$f(x) = x$ **Papers We Love**

$$f = id$$

Papers We Love Remote Meetup #2 - Philip Wadler

Recording Hosted by Papers We Love

More

papers-we-love-oct.pdf

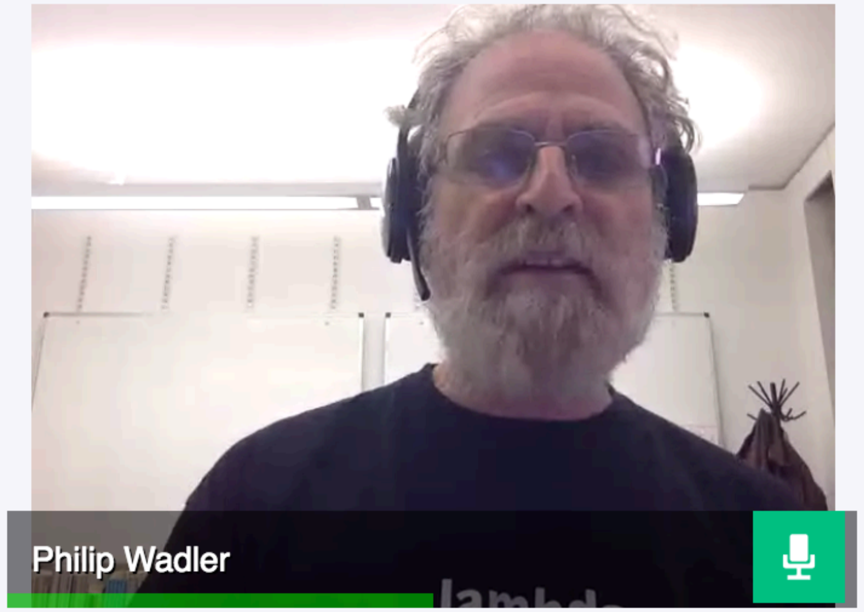
Full

John McCarthy, Toward a Mathematical Science of Computation, *IFIP Congress*, pages 21–28, 1962.

Peter Landin, The Next 700 Programming Languages, *CACM*, 9(3):157–166, March 1966.

Gordon Plotkin, Call-by-name, Call-by-value, and the Lambda Calculus, *TCS* 1:125–159, 1975.

Slide 5 of 45




Now speaking: **Philip Wadler**

CHAT PEOPLE POLL Q&A

#PwLRemote

N **nicolas** · 9:06:26 pm

:) thank you mr reynolds

**guest-475** · 9:08:02 pm

I imagine its this one: http://wadler.blogspot.se/2016/06/papers-we-love-john-reynolds_10.html

V Type your message here...

Q&A Post Tweet

# (Speculative) Motivation

- → "Early Vision"
- → computer vision papers keep mentioning *random fields*
- → especially they somehow end up in neural nets
- → words from AI BS bingo seem to have rigorously and probabilistically treated cousins

# Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images

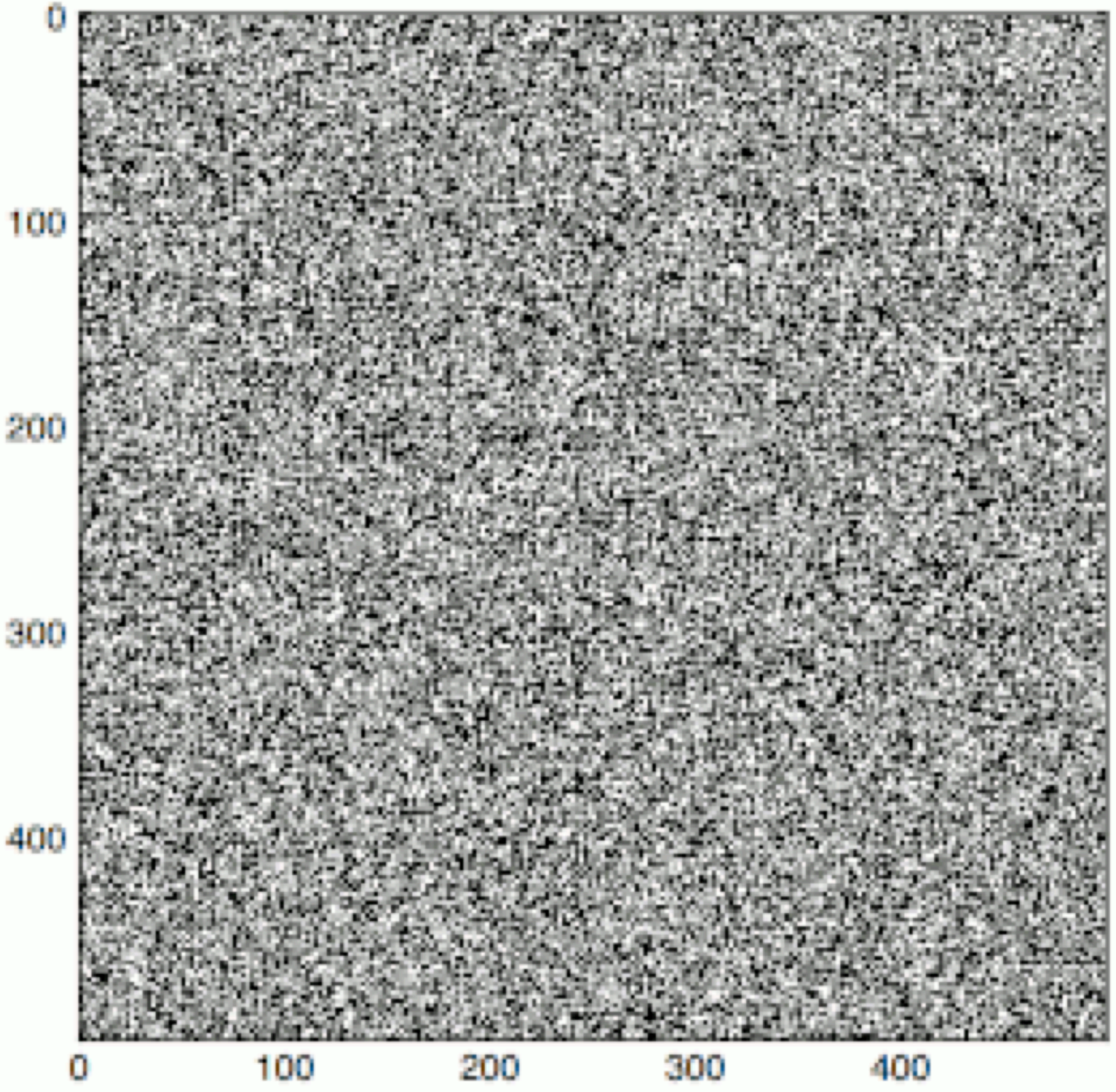
STUART GEMAN AND DONALD GEMAN

(Mis-)Interpretation by Vlad Ki

Scholar articles

Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images
S Geman, D Geman - IEEE Transactions on pattern analysis and machine …, 1984
Cited by 18679 - Related articles - All 36 versions

