Graph Probability

$$P(x_1, x_2, x_3, x_4) = f_1(x_1, x_2) \cdot f_2(x_2, x_3) \cdot f_3(x_3, x_4) \cdot f_4(x_4, x_1)$$

Example : Ising Model

Imagine you want to simulate a spin system Statistical physics people love doing this

Factor Graphs for Low-Level Vision



Properties of Image Factor Graphs [1]:

- Use observed state (data) to infer hidden state
- Have lattice structure like the ising
- Large number of vertices ($O(n)$ in number of pixels)
- $O(1)$ per-vertex connectivity
- Typically have homogeneous factors

Bayesian Factor Graphs

Bayes Rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{\sum_X P(Y|X)P(X)}$$

- $Y = y_{(i,j)}$: Observed Nodes
- $X = x_{(bi,j)}$: Hidden state we wish to estimate
- $P(Y|X)$: measurement model ("likelihood")
- $P(X)$: prior

Bayesian Factor Graphs For Low-Level vision

Our factor graph gives us P(X, Y), we want $P(X|Y)$.

Bayes Rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{\sum_X P(Y|X)P(X)} \propto P(Y|X) \cdot P(X)$$

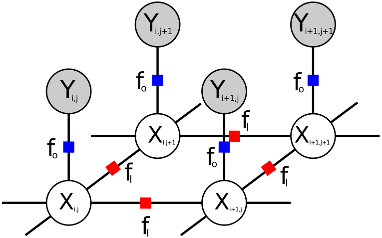
For a low-level vision factor graph,

$$P(Y|X) = \prod f_o(Y_i, X_i)$$
$$P(X) = \prod_{(i,j)\in G} f_l(X_i, X_j)$$

That sum, $\sum_X P(Y|X)P(X)$, looks hard! This is a big part of why Bayesian inference is tricky.

MRFs for Phase: Frey's approach

In Ying and Frey's model [8] they formulate 2-D phase unwrapping as a low-level vision MRF problem.



- $Y_{(i,j)} \in \mathbb{R}$ (continuous!)
- $x_{(i,j)} \in [0, 2\pi)$ (continuous!)

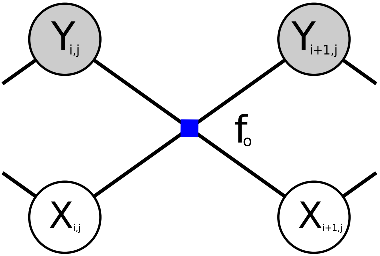
Observation Potential

$$f_o = \delta((Y_{(i,j)} mod 2\pi) - X_{(i,j)})$$

Latent potential

$$f_l(X_1, X_2) = (X_1 - X_2)^2$$

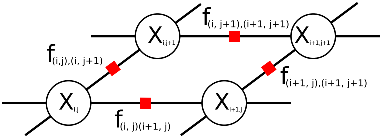
Discrete latent state, homogeneous factors



- $Y_{(i,j)} \in \mathbb{R}$
- $x_{(i,j)} \in [1 \ldots K]$

Discrete latent state, unique factors

I reformulated the problem to make inference easier.



- $x_{(i,j)} \in [1 \ldots K]$

Unique (non-homogeneous) factors

Every two adjacent X_i, X_j on the lattice have a unique factor between them that incorporates the information from the data.

Inference in MRFs

- The MRF tells us how to compute $P(X, Y)$.
- Bayes rule tells us how to compute $P(X|Y)$.

Bayes Rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{\sum_X P(Y|X)P(X)} \propto P(Y|X) \cdot P(X) = P(X, Y)$$

Inference in MRFs

- The MRF tells us how to compute $P(X, Y)$.
- Bayes rule tells us how to compute $P(X|Y)$.

Bayes Rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{\sum_X P(Y|X)P(X)} \propto P(Y|X) \cdot P(X) = P(X, Y)$$

Note: taking the above sum is hard!

There are some generic approaches to get around this:

- Traditional numerical integration

Inference in MRFs

- The MRF tells us how to compute $P(X, Y)$.
- Bayes rule tells us how to compute $P(X|Y)$.

Bayes Rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{\sum_X P(Y|X)P(X)} \propto P(Y|X) \cdot P(X) = P(X, Y)$$

Note: taking the above sum is hard!

There are some generic approaches to get around this:

- Traditional numerical integration
- optimize to find MAP solution

Inference in MRFs

- The MRF tells us how to compute $P(X, Y)$.
- Bayes rule tells us how to compute $P(X|Y)$.

