Algorithms from Statistical Physics for Generative Models of Images.

J.M. Coughlan
Smith-Kettlewell Eye Research Institute
San Francisco, CA 94115

Alan Yuille
Department of Statistics
University of California at Los Angeles
Los Angeles, CA 90095
yuille@stat.ucla.edu

In Image and Vision Computing. Vol. 21. No. 1. pp 29-36. January. 2003.

Algorithms from Statistical Physics for Generative Models of Images

James Coughlan and Alan Yuille

Smith-Kettlewell Eye Research Institute
2318 Fillmore Street, San Francisco, CA 94115.

Abstract

A general framework for defining generative models of images is Markov random fields (MRF's), with shift-invariant (homogeneous) MRF's being an important special case for modeling textures and generic images. Given a dataset of natural images and a set of filters from which filter histogram statistics are obtained, a shift-invariant MRF can be defined (as in Zhu [12]) as a distribution of images whose mean filter histogram values match the empirical values obtained from the data set. Certain parameters in the MRF model, called potentials, must be determined in order for the model to match the empirical statistics. Standard methods for calculating the potentials are computationally very demanding, such as Generalized Iterative Scaling (GIS), an iterative procedure that converges to the correct potential values. We define a fast approximation, called BKGIS, which uses the Bethe-Kikuchi approximation from statistical physics to speed up the GIS procedure. Results are demonstrated on a model using two filters, and we show synthetic images that have been sampled from the model. Finally, we show a connection between GIS and our previous work on the g-factor.

Key words: Markov random fields, Generalized Iterative Scaling, Minimax Entropy Learning, histogram statistics.

Email address: coughlan/yuille@ski.org (James Coughlan and Alan Yuille).

1 Introduction

It is increasingly important to learn generative models for vision from real data. A general framework for defining generative models of images is Markov random fields (MRF's), which may be used to define a probability distribution on an entire image pixel lattice. An MRF probability distribution is defined in terms of clique potential functions, referred to as "potentials," which are functions of local clusters of pixels ("cliques") that enforce the desired statistical relationships among the pixel intensity values in these clusters. An important sub-class of MRF's are those that are shift-invariant (homogeneous), i.e. those for which the potential functions are the same from clique to clique. Shift-invariant MRF's can be used for modeling statistically homogeneous patterns such as textures and generic images. Pioneering work by Zhu, Wu, and Mumford [12] introduced the Minimax Entropy Learning scheme which enabled them to learn shift-invariant MRF distributions for images based on filter histograms obtained from a dataset of images. This work gave an elegant connection between generative models on images (eg. [12],[11]) and empirical studies of the statistical properties of images, for example see [7],[6], [5].

Learning MRF distributions from empirical image data requires calculating the values of the potential functions that result in a distribution which is consistent with the empirical data. (This corresponds to the classic problem of estimating the parameters of a log-linear model.) Standard methods for calculating the potentials are computationally very demanding, such as Generalized Iterative Scaling (GIS), an iterative procedure due to Darroch and Ratcliff [3] that is guaranteed to converge to the correct potential values. GIS may be thought of as a form of steepest descent in which each iteration updates the potential values. For the type of MRF we consider in this paper, shift-invariant MRF's defined in terms of histograms of filter responses, each iteration of GIS requires calculating the mean filter histogram values, or histogram expectations, given the current value of the potentials. Calculating the histogram expectations is the computational bottleneck of GIS, since closed-form expressions for these expectations are intractable, and estimating expectations by methods such as MCMC is very slow.

To speed up this bottleneck, we apply recent work on the Bethe-Kikuchi approximation [9], which is a standard approximation in statistical physics [4], to estimate the histogram expectations at each step of GIS. We name this algorithm BKGIS. Since the Bethe-Kikuchi approximation is a variational procedure that requires constrained optimization, we employ the recently devised CCCP algorithm [10] to perform the required constrained optimization. We apply the CCCP algorithm in a way that exploits the homogeneous structure of the MRF, resulting in an algorithm that converges quickly to a solution of the Bethe-Kikuchi approximation. We demonstrate our work by learning MRF models for generic images and generating corresponding image samples.

In addition, we show a direct relationship between the GIS algorithm and a previous method, called the multinomial approximation, proposed for estimating potentials using an independence assumption [2]. We show that this previous approach corresponds to the first iteration of the GIS algorithm with uniform initial conditions. This means that a single iteration of GIS can be sufficient to get a good approximation to the MRF potentials. The same relationship holds for BKGIS.

In Section (2), we briefly review Markov random fields and Minimax Entropy Learning. Section (3) introduces the Generalized Iterative Scaling algorithm and demonstrates that the first iteration of GIS corresponds to the multinomial method. In Section (4) and (4.2) we describe the BKGIS algorithm which uses the Bethe-Kikuchi approximation and a CCCP algorithm to speed up GIS. Section (5) gives results. Finally, in Appendices A and B we review the multinomial approximation method and show how its properties can be computed efficiently.

2 Shift-Invariant Markov Random Fields

Suppose we have training image data which we assume has been generated by an (unknown) probability distribution $P_T(\vec{x})$, where $\vec{x}$ represents an image. A Markov random field (MRF) may be used to construct a generative model of $P_T(\vec{x})$, and one procedure for defining an appropriate MRF is given by Minimax Entropy Learning (MEL) [12]. The MEL procedure

approximates $P_T(\vec{x})$ by selecting the distribution with maximum entropy constrained by observed feature statistics $< \vec{\phi}(\vec{x}) >= \vec{\psi}_{obs}$. This gives the exponential form $P(\vec{x}|\vec{\lambda}) = \frac{e^{\vec{\lambda}\cdot\vec{\phi}(\vec{x})}}{Z[\vec{\lambda}]}$, where $\vec{\lambda}$ is a vector of parameters, called the *potentials*, chosen such that $\sum_{\mathbf{x}} P(\vec{x}|\lambda)\phi(\vec{x}) = \vec{\psi}_{obs}$. This procedure is equivalent to performing maximum likelihood estimation of $\vec{\lambda}$ given the exponential form.

