TABLE I
ESTIMATED VALUES OF P USING (7)–(9) FOR VARIOUS SNR AND THRESHOLD T, TRUE $P = 4$

$p = 4$	SNR			
T	3dB	10dB	20dB	30dB
0.4	1.8	1.6	2.8	2.4
0.8	2.8	3.0	3.2	2.2
1.2	3.4	3.4	3.2	2.0
1.6	3.6	3.4	3.0	1.8

TABLE II
ESTIMATED VALUES OF P USING (7)–(9) FOR VARIOUS SNR AND THRESHOLD T, TRUE $P = 8$

$p = 8$	SNR			
T	3dB	10dB	20dB	30dB
0.4	2.0	1.4	3.4	1.6
0.8	3.6	3.8	4.6	1.2
1.2	5.2	4.8	4.2	1.2
1.6	5.6	5.0	3.8	1.2

extracting a representative noise sample $\hat{n}$, which is not dominated by image components. There is also a bias in the estimates, but for this image, a threshold level between 1.2 and 1.6 provides the best performance. Considering the implications of Fig. 3, this bias is not serious for p larger than about 3.

Another important consideration is whether improvement is possible when the noise distribution is not a member of the gpG family. The following example illustrates that flexibility provided by the shape parameter enables the gpG model to approximate the noise distribution. In this case, the noise was distributed χ^2 (Chi squared) with three degrees of freedom, which is neither gpG nor symmetric. Noise data was scaled such that the resulting corrupted image had a 15-dB SNR. The estimated value for p using the method outlined in Section III-A is 1.2. Fig. 4(a) shows the blurred image with χ^2 noise, whereas Fig. 4(b) shows the restored image. Fig. 5 presents a comparison of the error curves for the adaptive restoration and least squares. Note that the minimum error point for $p = 1.2$ is significantly below that of $p = 2$. The encouraging observation here is that even in the presence of noise clearly not well modeled by a gpG distribution, the adaptive approach offers an improvement over methods that implicitly assume a strict Gaussian model.

REFERENCES

[1] B. R. Hunt, "The application of constrained least squares estimation to image restoration by digital computer," *IEEE Trans. Comput.*, vol. C-22, no. 9, Sept. 1973.

[2] S. F. Gull and J. Skilling, "Maximum entropy method in image processing," *Proc. IEEE*, vol. 131, Pt. F, no. 6, pp. 646–659, 1984.

[3] Y. Censor, "Finite series-expansion reconstruction methods," *Proc. IEEE*, vol. 71, pp. 409–418, 1983.

[4] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[5] J. B. McDonald, "Partially adaptive estimation of ARMA time series models," *Int. J. Forecasting*, vol. 5, pp. 217–230, 1989.

[6] T. T. Pham and R. J. P. deFigueiredo, "Maximum likelihood estimation of a class of non-Gaussian densities with application to l_p deconvolution," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 1, pp. 73–82, Jan. 1989.

[7] R. L. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images*. Boston: Kluwer, 1991.

[8] C. Bouman and K. Sauer, "A generalized Gaussian model for edge-preserving MAP estimation," *IEEE Trans. Image Processing*, vol. 2, pp. 296–310, July 1993.

[9] S. A. Kassam, *Signal Detection in Non-Gaussian Noise*. New York: Springer-Verlag, 1988.

[10] A. K. Katsaggelos, "Iterative image restoration algorithms," *Opt. Eng.*, vol. 28, no. 7, pp. 735–748, July 1989.

[11] M. E. Zervakis and T. M. Kwon, "Robust estimation techniques in regularized image restoration," *Opt. Eng.*, vol. 31, no. 10, pp. 2174–2192, Oct. 1992.