Bayes Rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{\sum_X P(Y|X)P(X)} \propto P(Y|X) \cdot P(X) = P(X, Y)$$

Note: taking the above sum is hard!

There are some generic approaches to get around this:

- Traditional numerical integration
- optimize to find MAP solution
- Solve a similar, easier problem and perturb (variational methods)

Inference in MRFs

- The MRF tells us how to compute $P(X, Y)$.
- Bayes rule tells us how to compute $P(X|Y)$.

Bayes Rule

$$P(X|Y) = \frac{P(Y|X)P(X)}{\sum_X P(Y|X)P(X)} \propto P(Y|X) \cdot P(X) = P(X, Y)$$

Note: taking the above sum is hard!

There are some generic approaches to get around this:

- Traditional numerical integration
- optimize to find MAP solution
- Solve a similar, easier problem and perturb (variational methods)
- draw samples from $p(X|Y)$ to empirically estimate posterior

Markov-Chain Monte Carlo

Markov Property: next state only depends on current state

$$p(x_{t+1}|x_{1:t}) = p(x_{t+1}|x_t)$$

Main idea

Construct a Markov chain with the equilibrium distribution equal to your target distribution

Used in situations where you want to sample from $\pi(x)$ but only can compute $\pi^*(x) = Z\pi(x)$, Z unknown.
But how do we construct a chain with equilibrium distribution $\pi(x)$?

Metropolis-Hastings (1953)

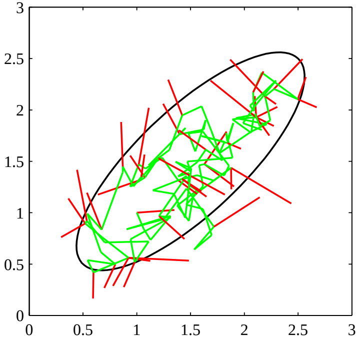
Simple idea: randomly propose new location in state space, sometimes go there! (From statistical physics [5])

We use a proposal distribution, $q(x \rightarrow x^*)$ Draw x^* as a *new target state* from this distribution.

$$x^* \sim q(x^*|x)$$

We compute the "Acceptance" ratio :

$$a = \min(1, \frac{p(x^*)}{p(x)} \cdot \frac{q(x \rightarrow x^*)}{q(x^* \rightarrow x)})$$



Gibbs Sampling (1990)

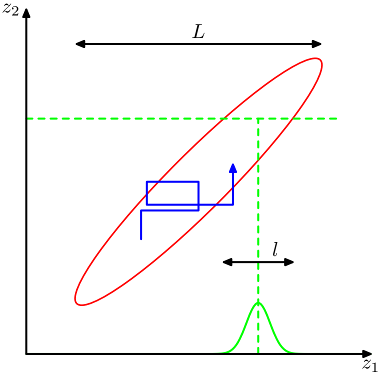
If we can sample from conditional distributions, we can use a more intelligent form of MH known as Gibbs Sampling [2]

Gibbs Sampling (1990)

If we can sample from conditional distributions, we can use a more intelligent form of MH known as Gibbs Sampling [2]

Requirements

To use Gibbs sampling on state variable x_i we must be able to draw samples from $x_i \sim p(x_i | \{x_{j \neq i}\})$

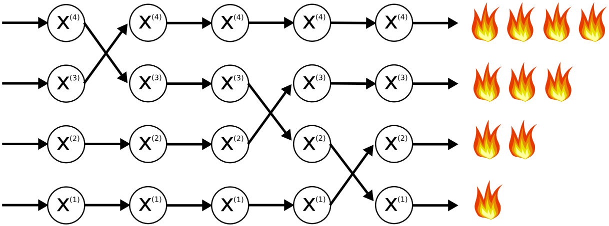


From Christopher Bishop.