# Statistical Physics


0
100
200
300
400
0
100
200
300
400

# Outline

→ **Bayesian**

*(an inference framework)*

→ **MRFs & Images**

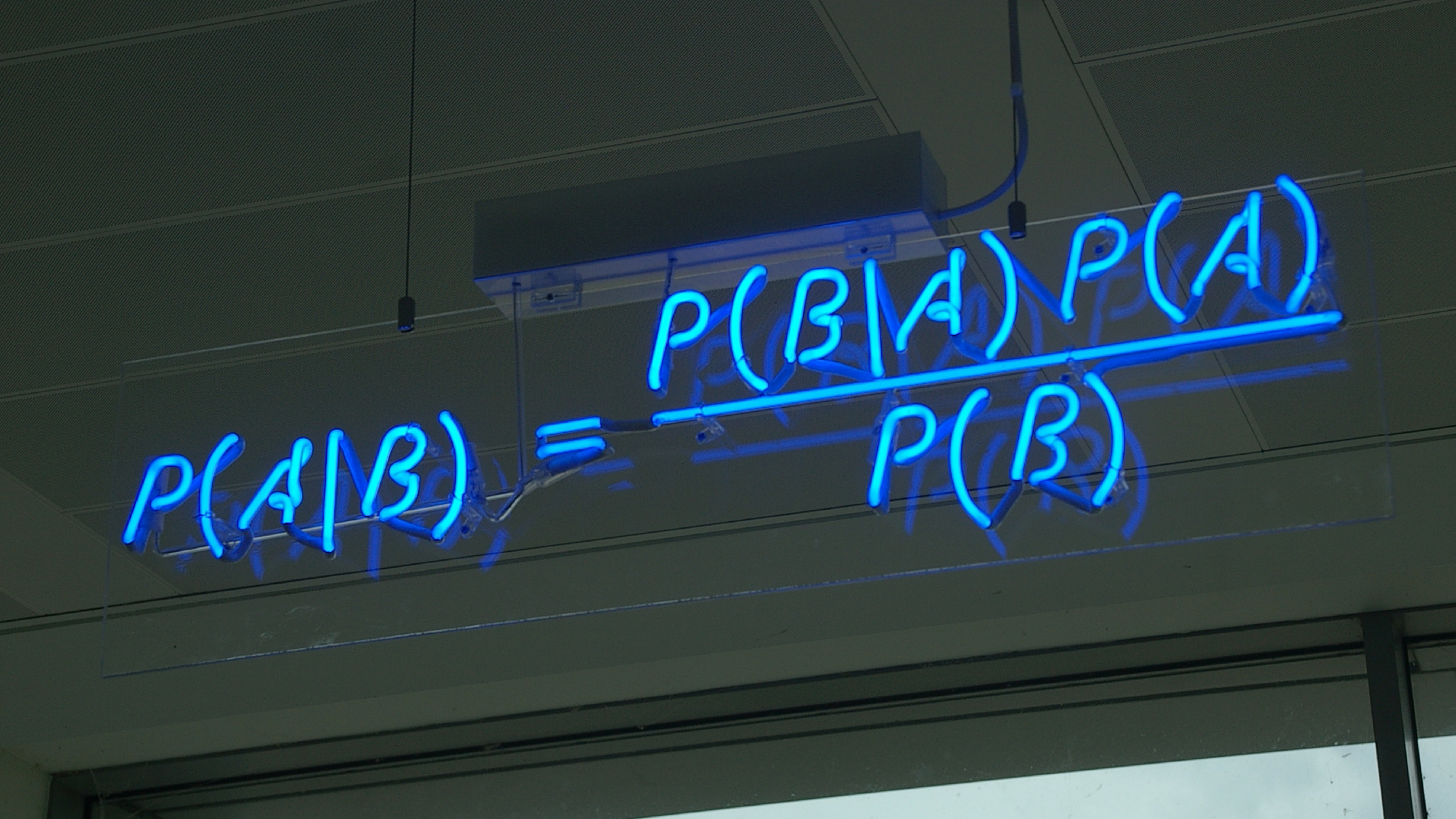
*(modeling image restoration images)*

→ **Gibbs Sampler**

*(a computationally tractable representation for MRFs defined over graphs)*

→ **Relaxation**

*(computing the best Gibbs)*


$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

# (Statistics) Framework Wars

Frequentist to Bayesian is like Angular to React

# Bayes

- → originally published to make theological arguments
- → made cool by Laplace
- → inference and decision come as plugins
- → works recursively (Kalman, Particle filters & more, aka sequential estimation)
- → AI without a GPU: just refine your priors

→ y - signal (measurement, evidence, input)

→ **x - state** (unknown, hypothesis, output)

think: model parameters vs data

$$P(\mathbf{x}|y) = \frac{P(y|\mathbf{x})P(\mathbf{x})}{P(y)}$$

**posterior = likelihood times prior, over evidence**

Bayes rule lets us swap those!

# Estimation problems

→ **MAP** - maximum a posteriori

→ **MLE** - maximum likelihood (aka MAP when you have no prior)

$$-\log P(\mathbf{x}|y) = -\log P(y|\mathbf{x}) - \log P(\mathbf{x}) + C$$

**log**s are like probability buffs: slay **exp**s,
turn ugly $\prod$ into neat $\sum$

negation turns maximization into minimization

# Aside

$$E(\mathbf{x}, y) = E_d(\mathbf{x}, y) + E_p(\mathbf{x})$$

**energy $\sim$ log likelihood**

(stat physics and CV people like saying *energy* a lot)

# Graphs

$$S = \{s_1, s_2, \cdots, s_N\}$$

sites (nodes)

$$\mathscr{G} = \{\mathscr{G}_s, s \in S\}$$

neighborhood family (edges per site)

$$\mathscr{G}_s = \{t \mid t \in S, t \in \mathscr{G}_s \leftrightarrow s \in \mathscr{G}_r\}$$

neighborhood set, symmetric (undirected)