Scalable Data Parallel Algorithms for Texture Synthesis Using Gibbs Random Fields

David A. Bader, Joseph JáJá, and Rama Chellappa

Abstract—**This correspondence introduces scalable data parallel algorithms for image processing. Focusing on Gibbs and Markov random field model representation for textures, we present parallel algorithms for texture synthesis, compression, and maximum likelihood parameter estimation, currently implemented on Thinking Machines CM-2 and CM-5. Use of fine-grained, data parallel processing techniques yields real-time algorithms for texture synthesis and compression that are substantially faster than the previously known sequential implementations. Although current implementations are on Connection Machines, the methodology presented here enables machine-independent scalable algorithms for a number of problems in image processing and analysis.**

I. INTRODUCTION

Random fields have been successfully used to sample and synthesize textured images [4]–[7], [9]. Texture analysis has applications in image segmentation and classification, biomedical image analysis, and automatic detection of surface defects. Of particular interest are the models that specify the statistical dependence of the gray level at a pixel on those of its neighborhood. There are several well-known algorithms describing the sampling process for generating synthetic textured images and algorithms that yield an estimate of the parameters of the assumed random process given a textured image. Impressive results related to real-world imagery have appeared in the literature [3], [5]–[8]. However, all these algorithms are quite computationally demanding because they typically require on the order of Gn^2 arithmetic operations per iteration for an image of size

Manuscript received October 9, 1993; revised December 17, 1994. This work was supported by NASA Graduate Student Researcher Fellowship NGT-50951, by the National Science Foundation under Grant CCR-9103135 and NSF HPCC/GCAG Grant BIR-9318183, and by the U.S. Air Force under Grant F49620-92-J0130. The associate editor coordinating the review of this paper and approving it for publication was Dr. Hsueh-Ming Hang.

The authors are with the Department of Electrical Engineering and the Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 USA.

IEEE Log Number 9413842.

Algorithm 1 *Direct Gaussian MRF Sampler*
Reconstruct a GMRF texture from parameters, assuming toroidal wrap-around and an M^2 image size
Input:
$\Theta \leftarrow$ the set of parameters for the given model.
$\{G\}$ is the number of grey levels.
image $\leftarrow$ a parallel variable for the image. (Complex)
$\{\Phi_r\} \leftarrow$ a parallel variable with serial elements for each parameter in the model.
begin
1. Initialize the real part of the image in parallel to Gaussian noise with mean = 0 and standard deviation = 1.
2. Initialize the imaginary part of the image in parallel to 0.
3. Divide the image by $\sqrt{\nu}$.
4. Perform a parallel, in-place FFT on the noise.
5. **For all** pixels σ **in parallel do**:
   5.1 **For each** $r \in N_s$, $\Phi_r = \cos \frac{2\pi}{M}\sigma^t r$
   5.2 Calculate μ_σ from Equation (4).
   5.3 Divide the image by $\sqrt{\mu_\sigma}$.
6. Perform a parallel, in-place, inverse FFT on the image.
7. Scale the result to grey levels in the interval [0..G-1].

end

Fig. 1. Gaussian MRF sampler algorithm.

Algorithm 2 *Least Squares Estimator for GMRF*
Using the method of Least Squares, estimate the parameters of image Y.
Assume toroidal wrap-around, an M^2 image size, and a given neighborhood.
Input:
$\{Y\} \leftarrow$ the image.
$\Theta \leftarrow$ the scalar array of parameter estimates for each neighborhood element.
begin
1. **For all** pixels **in parallel do:**
   1.1 **For each** $r \in N_s$ **do**
      1.1.1 $\mathbf{g}_\sigma[r] = y_{\sigma+r} + y_{\sigma-r}$
   1.2 **For** i from 1 to $|N_s|$ **do**
      1.2.1 **For** j from 1 to $|N_s|$ **do**
         1.2.1.1 **Calculate** $gcross_\sigma[i,j] = \mathbf{g}_\sigma[i] \times \mathbf{g}_\sigma[j]$.
2. **For** i from 1 to $|N_s|$ **do**
   2.1 **For** j from 1 to $|N_s|$ **do**
      2.1.1 Compute **in parallel** the sum $gmatrix[i,j] = \sum_{\sigma\in\Omega} gcross_\sigma[i,j]$.
3. **For all** pixels σ **in parallel do**:
   3.1 **For each** $r \in N_s$ **do**
      3.1.1 **Calculate** $gv_\sigma[r] = \mathbf{g}_\sigma[r] \times y_\sigma$
4. **For each** $r \in N_s$ **do**
   4.1 Compute **in parallel** the sum $gvec[r] = \sum_{\sigma\in\Omega} gv_\sigma[r]$
5. Solve the $|N_s| \times |N_s|$ linear system of equations:
   $[gmatrix]_{|N_s|\times|N_s|} \times [\Theta^*]_{|N_s|\times 1} = [gvec]_{|N_s|\times 1}$
6. **Calculate** $\nu^* = \frac{1}{M^2} \sum_{\sigma\in\Omega} (y_\sigma - \Theta^{*t}\mathbf{g}_\sigma)^2$