We will treat the special case where the statistics $\vec{\phi}$ are the histogram of a shift-invariant filter $\{f_i(\vec{x}) : i = 1, ..., N\}$, where N is the total number of pixels in the image. So $\psi_a = \phi_a(\vec{x}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{a,f_i(\vec{x})}$ where $a = 1, ..., Q$ indicates the (quantized) filter response values. The potentials become $\vec{\lambda}\cdot\vec{\phi}(\vec{x}) = \frac{1}{N}\sum_{a=1}^{Q}\sum_{i=1}^{N}\lambda(a)\delta_{a,f_i(\vec{x})} = \frac{1}{N}\sum_{i=1}^{N}\lambda(f_i(\vec{x}))$. Hence $P(\vec{x}|\vec{\lambda})$ becomes a MRF distribution with clique potentials given by $\lambda(f_i(\vec{x}))$. This determines a shift-invariant Markov random field with the clique structure given by the filter responses $\{f_i\}$.

Multiple filters may be used simultaneously. We denote each of the M filters by $f_i^{(k)}(\vec{x})$, where the superscript k ranges from 1 through M and labels the filter. In this case the distribution becomes

$$P(\vec{x}|\vec{\lambda}^{(1)}, \ldots, \vec{\lambda}^{(M)}) = \frac{e^{\sum_{k=1}^{M}\vec{\lambda}^{(k)}\cdot\vec{\phi}^{(k)}(\vec{x})}}{Z[\vec{\lambda}^{(1)}, \ldots, \vec{\lambda}^{(M)}]}, \tag{1}$$

where $\vec{\lambda}^{(k)}$ labels the M potentials and $\vec{\phi}^{(k)}$ labels the M statistics. The potentials are chosen so that $< \vec{\phi}^{(k)}(\vec{x}) >= \vec{\psi}_{obs}^{(k)}$ for each k. (The choice of which filters should be used is prescribed by a feature selection stage based on a minimum entropy principle, which favors filters that lower the entropy of the distribution on images as much as possible.)

Estimating the value of the potentials $\vec{\lambda}$ is computationally very demanding. Standard procedures entail performing steepest descent on $\vec{\lambda}$, with stochastic sampling of the entire distribution $P(\vec{x}|\vec{\lambda})$ required at each iteration. The goal of this paper is to introduce the BKGIS algorithm for rapidly estimating the potentials based on Generalized Iterative Scaling and the Bethe-Kikuchi approximation.

3 Generalized Iterative Scaling

In this section we introduce Generalized Iterative Scaling (GIS)[3] and explain the connection between it and the multinomial approximation, which is an approximation described in the Appendix A that gives a rapid procedure for estimating potentials. GIS is an iterative procedure for calculating clique potentials that is guaranteed to converge to the maximum likelihood values of the potentials given the desired empirical filter marginals (e.g. filter histograms). We show that estimating the potentials by the multinomial approximation is equivalent to the estimate obtained after performing *the first iteration* of GIS.

The GIS procedure calculates a sequence of distributions on the entire image (and is guaranteed to converge to the correct maximum likelihood distribution), with an update rule given by $P^{(t+1)}(\vec{x}) \propto P^{(0)}(\vec{x}) \prod_{a=1}^{Q} \{\frac{\psi_a^{obs}}{\psi_a^{(t)}}\}^{\phi_a(\vec{x})}$, where $\psi_a^{(t)} =< \phi_a(\vec{x}) >_{P^{(t)}(\vec{x})}$ is the expected histogram for the distribution at time t. This implies that the corresponding clique potential update equation is given by:

$$\lambda_a^{(t+1)} = \lambda_a^{(t)} + \log \psi_a^{obs} - \log \psi_a^{(t)}. \tag{2}$$

We note that the GIS procedure exploits the fact that the filter histograms ψ_a^{obs} and $\psi_a^{(t)}$ are normalized to 1. If two or more filters are used it can be shown that the GIS update procedure must be modified to the following form:

$$\lambda_a^{k,(t+1)} = \lambda_a^{k,(t)} + (1/M)(\log \psi_a^{k,obs} - \log \psi_a^{k,(t)}) \tag{3}$$

for M filters, where the superscript k ranges from 1 through M and labels the filter.

If we initialize GIS so that the initial distribution is the uniform distribution on images, i.e. $P^{(0)}(\vec{x}) = L^{-N}$, then the distribution after one iteration is $P^{(1)}(\vec{x}) \propto e^{\sum_a \phi_a(\vec{x}) \log(\psi_a^{obs}/\alpha_a)}$, where the $\{\alpha_a\}$ components are the mean histogram under a distribution of uniformly random images. These can be computed efficiently, see Appendix A for details. In other words, the distribution after one iteration is the MEL distribution *with clique potential given by the multinomial approximation.*

4 The BKGIS Algorithm

We could iterate GIS to improve the estimate of the clique potentials beyond the accuracy of the multinomial approximation. Indeed, with sufficient number of iterations we would be guaranteed to converge to the correct potentials. The main difficulty lies in estimating $\psi_a^{(t)}$ for $t > 0$. At $t = 0$ this expectation is just the mean histogram with respect to the uniform distribution, α_a, which can be calculated efficiently as shown in Appendix A.

In this section, we define a new algorithm called BKGIS. This algorithm updates the potentials using equation (2) by approximating the expectations $\psi_a^{(t)}$ for $t >$) using a Bethe-Kikuchi approximation technique [9]. Our technique, which was inspired by the Unified Propagation and Scaling Algorithm [8], is described in two stages. Firstly, see ubsection (4.1), we derive the Bethe-Kikuchi free energy [9] for our 2-d image lattice, and simplify it using the shift invariance of the lattice (which enables the algorithm to run swiftly). Secondly, see subsection (4.2), we use the Convex-Concave Procedure (CCCP) [10] procedure to obtain an iterative update equation to estimate the histogram expectations.

4.1 The Bethe-Kikuchi Free Energy

We first define the Bethe-Kikuchi free energy and then apply it to the case of two filters, $\partial/\partial x$ and $\partial/\partial y$, on a 2-d image lattice. We write the MRF probability distribution on the image $\vec{x}$ in the form $P(\vec{x}) = \prod_{i,j} \Psi_{i,j}(x_i, x_j)/Z$, where the product $\prod_{i,j}$ is over all pairs of neighboring pixels x_i and x_j (with the restriction $i < j$ to avoid double-counting the interactions). This form is a general way of re-expressing the MEL distribution for filters whose support is two pixels. In other words, $f_i(\vec{x})$ is a function only of two pixels for each i, i.e. $f_i(\vec{x}) = f(x_i, x_j)$ where x_j is the appropriate neighbor of pixel x_i. The Bethe-Kikuchi free energy is

$$F = \sum_{i,j} \sum_{x_i,x_j} b_{ij}(x_i, x_j) \log \frac{b_{ij}(x_i, x_j)}{\Psi_{ij}(x_i, x_j)}$$

$$- \sum_i (q_i - 1) \sum_{x_i} b_i(x_i) \log b_i(x_i) \tag{4}$$

where $b_i(x_i)$ are the unary marginals on individual pixels, $b_{ij}(x_i, x_j)$ are the binary marginals on neighboring pairs of pixels and q_i is the number of pixels directly coupled to pixel i ($q_i = 4$ in the case of the filters $\partial/\partial x$ and $\partial/\partial y$ applied to a 2-d lattice).