$$C \subset S$$

cliques are subsets of sites

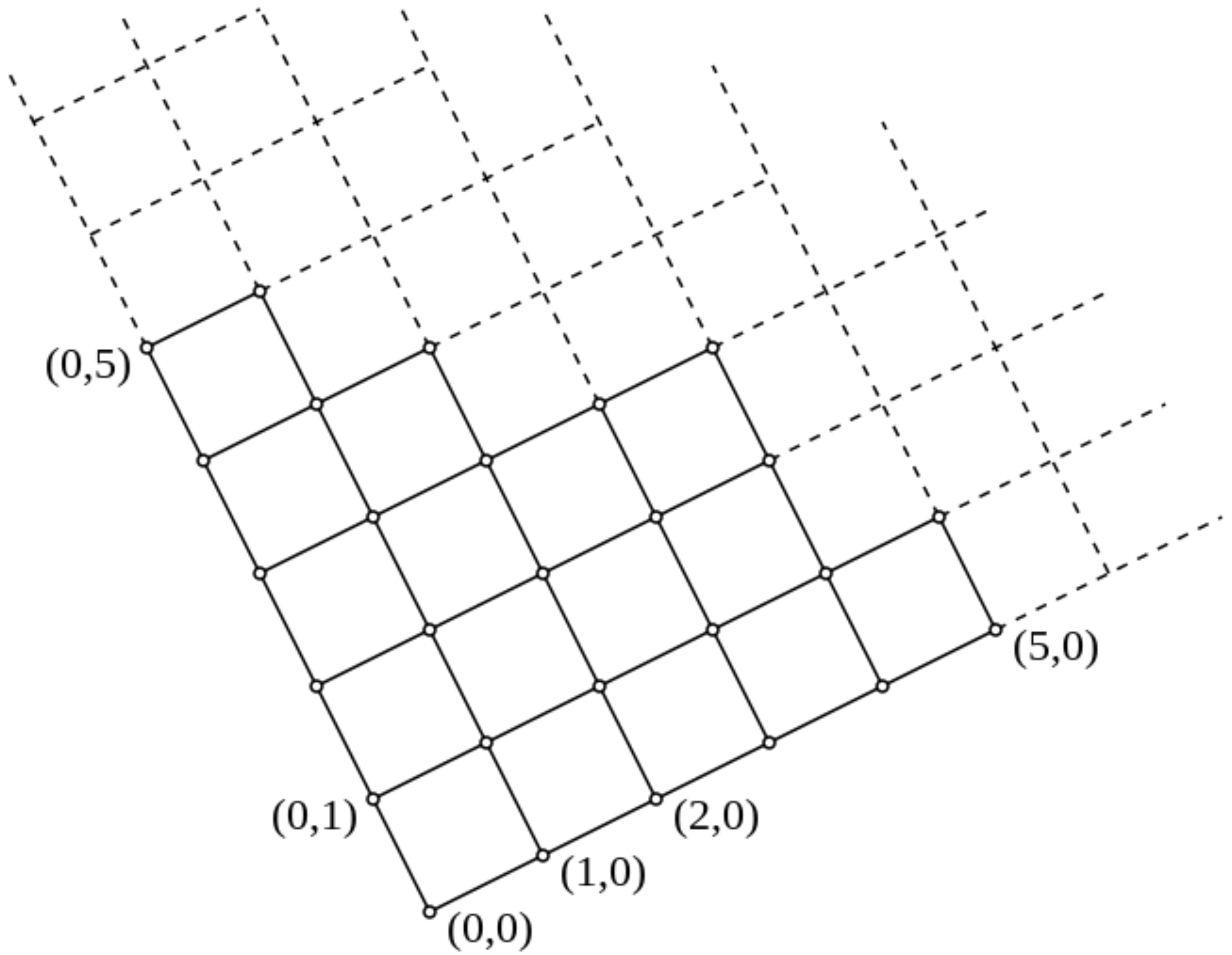
$$C = \{s, t, \cdots \in S \mid \forall (s \in C, t \in C) : s \in \mathscr{G}_t \wedge t \in \mathscr{G}_s\}$$

such that every pair in the set is a neighbor

$\mathcal{C}$ means "all cliques": $C \in \mathcal{C}$

# Graphs and Images

$m \times m$ **lattice**: $Z_m = \{(i, j) : 1 \leq i, j \leq m\}$



$$S_{m\times m} = Z_m$$

Pixels: $\{F(i,j) = f : \Lambda\}$

$\Lambda \in \{i \mid \mathbf{0} < i < L\}$ (grayscale)

$L = \mathbf{255}$ - 8-bit grayscale

$$\mathcal{F} = \{\mathcal{F}_{i,j} \,|\, (i,j) \in Z_m, \mathcal{F}_{i,j} \subset Z_m\}$$