Parallel Tempering (1991)

(aka "Replica Exchange Monte Carlo" [6], aka "Something to do with your 8 cores")

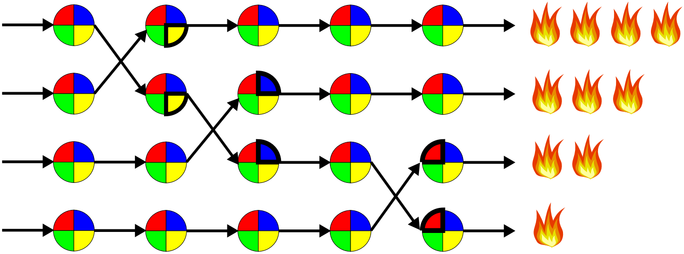
- Run N replicas of your chain, each at a different temperature
- Periodically propose MH-style swaps between adjacent chains
- Let's hot chains move around in flatter energy landscape



Partial Replica Exchange (2000)

(aka "Evolutionary MCMC" [4] aka "Return of the GA")

- Run N replicas of your chain, each at a different temperature
- Periodically propose MH-style swaps between **substates** in adjacent chains
- Let's hot chains move around in flatter energy landscape, but **lets better partial solutions move to cold chain**

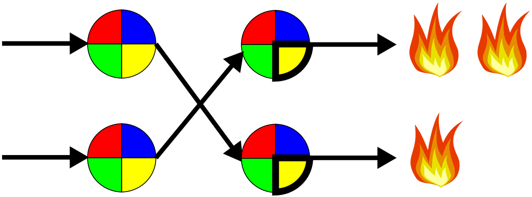


Data-driven MCMC (2002)

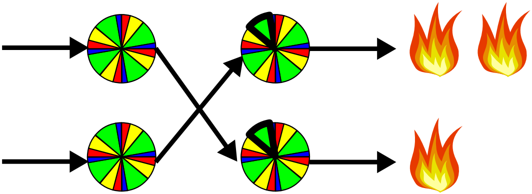
- Any "move" is valid as long as it is reversible.
- We can cheat a little bit and construct moves based on the data to help the chains mix, without changing the target distribution

Zhuowen and Tu [7] would pre-segment images with a heuristic and then use those segments as an MCMC proposal distribution.

Data-driven Partial Replica Exchange (2009)



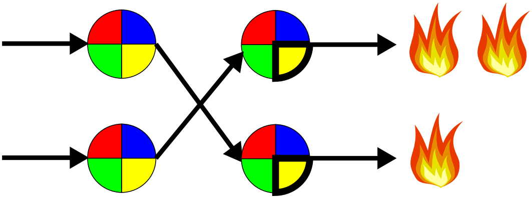
Traditional Partial Replica Exchange



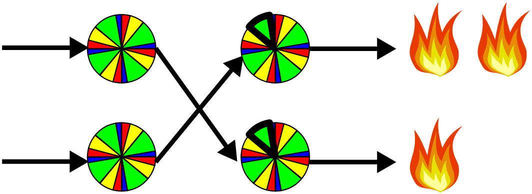
Data-Driven Partial Replica Exchange

- I just invented it!

Data-driven Partial Replica Exchange (2009)



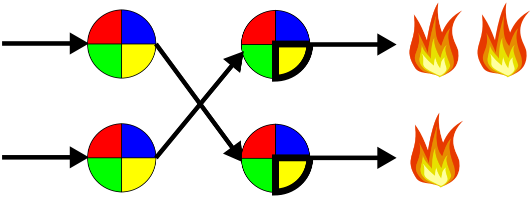
Traditional Partial Replica Exchange



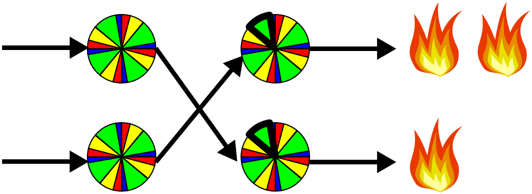
Data-Driven Partial Replica Exchange

- I just invented it!
- Well, okay, I glued together two existing methods.

Data-driven Partial Replica Exchange (2009)



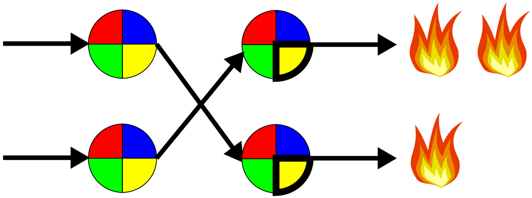
Traditional Partial Replica Exchange



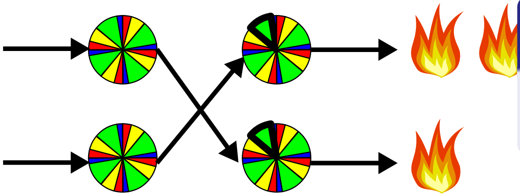
Data-Driven Partial Replica Exchange

- I just invented it!
- Well, okay, I glued together two existing methods.