Lagrange multipliers must be added to the Bethe-Kikuchi free energy to enforce the normalization of the unary marginals $b_i(x_i)$ and their consistency with the binary marginals $b_{i,j}(x_i, x_j)$. We write these constraints as follows:

$$\begin{aligned} &\sum_i \gamma_i (\sum_x b_i(x) - 1) \\ &+ \sum_{i,j} \sum_y \mu_{ij}(y) (\sum_x b_{i,j}(x,y) - b_j(y)) \\ &+ \sum_{i,j} \sum_x \mu_{ji}(x) (\sum_y b_{i,j}(x,y) - b_i(x)) \end{aligned} \tag{5}$$

where γ_i, $\mu_{i,j}(y)$ and $\mu_{j,i}(x)$ are Lagrange multipliers (and as before we restrict $i < j$).

When the Bethe-Kikuchi free energy is minimized with respect to the marginals $\{b_i(x_i)\}$ and $\{b_{ij}(x_i, x_j)\}$, such that the Lagrange multiplier constraints are satisfied, then the marginals approximate the true marginals of the distribution $P(\vec{x}) = \prod_{i,j} \Psi_{i,j}(x_i, x_j)/Z$, namely $\{P(x_i, x_j)\}$ and $\{P(x_i)\}$. The marginal estimates may then be used to calculate the histogram expectations required by the GIS update equation (2), since by shift invariance we have that $< \phi_a(\vec{x}) >= \sum_{x_i, x_j} P(x_i, x_j) \delta_{a, f_i(x_i, x_j)}$ for any i (and appropriate neighbor j).

We can discretize the expression for the x and y derivative filters as $x_{i+\Delta h} - x_i$ and $x_{i+\Delta v} - x_i$, respectively, where $x_{i+\Delta h}$ denotes the pixel just to the right of pixel x_i and $x_{i+\Delta v}$ denotes the pixel just above pixel x_i. Since these filters introduce only nearest-neighbor interactions on the lattice, we can group these interactions into horizontal and vertical interactions, and we can express the Bethe-Kikuchi free energy as

$$\begin{aligned} F = &\sum_i \sum_{x_i, x_{i+\Delta h}} b_{h_i}(x_i, x_{i+\Delta h}) \log \frac{b_{h_i}(x_i, x_{i+\Delta h})}{\Psi_{h_i}(x_i, x_{i+\Delta h})} \\ &+ \sum_i \sum_{x_i, x_{i+\Delta v}} b_{v_i}(x_i, x_{i+\Delta v}) \log \frac{b_{v_i}(x_i, x_{i+\Delta v})}{\Psi_{v_i}(x_i, x_{i+\Delta v})} \\ &- 3 \sum_i \sum_{x_i} b_i(x_i) \log b_i(x_i) \end{aligned} \tag{6}$$

plus the Lagrange multiplier terms.

Exploiting the shift invariance of the lattice we get that $b_i(x_i) = b(x_i)$, $b_{h_i}(x_i, x_{i+\Delta h}) = b_h(x_i, x_{i+\Delta h})$, $b_{v_i}(x_i, x_{i+\Delta v}) = b_v(x_i, x_{i+\Delta v})$, and similarly that $\psi_{h_i}(x_i, x_{i+\Delta h}) = \psi_h(x_i, x_{i+\Delta h})$, $\psi_{v_i}(x_i, x_{i+\Delta v}) = \psi_v(x_i, x_{i+\Delta v})$. As a result, the Bethe-Kikuchi free energy per pixel can be re-expressed as

$$\begin{aligned} F/N = \sum_{x,y} b_h(x,y) \log \frac{b_h(x,y)}{\Psi_h(x,y)} \\ + \sum_{x,y} b_v(x,y) \log \frac{b_v(x,y)}{\Psi_v(x,y)} - 3 \sum_x b(x) \log b(x) \end{aligned} \tag{7}$$

plus the following Lagrange multiplier terms:

$$\begin{aligned} \gamma(\sum_x b(x) - 1) + \sum_x \mu_{h,1}(x)(\sum_y b_h(x,y) - b(x)) \\ + \sum_y \mu_{h,2}(y)(\sum_x b_h(x,y) - b(y)) \end{aligned}$$

$$\begin{aligned} + \sum_x \mu_{v,1}(x)(\sum_y b_v(x,y) - b(x)) \\ + \sum_y \mu_{v,2}(y)(\sum_x b_v(x,y) - b(y)). \end{aligned} \tag{8}$$

4.2 CCCP Updates

We use the CCCP procedure [10] to write simple update equations which will lower the Bethe-Kikuchi free energy monotonically, and hence allow us to estimate the marginals $\psi_a^{(t)}$.

We express the Bethe-Kikuchi free energy equation (7) plus Lagrange multiplier terms in equation (8) as a sum of a concave energy function E_{ave} and a convex energy function E_{vex}:

$$E_{ave} = -4 \sum_x b(x) \log b(x), \tag{9}$$

$$E_{vex} = \sum_{x,y} b_h(x,y) \log \frac{b_h(x,y)}{\Psi_h(x,y)}$$
$$+ \sum_{x,y} b_v(x,y) \log \frac{b_v(x,y)}{\Psi_v(x,y)} + \sum_x b(x) \log b(x)$$
$$+\text{Lagrange constraints.} \tag{10}$$

The CCCP update equation is given by:

$$\vec{\nabla} E_{vex}(\vec{z}^{(t+1)}) = -\vec{\nabla} E_{ave}(\vec{z}^{(t)}) \tag{11}$$

where $\vec{z}$ denotes the vector of all marginals $b_h(.,.)$, $b_v(.,.)$, $b(.)$, such that the Lagrange multiplier constraints are satisfied at each time t. This update equation gives rise to a "double-loop" algorithm consisting of an outer loop, in which the marginals are updated as a function of their previous values and of the Lagrange multipliers, and an inner loop, in which the Lagrange multipliers are updated.