# Homogeneous neighborhoods

$$\mathscr{G} = \mathcal{F}_c = \{\mathcal{F}_{i,j} \mid (i,j) \in Z_m\}$$

$$\mathcal{F}_{i,j} = \{(k,l) \in Z_m \mid 0 \leq (k-i)^2 + (l-j)^2 \leq c\}$$

# More than pixels

$$S = D_m$$

"dual" lattice for edge elements, midway between each pixel pair

# Or take both

$$S = Z_m \cup D_m$$

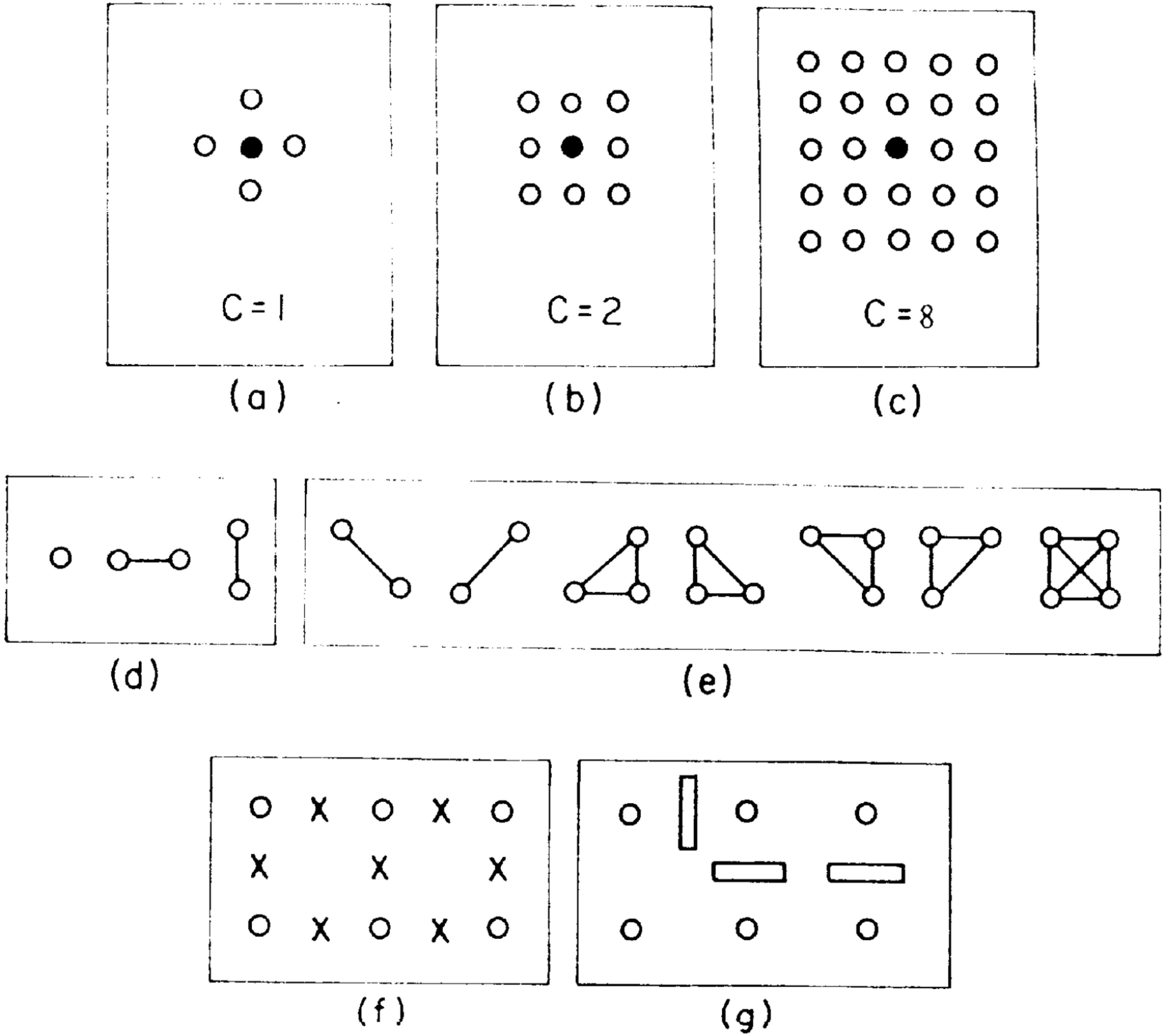



Fig. 1.

# MRF

## Markov Random Field

$$\boldsymbol{X} = \{\boldsymbol{X}_s \mid \boldsymbol{s} \in \boldsymbol{S}\}$$

family of random variables per site

X is a **MRF** wrt $\mathscr{G}$ if

$$P(X = \omega) > \mathbf{0}\ \forall \omega \in \mathbf{\Omega}$$

all probabilities are positive

$$P(X_s = x_s \mid X_r = x_r, r \neq s)$$

probability of a site given others

$$= P(X_s = x_s | X_r = x_r, r \in \mathscr{G}_s)$$

is the same as a probability of a site given its neighbors (Markov Property)

# Markov Property

→ probability at site $\boldsymbol{k}$ depends only at values of a finite neighborhood

$$\boldsymbol{P(\sigma_k = s \mid \sigma_j,\, j \neq k) = P(\sigma_k = s \mid \sigma_j,\, j \in N_k)}$$

→ $\boldsymbol{N_k}$ neighborhood of site $\boldsymbol{k}$

# Bayesian Model

# Noisy Image Prior

$$G = \phi(H(F)) \odot N$$

→ $\boldsymbol{F(i,j) : \Lambda}$: pixel intensities ($\boldsymbol{\Lambda \in \{i \mid 0 \leq i \leq 255\}}$)

→ $\phi$ - some nonlinearity (like $\sqrt{\boldsymbol{x}}$)

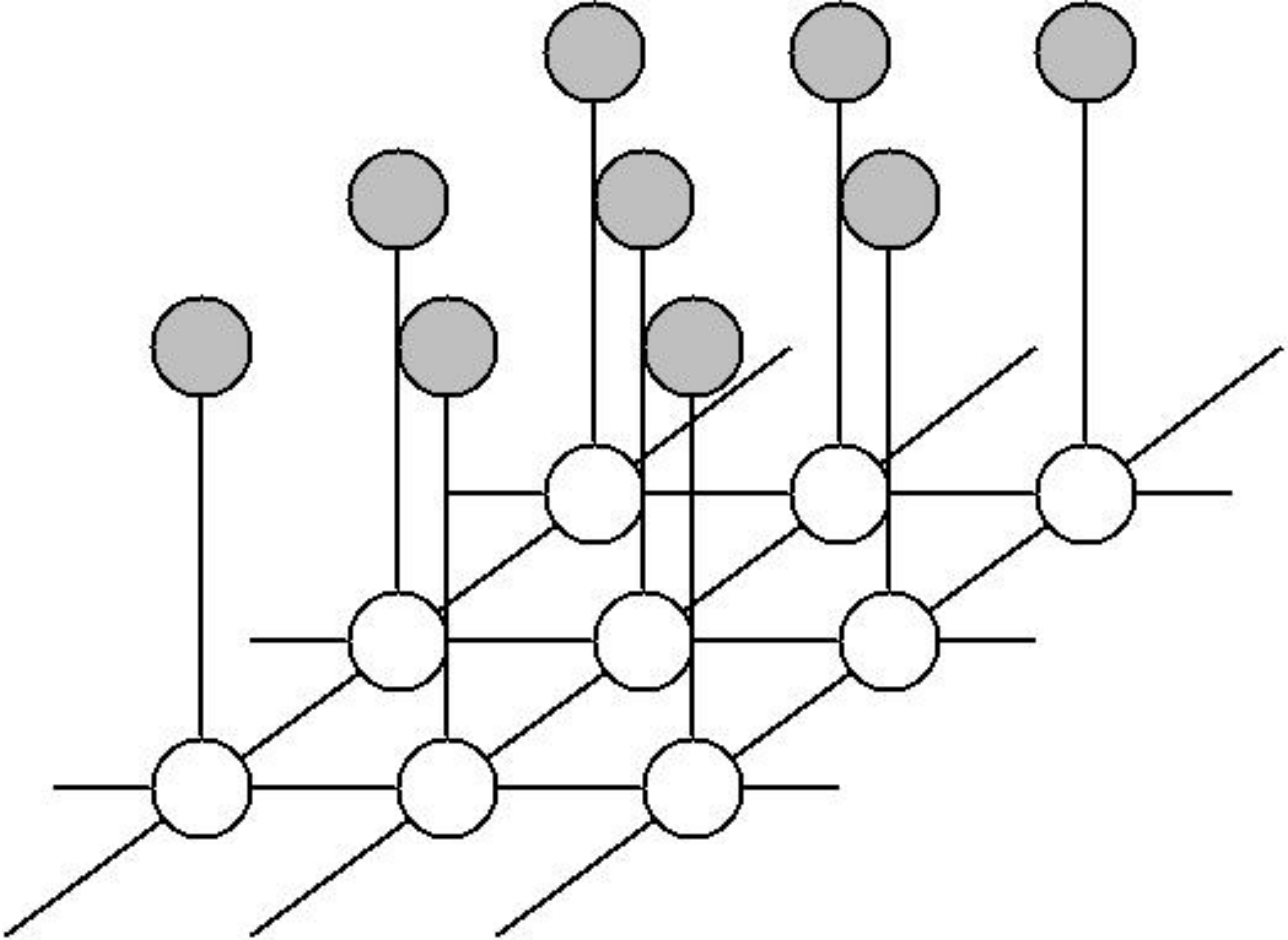
→ $\boldsymbol{H}$ - blur

→ $\boldsymbol{N}$ - invertible noise

# Original Image

$X = (F, \mathcal{L})$ is a MRF over a graph that contains original intensities ($F$) and image edges ($\mathcal{L}$).