end

Fig. 2. Least squares estimator algorithm.

Algorithm 3 *Maximum Likelihood Estimator for GMRF*
Note that $\Theta_k \equiv (\theta_1, \theta_2, \ldots, \theta_\tau, \nu)$.
Also, this algorithm assumes toroidal wrap-around of the image.
Note that in Step 5, $\beta < 1.0$, and we use $\beta = 0.8$.
Input:
$\{Y\} \leftarrow$ the image.
begin
1. **Find** Initial Guess Θ_0 using LSE Algorithm 2.
2. **Compute** $\nabla F(\Theta_k) \equiv \langle \frac{\partial F}{\partial \theta_1}, \frac{\partial F}{\partial \theta_2}, \ldots, \frac{\partial F}{\partial \theta_\tau}, \frac{\partial F}{\partial \nu} \rangle$.
3. **Compute** $\nabla^2 F(\Theta_k)$
4. **Solve** the following linear system of equations for vector ϑ
   $[\nabla^2 F(\Theta_k)]_{(\tau+1)\times(\tau+1)} \{\vartheta\}_{(\tau+1)\times 1} = -[\nabla F(\Theta_k)]_{(\tau+1)\times 1}$
5. **Determine** the largest λ *from* $\{1, \beta, \beta^2, \beta^3, \ldots\}$ such that
   (5a.) $1 - 2 \sum_{\tau_i \in N_s} \left(\Theta_{\tau_i}\, \Phi_{\tau_i}(\sigma)\right) > 0$; (note that these represent M^2 constraints)
   (5b.) $F(\Theta_k + \lambda\vartheta) > F(\Theta_k)$
6. **Set** $\Theta_{k+1} = \Theta_k + \lambda\vartheta$
7. If $|F(\Theta_{k+1}) - F(\Theta_k)| > \epsilon$ then go to Step 2.

end

Fig. 3. Maximum likelihood estimator algorithm.

$n \times n$ with G gray levels. The implementations known to the authors are slow and operate on images of size 128×128 or smaller.

In this correspondence, we develop scalable data parallel algorithms for implementing the most important texture sampling and synthesis algorithms. The data parallel model is an architecture-independent programming model that allows an arbitrary number of virtual processors to operate on large amounts of data in parallel. All the algorithms described in this paper have been implemented and thoroughly tested on a Connection Machine CM-2 and a Connection Machine CM-5.

Section II develops parallel algorithms for texture synthesis using Gibbs and Gaussian Markov random fields (GMRF's). Parameter estimation for Gaussian Markov random field textures (see Fig. 1), using least squares (see Fig. 2), as well as maximum likelihood techniques (see Fig. 3) are given in Section III. Conclusions are given in Section IV.

II. Texture Synthesis

A. A Parallel Gibbs Sampler

A discrete Gibbs random field (GRF) is specified by a probability mass function of the image as follows:

$$\Pr(X = x) = \frac{e^{-U(x)}}{Z} \tag{1}$$

where $U(x)$ is the energy function, and $Z = \sum U(x)$, over all G^n images; G being the number of gray levels, and the image is of size $\sqrt{n} \times \sqrt{n}$. Except in very special circumstances, it is not feasible to compute Z. A relaxation-type algorithm described in [6] simulates a Markov chain through an iterative procedure that readjusts the gray levels at pixel locations during each iteration. This algorithm sequentially initializes the value of each pixel using a uniform distribution. Then a single pixel location is selected at random, and using the conditional distribution that describes the Markov chain, the new gray level at that location is selected, dependent only upon the gray levels of the pixels in its local neighborhood. The sequential algorithm terminates after a given number of iterations.