The outer loop updates are obtained directly from equation (11):

$$b^{(t+1)}(x) = e^3 [b^{(t)}(x)]^4 e^{-\gamma + \mu_{h1}(x) + \mu_{h2}(x) + \mu_{v1}(x) + \mu_{v2}(x)} \tag{12}$$

$$b_h^{(t+1)}(x,y) = \psi_h(x,y) e^{-1} e^{-\mu_{h1}(x) - \mu_{h2}(y)} \tag{13}$$

and

$$b_v^{(t+1)}(x,y) = \psi_v(x,y) e^{-1} e^{-\mu_{v1}(x) - \mu_{v2}(y)} \tag{14}$$

The inner loop equations update one Lagrange multiplier at a time so as to properly impose the Lagrange constraints (several iterations of the inner loop may be required to satisfy the constraints). They are obtained by writing the constraint conditions in terms of the above outer loop expressions for the marginal updates. The γ update is obtained from the condition $\sum_x b(x) = 1$:

$$e^{\gamma^{new}} = \sum_x e^3 [b(x)]^4 e^{\mu_{h1}(x) + \mu_{h2}(x) + \mu_{v1}(x) + \mu_{v2}(x)} \tag{15}$$

where γ^{new} is shorthand for $\gamma^{(\tau+1)}$, the Lagrange multipliers on the right-hand side are the values at time τ, and $b(x)$ is shorthand for $b^{(t)}(x)$. (The time superscript τ counts updates within the inner loop, while the superscript t counts updates within the outer loop.)

The marginal consistency constraint $\sum_y b_h(x,y) = b(x)$ yields the update for μ_{h1}:

$$e^{2\mu_{h1}^{new}(x)} = e^{2\mu_{h1}(x)} \frac{\psi_h(x,y) e^{-1-\mu_{h1}(x)-\mu_{h2}(y)}}{e^3 [b(x)]^4 e^{-\gamma} e^{\mu_{h1}(x)+\mu_{h2}(x)+\mu_{v1}(x)+\mu_{v2}(x)}} \tag{16}$$

(The same superscript convention is used as before.) Similarly, the marginal consistency constraints $\sum_y b_v(x,y) = b(x)$, $\sum_x b_h(x,y) = b(y)$ and $\sum_x b_v(x,y) = b(y)$, respectively, give rise to the other update equations:

$$e^{2\mu_{v1}^{new}(x)} = e^{2\mu_{v1}(x)} \frac{\psi_v(x,y) e^{-1-\mu_{v1}(x)-\mu_{v2}(y)}}{e^3 [b(x)]^4 e^{-\gamma} e^{\mu_{h1}(x)+\mu_{h2}(x)+\mu_{v1}(x)+\mu_{v2}(x)}} \tag{17}$$

$$e^{2\mu_{h2}^{new}(y)} = e^{2\mu_{h2}(x)} \frac{\psi_h(x,y) e^{-1-\mu_{h1}(x)-\mu_{h2}(y)}}{e^3 [b(y)]^4 e^{-\gamma} e^{\mu_{h1}(y)+\mu_{h2}(y)+\mu_{v1}(y)+\mu_{v2}(y)}} \tag{18}$$

$$e^{2\mu_{v2}^{new}(y)} = e^{2\mu_{v2}(x)} \frac{\psi_v(x,y) e^{-1-\mu_{v1}(x)-\mu_{v2}(y)}}{e^3 [b(y)]^4 e^{-\gamma} e^{\mu_{h1}(y)+\mu_{h2}(y)+\mu_{v1}(y)+\mu_{v2}(y)}}. \tag{19}$$

Note that the Lagrange multiplier update equations above are calculated sequentially rather than in parallel (convergence is not guaranteed for paralled updates). Several iterations (empirically, about five to ten suffice for our application) of the inner loop are performed to update the Lagrange multipliers, followed by one iteration of the outer loop equations to estimate the marginals. This procedure is repeated until convergence is obtained (empirically, about fifteen repetitions suffice).

5 Results

The BKGIS algorithm, as defined above, is slightly unstable because the histogram expectations are estimated only approximately. To circumvent this problem, we modified the basic GIS update equation so that it makes more conservative updates and avoids instabilities:

$$\lambda_a^{k,(t+1)} = \lambda_a^{k,(t)} + \beta(1/M)(\log \psi_a^{k,obs} - \log \psi_a^{k,(t)}) \tag{20}$$

where $\beta < 1$ is a coefficient that sets the scale of the update ($\beta = 1$ corresponds to standard GIS). We chose $\beta = 0.2$ as a compromise between stability and speed of convergence.

To test the modified BKGIS algorithm on the MEL distribution with the $\partial/\partial x$ and $\partial/\partial y$ filters, we initialized the potentials $\vec{\lambda}^{(x)}$ and $\vec{\lambda}^{(y)}$ to the values given by the multinomial approximation (see Appendix). The modified GIS update equations converged (though with some oscillation) after about 15 iterations (using 10 outer loop iterations and 10 inner loop iterations in CCCP). The algorithm took about thirty seconds (on a Pentium 400 MHz PC) to complete its iterations.

Figure (1) shows the empirical $\partial/\partial x$ filter response marginal obtained from natural images (taken from the Sowerby database of rural images), the potential determined by the multinomial approximation, and the error between the empirical marginal and the marginal obtained using the multinomial potential (estimated using the Bethe-Kikuchi method). Note that the long, heavy tails characteristic of filter marginals obtained from natural images are obscured here due to the coarse quantization of filter response values a. (Results are almost identical for the $\partial/\partial y$ filter.) The multinomial potentials correspond to the first iteration of BKGIS when it is initialized with uniform potentials.

In our experiments we initialized BKGIS with the multinomial potentials. After running the algorithm, the potential was only slightly changed, but the error was reduced, as shown in Figure (2). (Note that the vertical axis scale is about an order of magnitude larger in the first figure relative to the second.)

The potentials obtained by BKGIS were used to generate samples by an MCMC sampler, which are shown in Figure (3).