A set of all possible configurations of $X$:
$\Omega = \{\omega = x_{s_1}, \cdots, x_{s_N}\}$



# Posterior

$$P(X = \omega \mid G = g) \propto P(G = g \mid X = \omega) \times P(X = \omega)$$

What was the original image given the data?

# Maximum A Posteriori

Let's enumerate $L^{m^2}$ states!

# Hammersley-Clifford Theorem

→ any probability measure that satisfies a Markov property is a Gibbs distribution for an appropriate choice of (locally defined) energy function

**MRF $\sim$ Gibbs**

# Gibbs Distribution

aka Boltzmann Distribution

$$\pi(\omega) = \frac{1}{Z}\, e^{-\frac{U(\omega)}{T}}$$

$T$ - temperature
$U(\omega)$ - **energy** function

$$Z = \sum_{w} e^{-\frac{U(\omega)}{T}}$$

→ normalizing **constant** ("partition function")

# What would U be?

$$U(\omega) = \sum_{c \in \mathcal{C}} V_c(\omega)$$

→ $\{V_c, c \in \mathcal{C}\}$ - set of graph clique potentials

# Gibbs Sampler

→ introduced in the subj paper

→ produces a Markov chain $\{X(t), t = 0, 1, 2 \dots\}$ with $\boldsymbol{\pi}$ as equilibrium distribution

→ MCMC algorithm family

$$P(X_s(t) = x_s, s \in S) = \pi(X_{n_t} = x_{n_t} \mid X_s = x_s, s \neq n_t)\, P(X_s(t-1) = x_s, s \neq n_t)$$

**P(new state of site) = Gibbs(visited now | others) P(everyone else before)**

→ This MAP is a statistical process itself (hence MCMC)

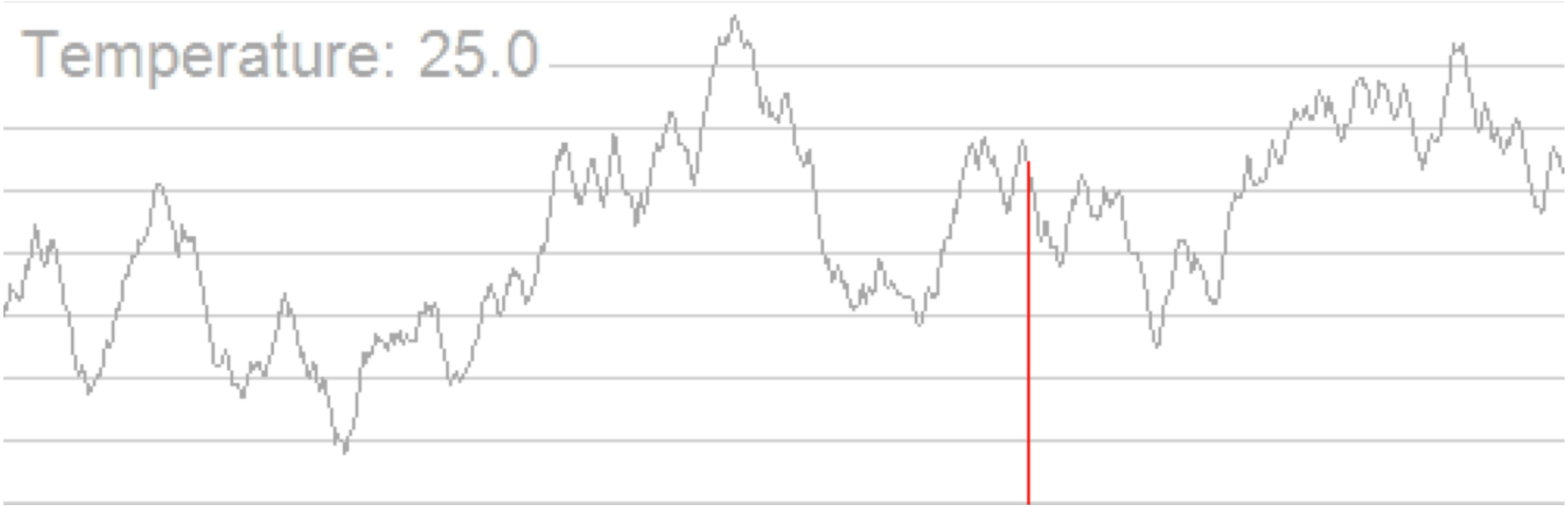
→ Parallel!

# Theorem A (Relaxation)

$$\lim_{t \to \infty} P(X(t) = \omega | X(0) = \eta) = \pi(\omega)$$

No matter where we start ($\boldsymbol{\eta}$), our state $\boldsymbol{\omega}$ will end up a Gibbs Distribution $\boldsymbol{\pi}$ if we keep ($\mathbf{lim}$) sampling infinitely $\boldsymbol{t \to \infty}$.