The sequential algorithm to generate a Gibbs random field described in [6] and [7] is used as a basis for our parallel algorithm. For all the algorithms given in this correspondence, we use a symmetric neighborhood N_s, which is half the size of the standard neighborhood model N. This implies that if the vector $(i, j) \in N$, then $(-i, -j) \in N$, but only one of $\{(i, j), (-i, -j)\}$ is in N_s. Each element of array Θ is taken to represent the parameter associated with its corresponding element in N_s. We use the notation y_σ to represent the gray level of the image at pixel location σ.

Our Gibbs random field is generated using a simulated annealing type process. For an image with G gray levels, the probability $\Pr(X = k \mid \text{neighbors})$ is binomial with parameter $\Psi(T) = \frac{e^T}{1+e^T}$, and number of trials $G - 1$. The array $\{T\}$ is given in the following equation for a first-order model:

$$\begin{aligned} T = \alpha &+ \theta_{(1,0)}[y_{\sigma+(1,0)} + y_{\sigma-(1,0)}] \\ &+ \theta_{(0,1)}[y_{\sigma+(0,1)} + y_{\sigma-(0,1)}] \end{aligned} \tag{2}$$

and is a weighted sum of neighboring pixels at each pixel location. Additional examples of $\{T\}$ for higher order models may be found in [6].

This algorithm is ideal for parallelization. The calculation of $\{T\}$ requires uniform communication between local processing elements, and all other operations needed in the algorithm are data independent, uniform at each pixel location, scalable, and simple.

An example of a binary synthetic texture generated by the Gibbs sampler is given in Fig. 4.

Table I shows the timings of a binary Gibbs sampler for model orders 1, 2, and 4, on the CM-5. More extensive tables for both the CM-2 and CM-5 can be found in [1].

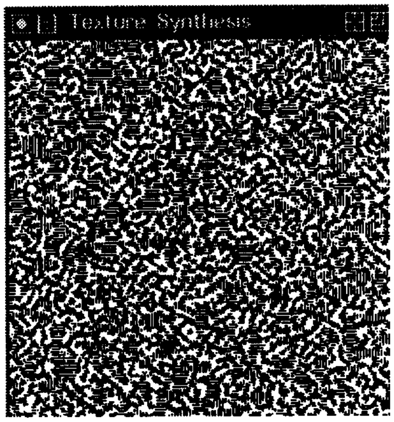

Fig. 4. Isotropic inhibition texture using Gibbs sampler (Texture 9b from [6]).

TABLE I
GIBBS SAMPLER TIMINGS FOR A BINARY ($G = 2$) IMAGE (EXECUTION TIME IN SECONDS PER ITERATION ON A CM-5 WITH VECTOR UNITS)

Image Size	*Order = 1*		*Order = 2*		*Order = 4*	
	16/vu CM-5	32/vu CM-5	16/vu CM-5	32/vu CM-5	16/vu CM-5	32/vu CM-5
8k	0.046053	0.024740	0.051566	0.027646	0.068486	0.038239
16k	0.089822	0.046824	0.099175	0.052411	0.130501	0.068630
32k	0.176997	0.089811	0.199399	0.099493	0.252421	0.132646
64k	0.351123	0.178046	0.398430	0.194271	0.560224	0.257647
128k	0.698873	0.351517	0.759017	0.383425	0.943183	0.582303
256k	1.394882	0.700164	1.526422	0.759747	1.874973	0.962165
512k	2.789113	1.394216	3.047335	1.520437	3.744542	1.892460
1M	5.577659	2.782333	6.009608	3.063054	7.428823	3.785890

B. Gaussian Markov Random Field Sampler

In this section, we consider the class of 2-D noncausal models known as GMRF models, which are described in [3], [5], and [9]. Pixel gray levels have joint Gaussian distributions and correlations controlled by a number of parameters representing the statistical dependence of a pixel value on the pixel values in a symmetric neighborhood. There are two basic schemes for generating a GMRF image model, both of which are discussed in [3]. The iterative GMRF sampler is similar to the Gibbs sampler, but instead of the binomial distribution, we use the continuous Gaussian distribution as the probability function. An efficient parallel implementation is straightforward and similar to that of the Gibbs sampler.