We point out that our error measure compares the observed marginals with the marginals estimated using the Bethe-Kikuchi approximation. But there are more accurate error measures for determining how well we have estimated the clique potentials. Observe that our error measure involves two types of approximations. Firstly, the BKGIS algorithm makes

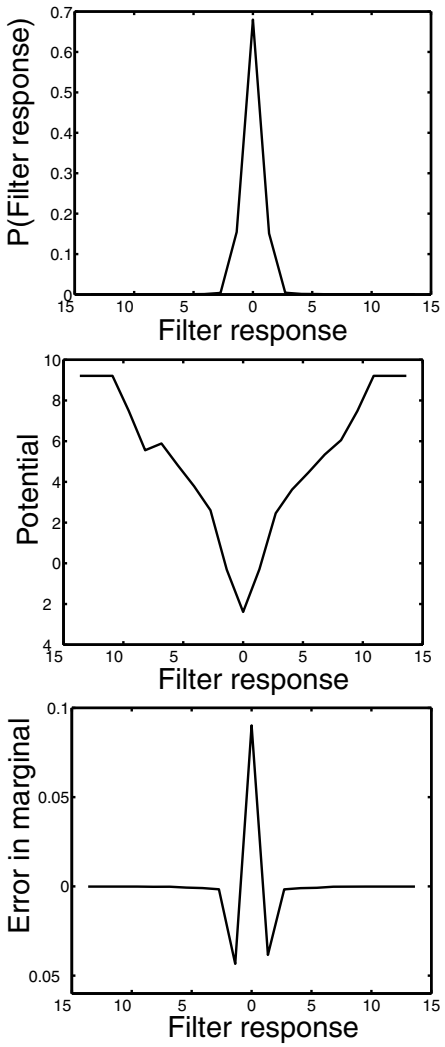

Fig. 1. Top panel, observed marginal for $\partial/\partial x$ filter response values obtained from natural images. (Horizontal axis indicates value of $\partial/\partial x$ filter response, vertical axis indicates empirical probability.) Middle panel, potential determined by multinomial approximation, shown as $-\lambda_a^{(x)}$ along vertical axis with $a =$ filter response of $\partial/\partial x$ filter displayed along horizontal axis. Bottom panel displays error, i.e. difference between observed marginal and the marginal obtained using the multinomial potential, along vertical axis, as a function of a along horizontal axis.

use of the Bethe-Kikuchi approximation for estimating the potentials. Secondly, we used the Bethe-Kikuchi approximation to estimate the marginals when evaluating the performance of the algorithm. Alternatively, we could evaluate the potentials calculated by BKGIS by estimating the marginals by MCMC and comparing them to the observed marginals. This, however, requires running the MCMC algorithm for a long time to get accurate estimations.

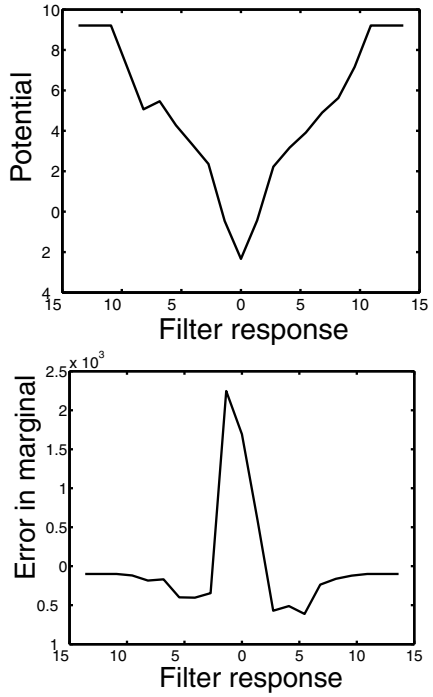

Fig. 2. Top panel, potential obtained from BKGIS. Bottom panel shows error, which is smaller than than that obtained using multinomial potential (previous figure). (Same axes as in the last two panels of the previous figure, except that the vertical scale for the error is smaller than the version in the previous figure.)

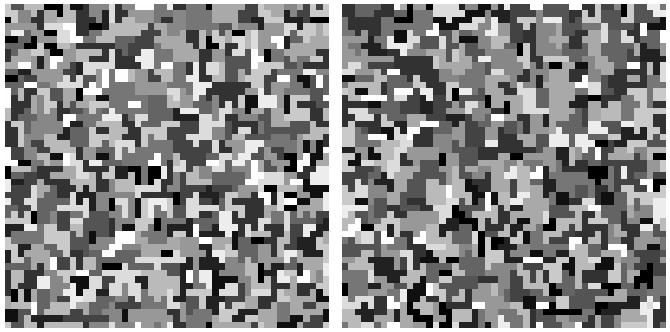

Fig. 3. Stochastic image samples, obtained by an MCMC algorithm, of the MRF distribution whose potentials have been determined by BKGIS.

6 Discussion

This paper applied the Generalized Iterative Scaling algorithm [3] to the problem of learning MRF potentials [12] from image histograms. We introduced a new algorithm called BKGIS which used a Bethe-Kikuchi approximation [4],[9] and a CCCP algorithm [10] to speed up a crucial stage of the GIS algorithm. In addition, we demonstrated that the first iteration of

GIS, or BKGIS, corresponds to an approximate method for estimating potentials [2].

We note that our method can be generalized to apply to any set of filters (of arbitrary form, linear or non-linear) by using higher-order Kikuchi [9] approximations. (The Bethe approximation is directly applicable only for filters whose support size is two pixels.) However, this generalization comes at the cost of a computational complexity that increases exponentially in the support size of the filters. An alternative scheme is to introduce auxiliary variables that allow the use of the Bethe approximation for any size filters, with a computational cost that is linear in the size of the filters. It remains to be seen if this technique is practical, both in terms of computational complexity and the quality of the underlying Bethe approximation. Finally, we point out that our method is formulated for MRF's defined in terms of filter marginals, and it is straightforward to extend the method to handle joint histograms of two or more filters. (However, is unclear how to extend the method to distributions defined in terms of *moments* of filters.)

Acknowledgments

We would like to thank Jonathan Yedida for helpful email correspondence. This work was supported by the National Institute of Health (NEI) with grant number RO1-EY 12691-01.

Appendix A: The g-Factor and the Multinomial Approximation

This section defines the g-factor function, which was introduced in earlier work [1,2] to establish connections between the distribution on images, $P(\vec{x}|\vec{\lambda})$, with the corresponding distribution induced on features. The g-factor also motivated the *multinomial approximation*, which was used to determine an efficient procedure for approximating the potentials.