# Simulated Annealing



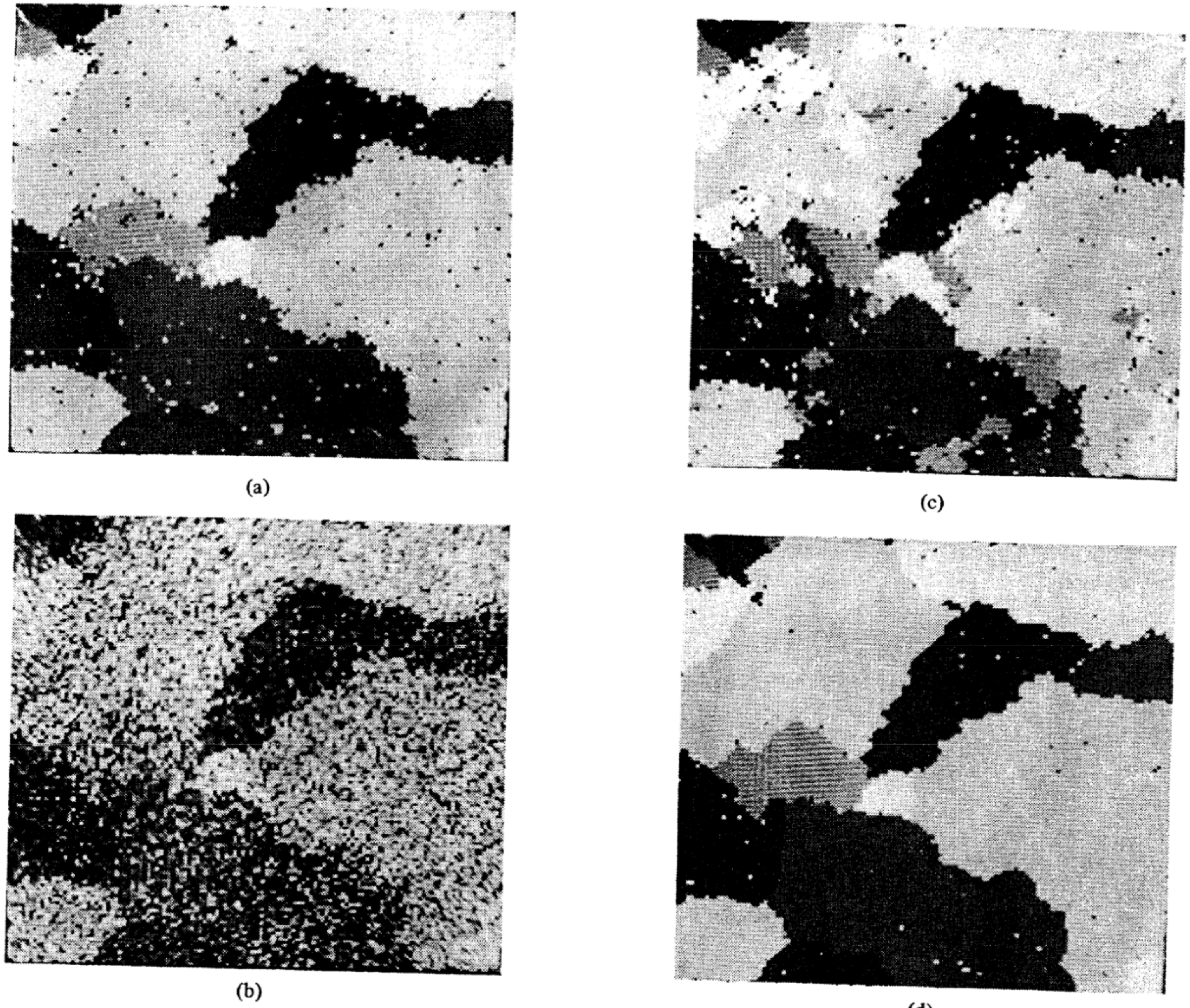
# Experiments

# Clique potentials?

## What configurations of local variables are preferred to others

# What are clique potentials for restoring images?

[end of section **XIII**]



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

Fig. 2. (a) Original image: Sample from MRF. (b) Degraded image: Additive noise. (c) Restoration: 25 iterations. (d) Restoration: 300 iterations.

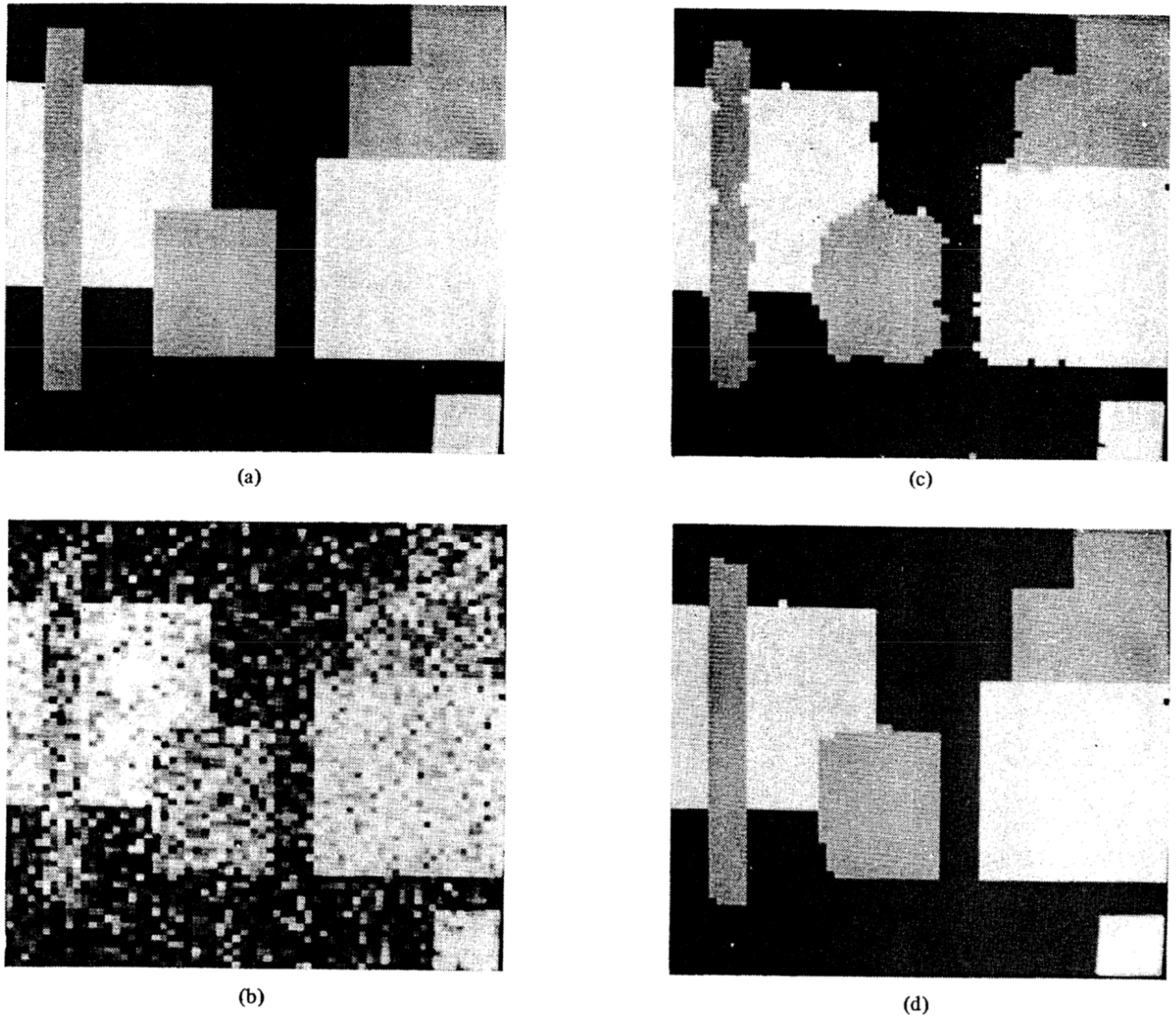


Fig. 4. (a) Original image: "Hand-drawn." (b) Degraded image: Additive noise. (c) Restoration: Without line process; 1000 iterations. (d) Restoration: Including line process; 1000 iterations.