The previous section outlined an algorithm for sampling GMRF textured images using an iterative method. Unfortunately, this algorithm may have to perform hundreds or even thousands of iterations before a stable texture is realized. Next, we present a scheme that makes use of 2-D Fourier transforms and does not need to iterate. The direct GMRF sampler algorithm is realized from [3] as follows. We use the following scheme to reconstruct a texture from its parameters Θ and a neighborhood N_s:

$$\mathbf{y} = \frac{1}{M^2} \sum_{\sigma \in \Omega} \boldsymbol{f}_\sigma \frac{x_\sigma}{\sqrt{\mu_\sigma}} \tag{3}$$

where $\mathbf{y}$ is the resulting M^2 array of the texture image, and

$$\begin{aligned} x_\sigma &= \boldsymbol{f}_\sigma^{*t} \boldsymbol{\eta}, \\ \mu_\sigma &= (1 - 2\Theta^T \Phi_\sigma), \quad \forall \sigma \in \Omega, \\ \Phi_\sigma &= \mathrm{Col}\left[\cos \frac{2\pi}{M} \sigma^t r,\ r \in N_s\right]. \end{aligned} \tag{4}$$

The sampling process is as follows. We begin with $\boldsymbol{\eta}$, a Gaussian zero-mean noise vector with identity covariance matrix. We generate its Fourier series, via the fast Fourier transform, using $\boldsymbol{f}_\sigma$, the Fourier vector defined below, and finally apply (3).

$$\boldsymbol{f}_\sigma = \mathrm{Col}\,[1, \lambda_i, \lambda_i^2 \boldsymbol{t}_j, \cdots, \lambda_i^{M-1} \boldsymbol{t}_j], \quad \text{is an } M^2 \text{ vector}, \tag{5}$$

$$\boldsymbol{t}_j = \mathrm{Col}\,[1, \lambda_j, \lambda_j^2, \cdots, \lambda_j^{M-1}], \quad \text{is an } M\text{-vector},$$

$$\text{and} \quad \lambda_i = \exp\left(\sqrt{-1}\, \frac{2\pi i}{M}\right). \tag{6}$$

III. PARAMETER ESTIMATION FOR GMRF TEXTURES

Given a real textured image, we wish to determine the parameters of a GMRF model that could be used to reconstruct the original texture through the samplers given in the previous section.

This section develops parallel algorithms for estimating the parameters of a GMRF texture. The methods of least squares (LSE) and of maximum likelihood (MLE), both described in [3], are used. We present efficient parallel algorithms to implement both methods. The MLE performs better than the LSE. This can be seen visually by comparing the textures synthesized from the LSE and MSE parameters, or by noting that the asymptotic variance of the MLE is lower than the LSE [2], [10].

A. Least Squares Estimate of Parameters

The least squares estimate detailed in [3] assumes that the observations of the GMRF image $\{y_\sigma\}$ obey the model

$$y_\sigma = \sum_{r \in N_s} \Theta_r [y_{\sigma+r} + y_{\sigma-r}] + e_\sigma, \quad \forall \sigma \in \Omega \tag{7}$$

where $\{e_\sigma\}$ is a zero-mean correlated noise sequence with variance ν and correlation with the following structure:

$$E(e_\sigma e_r) = \begin{cases} -\Theta_{\sigma-r}\nu, & \text{if } (\sigma - r) \in N, \\ \nu, & \text{if } \sigma = r, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Then, for $\boldsymbol{g}_\sigma = \mathrm{Col}\,[y_{\sigma+r'} + y_{\sigma-r'},\ r' \in N_s]$, the LSE are

$$\Theta^* = \left[\sum_\Omega \boldsymbol{g}_\sigma \boldsymbol{g}_\sigma^t\right]^{-1} \left(\sum_\Omega \boldsymbol{g}_\sigma y_\sigma\right)$$

$$\nu^* = \frac{1}{M^2} \sum_\Omega (y_\sigma - \Theta^{*t} \boldsymbol{g}_\sigma)^2 \tag{9}$$

where Ω is the complete set of M^2 pixels, and toroidal wrap-around is assumed.