The g-factor $g(\vec{\psi})$ is defined as follows:

$$g(\vec{\psi}) = \sum_{\vec{x}} \delta_{\vec{\phi}(\vec{x}),\vec{\psi}}. \tag{21}$$

Here L is the number of grayscale levels of each pixel, so that L^N is the total number of possible images. The g-factor is essentially a combinational factor which counts the number of ways that one can obtain statistics $\vec{\psi}$, see figure (4). Equivalently, we can define an associated distribution $\hat{P}_0(\vec{\psi}) = \frac{1}{L^N} g(\vec{\psi})$, which is the default distribution on $\vec{\psi}$ if the images are generated by uniform noise (i.e. completely random images).

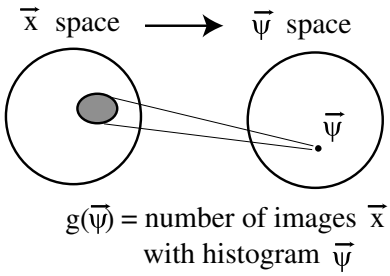

Fig. 4. The g-factor $g(\vec{\psi})$ counts the number of images $\vec{x}$ that have statistics $\vec{\psi}$. Note that the g-factor depends only on the choice of filters and is independent of the training image data.

We can use the g-factor to compute the induced distribution $\hat{P}(\vec{\psi}|\vec{\lambda})$ on the statistics determined by MEL:

$$\hat{P}(\vec{\psi}|\vec{\lambda}) = \sum_{\vec{x}} \delta_{\vec{\psi},\vec{\phi}(\vec{x})} P(\vec{x}|\vec{\lambda}) = \frac{g(\vec{\psi}) e^{\vec{\lambda}\cdot\vec{\psi}}}{Z[\vec{\lambda}]} \tag{22}$$

where the partition function is:

$$Z[\vec{\lambda}] = \sum_{\vec{\psi}} g(\vec{\psi}) e^{\vec{\lambda}\cdot\vec{\psi}}. \tag{23}$$

Observe that both $\hat{P}(\vec{\psi}|\vec{\lambda})$ and $\log Z[\vec{\lambda}]$ are sufficient for computing the parameters $\vec{\lambda}$. The $\vec{\lambda}$ can be found by solving either of the following two (equivalent) equations: $\sum_{\vec{\psi}} \hat{P}(\vec{\psi}|\vec{\lambda})\vec{\psi} = \vec{\psi}_{obs}$, or $\frac{\partial \log Z[\vec{\lambda}]}{\partial \vec{\lambda}} = \vec{\psi}_{obs}$, which shows that *knowledge of the g-factor and* $e^{\vec{\lambda}\cdot\vec{\psi}}$ *are all that is required to do MEL.*

Observe from equation (22) that we have $\hat{P}(\vec{\psi}|\vec{\lambda} = 0) = P_0(\vec{\psi})$. In other words, setting $\vec{\lambda} = 0$ corresponds to a uniform distribution on the images $\vec{x}$.

The Multinomial Approximation

We now consider the case where the statistic is a single histogram. Our results, of course, can be directly extended to multiple histograms. We describe the multinomial approximation of the g-factor and discuss the procedure it prescribes for estimating the potentials.

We rescale the $\vec{\lambda}$ variables by N so that we have:

$$P(\vec{x}|\lambda) = \frac{e^{N\vec{\lambda}\cdot\vec{\phi}(\vec{x})}}{Z[\vec{\lambda}]}, \quad \hat{P}(\vec{\psi}|\lambda) = g(\vec{\psi})\frac{e^{N\vec{\lambda}\cdot\vec{\psi}}}{Z[\vec{\lambda}]}, \tag{24}$$

We now consider the approximation that the filter responses $\{f_i\}$ are *independent of each other when the images are uniformly distributed*. This is the *multinomial approximation*. It implies that we can express the g-factor as being proportional to a multinomial distribution:

$$g(\vec{\psi}) = L^N \frac{N!}{(N\psi_1)!...(N\psi_Q)!}\alpha_1^{N\psi_1}...\alpha_Q^{N\psi_Q} \tag{25}$$

where $\sum_{a=1}^{Q}\psi_a = 1$ (by definition) and the $\{\alpha_a\}$ are the means of the components $\{\psi_a\}$ with respect to the distribution $\hat{P}_0(\vec{\psi})$. As we will describe later, the $\{\alpha_a\}$ will be determined by the filters $\{f_i\}$. See Coughlan and Yuille, in preparation, for details of how to compute the $\{\alpha_a\}$.

This approximation enables us to calculate MEL *analytically*.

Theorem *With the multinomial approximation the log partition function is:*

$$\log Z[\vec{\lambda}] = N\log L + N\log\{\sum_{a=1}^{Q} e^{\lambda_a + \log\alpha_a}\}, \tag{26}$$

and the "potentials" $\{\lambda_a\}$ can be solved in terms of the observed data $\{\psi_{obs,a}\}$ to be:

$$\lambda_a = \log\frac{\psi_{obs,a}}{\alpha_a}, \quad a = 1, ..., Q. \tag{27}$$

We note that there is an ambiguity $\lambda_a \mapsto \lambda_a + K$ where K is an arbitrary number (recall that $\sum_{a=1}^{Q}\psi(a) = 1$). We fix this ambiguity by setting $\vec{\lambda} = 0$ if $\vec{\alpha} = \vec{\psi}_{obs}$.

Proof. *Direct calculation, using the fact that $\frac{\partial \log Z[\vec{\lambda}]}{\partial\vec{\lambda}} = \vec{\psi}_{obs}$.*

Our simulation results show that this simple approximation gives the typical potential forms generated by Markov Chain Monte Carlo (MCMC) algorithms for Minimax Entropy Learning.

Appendix B

Computing the Mean and Covariance of $g(\vec{\phi})$

We need to calculate the mean and covariance of $g(\vec{\phi})$ in order to estimate the vector $\vec{\lambda}$. In this section we present an exact method for calculating the mean and covariance that requires only knowledge of the combinatorics of individual and pairwise filter responses, without needing to enumerate all possible images $\mathbf{I}$ or statistics $\vec{\phi}$.

As before, we will treat $g(\vec{\phi})$ as an un-normalized probability distribution derived from a uniform distribution $P_g(\mathbf{I})$ on the set of all possible images. All expectations $< . >_g$ will be taken with respect to the uniform distribution $P_g(\mathbf{I})$.