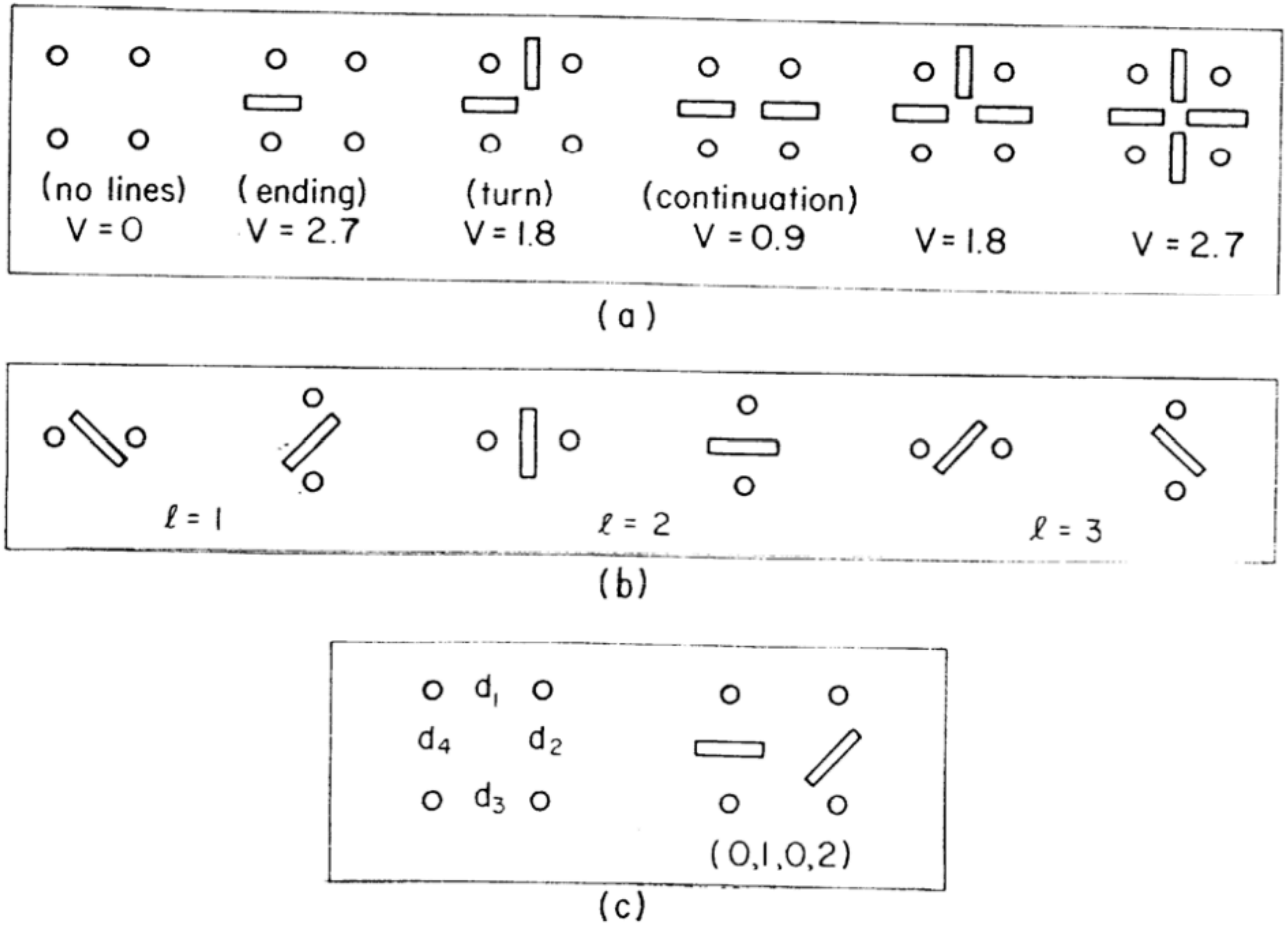

(no lines)
V = 0
(ending)
V = 2.7
(turn)
V = 1.8
(continuation)
V = 0.9
V = 1.8
V = 2.7
(a)
ℓ = 1
ℓ = 2
ℓ = 3
(b)
d1
d4
d2
d3
(0,1,0,2)
(c)

Fig. 5.