Data-driven Partial Replica Exchange (2009)



Traditional Partial Replica Exchange



Data-Driven Partial Replica Exchange

- I just invented it!
- Well, okay, I glued together two existing methods.

This is the power of MCMC

Transition kernels (moves) nicely compose.

MRFs and Parallelism

The conditional independence assumptions allow fine-grained parallelism

Our Implementation

- Multi-threaded C++ engine, scales to 8 cores
- Python wrapper for GUI, vis, control, parameter tuning,

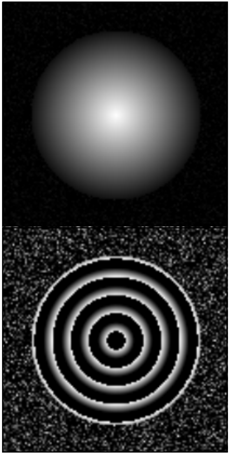
Our Implementation

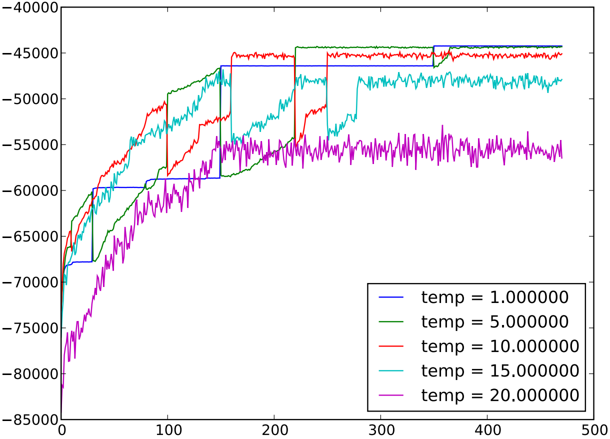
- Multi-threaded C++ engine, scales to 8 cores
- Python wrapper for GUI, vis, control, parameter tuning,

Try it out

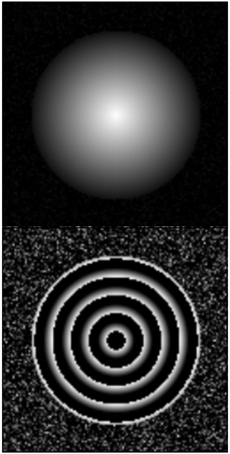
Source and presentation available for download on github
`https://github.com/ericmjonas/mrimrf/`