Case I: a single filter

First we consider a single filter. The mean is $\vec{c} =< \vec{\phi}(\mathbf{I}) >_g = \frac{1}{N} < \sum_{\mathbf{x}} \vec{b}_{\mathbf{x}}(\mathbf{I}) >_g$, and by translation invariance this becomes $\vec{c} =< \vec{b}_{\mathbf{x}}(\mathbf{I}) >_g$ for any $\mathbf{x}$. Equivalently, $\vec{c} = \sum_{\vec{b}_{\mathbf{x}}(\mathbf{I})} P_g(\vec{b}_{\mathbf{x}}(\mathbf{I}))\vec{b}_{\mathbf{x}}(\mathbf{I})$, i.e. the mean is an average over all possible filter responses at $\mathbf{x}$. (A technique for computing $P_g(\vec{b}_{\mathbf{x}}(\mathbf{I}))$ is presented in the next section.)

The covariance is defined as $C =< (\vec{\phi}(\mathbf{I}) - \vec{c})(\vec{\phi}(\mathbf{I}) - \vec{c})^T >_g = \frac{1}{N^2} < \sum_{\mathbf{x}} (\vec{b}_{\mathbf{x}}(\mathbf{I}) - \vec{c}) \sum_{\mathbf{y}} (\vec{b}_{\mathbf{y}}(\mathbf{I}) - \vec{c})^T >_g$, where $\vec{\phi}(\mathbf{I})$ is defined in terms of the binary representations $\vec{b}^{(i)}_{\mathbf{x}}(\mathbf{I})$ (introduced in Section 3). Dropping the image argument $\mathbf{I}$ for simplicity, we get $C = \frac{1}{N^2} \sum_{\{\vec{b}_{\mathbf{x}}\}, \forall \mathbf{x}} P_g(\{\vec{b}_{\mathbf{x}}\}, \forall \mathbf{x}) \sum_{\mathbf{x}} (\vec{b}_{\mathbf{x}} - \vec{c}) \sum_{\mathbf{y}} (\vec{b}_{\mathbf{y}} - \vec{c})^T$ where the expression $\{\vec{b}_{\mathbf{x}}\}, \forall \mathbf{x}$ denotes the set of filter responses at every pixel (expressed in the histogram representation $\vec{b}_{\mathbf{x}}$). The sums over pixel locations $\mathbf{x}$ and $\mathbf{y}$ can be divided into two categories: pairs of pixels $\mathbf{x}$ and $\mathbf{y}$ whose filter responses are independent of

each other (i.e. the pixels are far enough apart that the kernel centered at $\mathbf{x}$ does not overlap with the kernel centered at $\mathbf{y}$), and those whose filter responses are correlated because of kernel overlap. For each pixel $\mathbf{x}_0$ we define the neighborhood $N(\mathbf{x}_0)$ to be the set of pixels whose kernel overlaps with the kernel of $\mathbf{x}_0$.

Now C may be written

$$C = \frac{1}{N^2} \sum_{\{\vec{b}_{\mathbf{x}}\},\forall \mathbf{x}} P(\{\vec{b}_{\mathbf{x}}\},\forall \mathbf{x})[\sum_{\mathbf{x}_0} \sum_{\mathbf{y}\in N(\mathbf{x}_0)} (\vec{b}_{\mathbf{x}_0} - \vec{c})(\vec{b}_{\mathbf{y}} - \vec{c})^T + \sum_{\mathbf{x}_0} \sum_{\mathbf{y}\notin N(\mathbf{x}_0)} (\vec{b}_{\mathbf{x}_0} - \vec{c})(\vec{b}_{\mathbf{y}} - \vec{c})^T].$$

We will show that the second (non-overlap) term vanishes. For each $\mathbf{x}_0$ and $\mathbf{y}$, we can sum over $\vec{b}_{\mathbf{x}}$ for every pixel but $\mathbf{x}_0$ and $\mathbf{y}$. All that remains is a sum over $\vec{b}_{\mathbf{x}_0}$ and $\vec{b}_{\mathbf{y}}$, but $P_g(\vec{b}_{\mathbf{x}_0}, \vec{b}_{\mathbf{y}}) = P_g(\vec{b}_{\mathbf{x}_0}) P_g(\vec{b}_{\mathbf{y}})$ and the expression drops out since $< \vec{b}_{\mathbf{x}_0} >_g = < \vec{b}_{\mathbf{y}} >_g = \vec{c}$. Thus we have $C = \frac{1}{N^2} \sum_{\mathbf{x}_0} \sum_{\{\vec{b}_{\mathbf{x}}\},\forall \mathbf{x}} P_g(\{\vec{b}_{\mathbf{x}}\},\forall \mathbf{x}) \sum_{\mathbf{y}\in N(\mathbf{x}_0)} (\vec{b}_{\mathbf{x}_0} - \vec{c})(\vec{b}_{\mathbf{y}} - \vec{c})^T$.

Now, because of translation invariance, each term in the sum over $\mathbf{x}_0$ is equivalent. Thus we can choose any $\mathbf{x}_0$ for convenience and rewrite the covariance as

$$C = N \frac{1}{N^2} \sum_{\{\vec{b}_{\mathbf{x}}\}, \mathbf{x}\in N(\mathbf{x}_0)} P(\{\vec{b}_{\mathbf{x}}\}, \mathbf{x} \in N(\mathbf{x}_0)) \sum_{\mathbf{y}\in N(\mathbf{x}_0)} (\vec{b}_{\mathbf{x}_0} - \vec{c})(\vec{b}_{\mathbf{y}} - \vec{c})^T$$

$$= \frac{1}{N} \sum_{\mathbf{y}\in N(\mathbf{x}_0)} \sum_{\vec{b}_{\mathbf{x}_0}} \sum_{\vec{b}_{\mathbf{y}}} P(\vec{b}_{\mathbf{x}_0}, \vec{b}_{\mathbf{y}}) (\vec{b}_{\mathbf{x}_0} - \vec{c})(\vec{b}_{\mathbf{y}} - \vec{c})^T.$$

(A technique for computing $P_g(\vec{b}_{\mathbf{x}_0}, \vec{b}_{\mathbf{y}})$ is shown in the next section.)