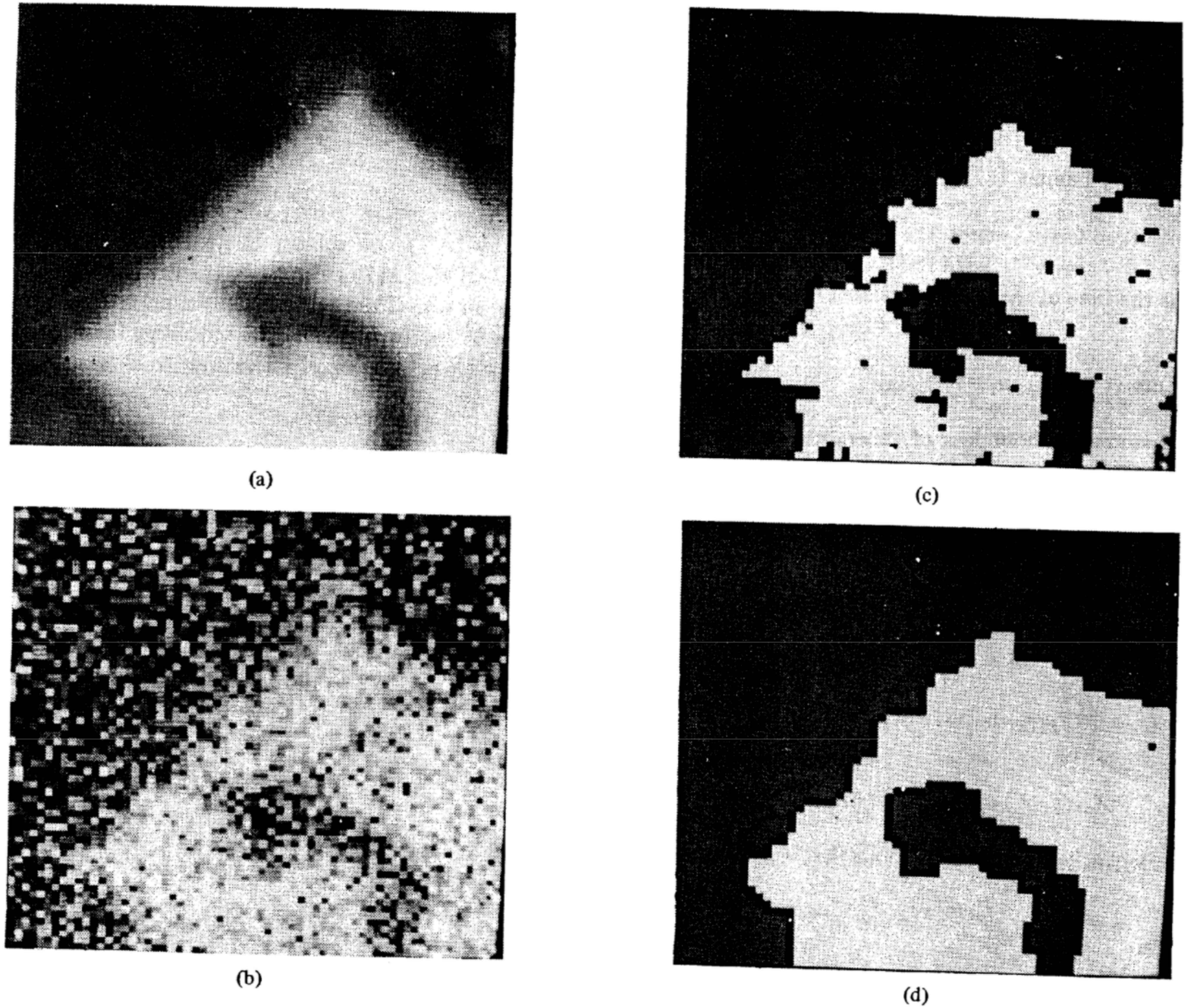



Fig. 7. (a) Blurred image (roadside scene). (b) Degraded image: Additive noise. (c) Restoration including line process; 100 iterations. (d) Restoration including line process; 1000 iterations.

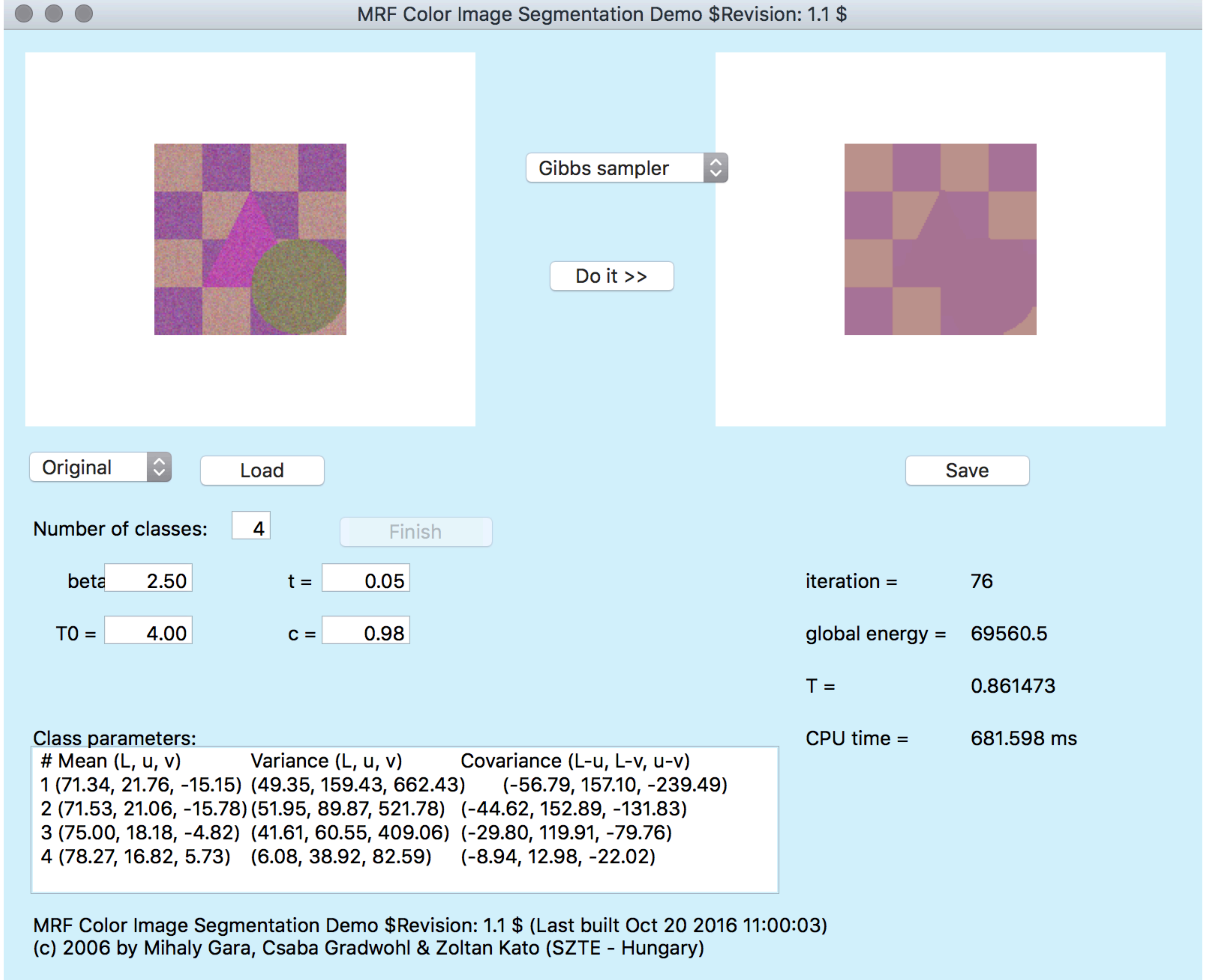
# Runnable Demo

→ http://www.inf.u-szeged.hu/~kato/software/

→ "Supervised Image Segmentation Using Markov Random Fields"

→ "Supervised Color Image Segmentation in a Markovian Framework"

→ Usable on https://github.com/proger/mrf


MRF Color Image Segmentation Demo $Revision: 1.1 $
Gibbs sampler
Do it >>
Original
Load
Save
Number of classes: 4
Finish
beta 2.50
t = 0.05
T0 = 4.00
c = 0.98
iteration = 76
global energy = 69560.5
T = 0.861473
CPU time = 681.598 ms
Class parameters:
# Mean (L, u, v) Variance (L, u, v) Covariance (L-u, L-v, u-v)
1 (71.34, 21.76, -15.15) (49.35, 159.43, 662.43) (-56.79, 157.10, -239.49)
2 (71.53, 21.06, -15.78) (51.95, 89.87, 521.78) (-44.62, 152.89, -131.83)
3 (75.00, 18.18, -4.82) (41.61, 60.55, 409.06) (-29.80, 119.91, -79.76)
4 (78.27, 16.82, 5.73) (6.08, 38.92, 82.59) (-8.94, 12.98, -22.02)
MRF Color Image Segmentation Demo $Revision: 1.1 $ (Last built Oct 20 2016 11:00:03)
(c) 2006 by Mihaly Gara, Csaba Gradwohl & Zoltan Kato (SZTE - Hungary)

# Beyond

→ Do more things with MRFs! (like segmentation)

→ MAP using Graph Cuts (Ford-Fulkerson for max-flow/min-cut)

→ CRF (learning potentials by conditioning on training data)

...

# Next steps

Probabilistic Graphical Models by Daphne Koller

(MRF is a "undirected probabilistic graphical model")

Pattern Recognition and Machine Learning by Christopher Bishop

(More theory on everything)