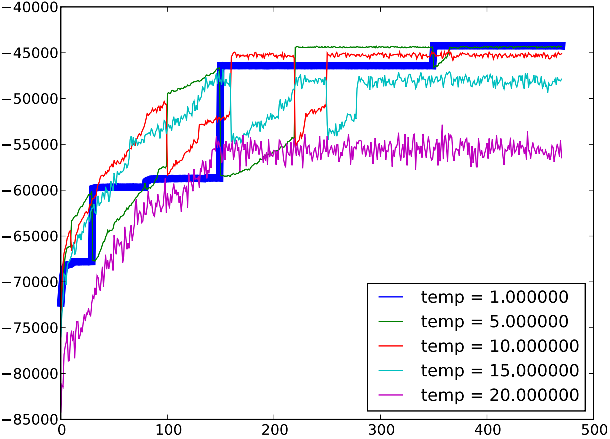
2-D Synthetic Data : The Sphere



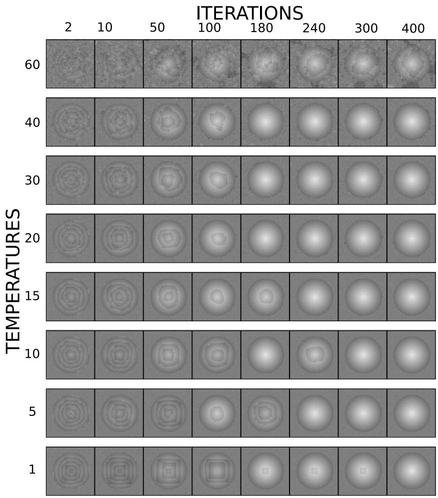


2-D Synthetic Data : The Sphere

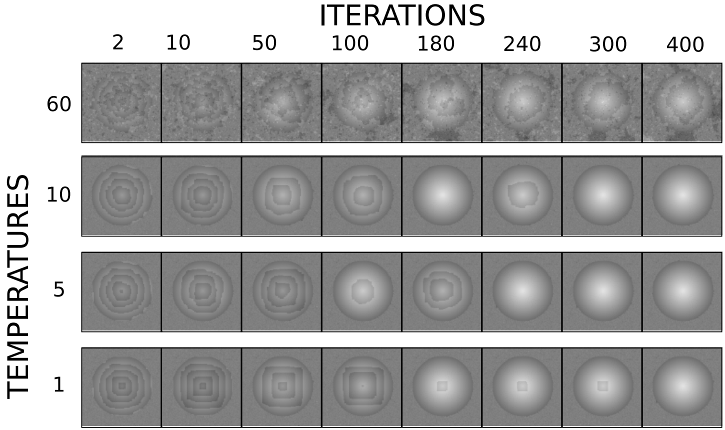




2-D Synthetic Data : The Sphere : PT exchange



2-D Synthetic Data : The Sphere : PT exchange



Div and Audrey

Why, man, why?

Why go through all this effort for mediocre performance?

- Why be Bayesian? - incorporate prior information explicitly - probabilistic models compose

Why, man, why?

Why go through all this effort for mediocre performance?

- Why be Bayesian? - incorporate prior information explicitly - probabilistic models compose
- Why use MCMC? - MCMC "black boxes" often do the right thing with minimal work - Kernels compose nicely [**?**]

Why, man, why?

Why go through all this effort for mediocre performance?

- Why be Bayesian? - incorporate prior information explicitly - probabilistic models compose
- Why use MCMC? - MCMC "black boxes" often do the right thing with minimal work - Kernels compose nicely [**?**]
- Why use factor graphs? - reasonable prior expressing local dependence - factors allow programmatic evaluation of conditional independencies, thus can easily identify opportunities for parallelism

Where to now?

- Exact sampling using Systematic Stochastic Search [?]

Where to now?

- Exact sampling using Systematic Stochastic Search [?]
- Better neighborhood connectivity

Where to now?

- Exact sampling using Systematic Stochastic Search [?]
- Better neighborhood connectivity
- Better data likelihood

Where to now?

- Exact sampling using Systematic Stochastic Search [?]
- Better neighborhood connectivity
- Better data likelihood
- GPU implementation – CUDA makes this easy

Where to now?

- Exact sampling using Systematic Stochastic Search [?]
- Better neighborhood connectivity
- Better data likelihood
- GPU implementation – CUDA makes this easy
- Better visualization of posterior?

Where to now?

- Exact sampling using Systematic Stochastic Search [?]
- Better neighborhood connectivity
- Better data likelihood
- GPU implementation – CUDA makes this easy
- Better visualization of posterior?
- Reconstruction?

More information

Try it out

Source and presentation available for download on github

`https://github.com/ericmjonas/mrimrf/`

More information

Try it out

Source and presentation available for download on github

https://github.com/ericmjonas/mrimrf/

Questions?

References I

William Freeman and Egon Pasztor.
Markov networks for low-level vision.
MERL Tech Report, TR99(08), February 1999.

S. Geman and D. Geman.
Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, pages 452–472.
Morgan Kaufmann Publishers Inc., 1990.

F.R. Kschischang, B.J. Frey, and H.-A. Loeliger.
Factor graphs and the sum-product algorithm.
Information Theory, IEEE Transactions on, 47(2):498–519, 2001.

References II

Faming Liang and Wing Hung Wong.
Evolutionary monte carlo: Applications to c_p model sampling and change point problem.
STATISTICA SINICA, 10:317—342, 2000.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller.
Equation of state calculations by fast computing machines.
The Journal of Chemical Physics, 21(6):1087–1092, June 1953.

Robert H. Swendsen and Jian-Sheng Wang.
Replica monte carlo simulation of Spin-Glasses.
Physical Review Letters, 57(21):2607, November 1986.
Copyright (C) 2009 The American Physical Society; Please report any problems to prola@aps.org.

References III

Zhuowen Tu and Song-Chun Zhu.
Image segmentation by data-driven markov chain monte carlo.
Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(5):657–673, 2002.

Lei Ying, Zhi-Pei Liang, D.C. Munson, R. Koetter, and B.J. Frey.
Unwrapping of MR phase images using a markov random field model.
Medical Imaging, IEEE Transactions on, 25(1):128–136, 2006.