Computing Probabilities of Single and Pairwise Filter Responses

The calculations of the mean and covariance of $g(\vec{\phi})$ require explicit knowledge of $P_g(f_{\mathbf{x}}^{(i)})$ and $P_g(f_{\mathbf{x}}^{(i)}, f_{\mathbf{y}}^{(j)})$, i.e. the fraction of all possible images that produce a filter response $f_{\mathbf{x}}^{(i)}$ or joint filter responses $f_{\mathbf{x}}^{(i)}$ and $f_{\mathbf{y}}^{(j)}$ (these responses are correlated when the kernel of the first overlaps the kernel of the second). In this section we present an efficient method of computing these filter response probabilities that applies to a large class of filters: any non-linear scalar function of a linear filter whose kernel contains integer coefficients. Many filters used in computer vision, image analysis and texture analysis are included in this category,

such as $\partial I/\partial x, |\nabla I|, \nabla^2 I$, and $G \star I$.

Response of one linear filter

First we consider the case of a single linear filter with kernel k: $f_{\mathbf{x}}(\mathbf{I}) = (k \star I)(\mathbf{x})$, where the coefficients of k are integers. The filter response at any arbitrary image location $\mathbf{x}$, which we will write simply as f, is a linear combination of those pixel intensities lying within the kernel: $f = \sum_{\mathbf{x}'} k(\mathbf{x}')I(\mathbf{x} - \mathbf{x}')$. For combinatoric purposes we can regard the pixel intensities as i.i.d. variables drawn from a uniform distribution on $\{0, 1, \ldots, Q-1\}$ (i.e. all pixel intensities 0 through $Q-1$ are equally likely). For each pixel location $\mathbf{x}'$ within the kernel, the quantity $v(\mathbf{x}') = k(\mathbf{x}')I(\mathbf{x} - \mathbf{x}')$ is thus a random variable with distribution $P_g(v(\mathbf{x}')) = (1/Q)\sum_{q=0}^{Q-1} \delta_{qk(\mathbf{x}'),v(\mathbf{x}')}$, i.e. $v(\mathbf{x}')$ may assume any of the values $\{0, k(\mathbf{x}'), 2k(\mathbf{x}'), \ldots, (Q-1)k(\mathbf{x}')\}$ with equal probability, and no other values are possible.

Since the quantities $v(\mathbf{x}')$ are independent from one $\mathbf{x}'$ to the next, the probability of any filter response $f = \sum_{\mathbf{x}'} v(\mathbf{x}')$ equals the convolution of the probabilities on each $v(\mathbf{x}')$:

$$P_g(f) = P_g(v(\mathbf{x}'_1)) \star P_g(v(\mathbf{x}'_2)) \star \ldots \star P_g(v(\mathbf{x}'_K)),$$

where $\mathbf{x}'_1$ through $\mathbf{x}'_{S'}$ are the K pixels that lie in the kernel. These convolutions can be performed numerically using vectors of length $Q|k(\mathbf{x}')| + 1$ to represent $P_g(v(\mathbf{x}'))$, yielding a vector of length f_{max} (i.e. the maximum possible filter response) to represent $P_g(f)$.

Non-linear generalization

We can generalize the calculation of $P_g(\mathbf{f})$ (or the scalar case $P_g(f)$, which may be regarded as a special case of this pair-wise response probability) to the case where the filter responses are defined as non-linear functions of linear filter responses: $\mathbf{f} = \mathbf{h}(\mathbf{f}_L)$, where $\mathbf{f}_L$ is obtained by linear convolution of $\mathbf{I}$ with $\mathbf{k}$, and $\mathbf{h}(.)$ is a non-linear vector function. The probability we seek, $P_g(\mathbf{f})$, is induced by the probability we already know how to calculate, $P_g(\mathbf{f}_L)$: $P_g(\mathbf{f}) = \sum_{\mathbf{f}_L} P_g(\mathbf{f}_L)\delta_{\mathbf{h}(\mathbf{f}_L),\mathbf{f}}$. This sum is straightforward to calculate numerically: initialize $P_g(\mathbf{f})$ to 0 everywhere, and for each $\mathbf{f}_L$ calculate $\mathbf{f} = \mathbf{h}(\mathbf{f}_L)$ and add $P_g(\mathbf{f}_L)$ to $P_g(\mathbf{f})$.

References

[1] Coughlan, J M and Yuille, A L. A Phase Space Approach to Minimax Entropy Learning and The Minutemax approximation. In *Proceedings NIPS'98*. (1998) 761-767.

[2] Coughlan, J M and Yuille, A L. The g Factor: Relating Distributions on Features to Distributions on Images. In Proceedings NIPS'01. (2002). To appear.

[3] Darroch, J N and Ratcliff, D. Generalized Iterative Scaling for Log-Linear Models. The Annals of Mathematical Statistics. Vol. 43, No. 5. (1972) 1470-1480.

[4] Domb, C and Green, M S (Eds). Phase Transitions and Critical Phenomena. Vol. 2. Academic Press. London. (1972).

[5] Konishi, S M, Yuille, A L, Coughlan, J M and Zhu, S C. Fundamental Bounds on Edge Detection: An Information Theoretic Evaluation of Different Edge Cues. In Proceedings Computer Vision and Pattern Recognition CVPR'99. Fort Collins, Colorado. (1999) 573-579.

[6] Lee, A B, Mumford, D B and Huang, J. Occlusion Models of Natural Images: A Statistical Study of a Scale-Invariant Dead Leaf Model. International Journal of Computer Vision. Vol. 41, No.'s 1/2. (2001) 35-59.

[7] Portilla, J and Simoncelli, E P. Parametric Texture Model based on Joint Statistics of Complex Wavelet Coefficients. International Journal of Computer Vision. (2000) 49-71.

[8] Teh, Y W and Welling, M. The Unified Propagation and Scaling Algorithm. In Proceedings NIPS'01. (2002). To Appear.

[9] Yedidia, J S, Freeman, W T and Weiss, Y. Generalized Belief Propagation. In Proceedings NIPS'00. (2000) 698-695.

[10] Yuille, A L and Rangarajan, A. The Concave-Convex Procedure (CCCP). In Proceedings NIPS 2001. Vancouver, Canada. (2002). To appear.

[11] Zhu, S.C. and Mumford, D. Prior Learning and Gibbs Reaction-Diffusion. PAMI vol.19, no.11. (1997) 1236-1250.

[12] Zhu, S C, Wu, Y N and Mumford, D B. Minimax Entropy Principle and Its Application to Texture Modeling. Neural Computation. Vol. 9. no. 8. (1997) 1627-1660.