B. Maximum Likelihood Estimate of Parameters

We introduce the following approach as an improved method for estimating GMRF parameters of textured images. The method of maximum likelihood gives a better estimate of the texture parameters since the asymptotic variance of the MLE is lower than that of the LSE. We also show a much faster algorithm for optimizing the joint probability density function, which is an extension of the Newton–Raphson method and is also highly parallelizable.

Assuming a toroidal lattice representation for the image $\{y_\sigma\}$ and Gaussian structure for noise sequence $\{e_\sigma\}$, the joint probability density function is the following:

$$p(y \mid \Theta, \nu) = \frac{1}{(2\pi\nu)^{\frac{M^2}{2}}} \sqrt{\prod_{\sigma \in \Omega} \left\{1 - 2 \sum_{\tau_i \in N_s} [\Theta_{\tau_i} \Phi_{\tau_i}(\sigma)]\right\}} \cdot \exp\left(-\frac{1}{2\nu}\left\{C(0) - \sum_{\tau_i \in N} [\Theta_{\tau_i} C(\tau_i)]\right\}\right). \tag{10}$$

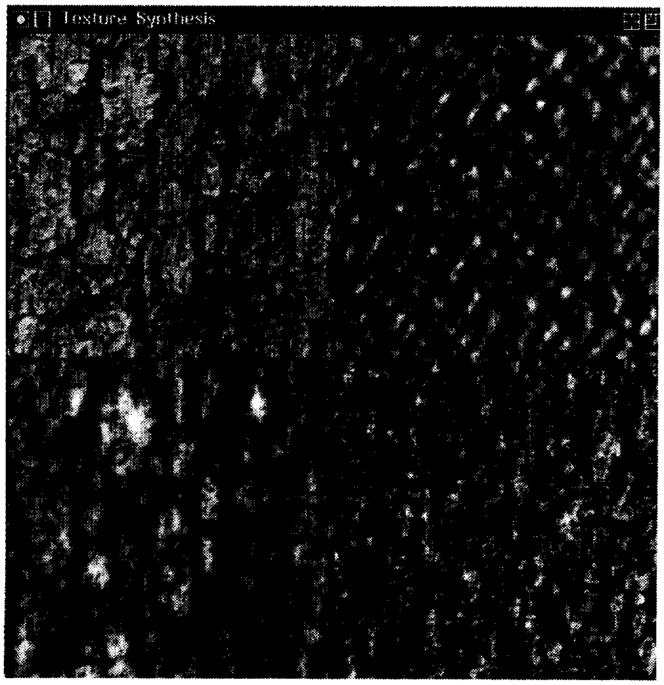

Fig. 5. Tree bark texture: (clockwise from top left) original image, reconstructed from the LSE, MLE, and Compressed image. A model whose parameters are listed in Table II was used.

In (10), $C(\tau_i)$ is the sample correlation estimate at lag τ_i. As described in [2] and [3], the log-likelihood function can be maximized: (note that $F(\Theta, \nu) = \log p(y \mid \Theta, \nu)$)

$$F(\Theta, \nu) = -\frac{M^2}{2} \log 2\pi\nu + \frac{1}{2} \sum_{\sigma \in \Omega} \cdot \left(\log \left\{ 1 - 2 \sum_{\tau_i \in N_s} [\Theta_{\tau_i} \Phi_{\tau_i}(\sigma)] \right\} \right) - \frac{1}{2\nu} \sum_{\sigma \in \Omega} \left(y(\sigma)^2 - y(\sigma) \sum_{\tau_i \in N_s} \cdot \{\Theta_{\tau_i} [y(\sigma + r_i) + y(\sigma - r_i)]\} \right). \tag{11}$$

For a square image, Φ_{τ_i} is given as follows:

$$\Phi_{\tau_i}(\sigma) = \cos\left(\frac{2\pi}{M} \sigma^T r_i\right). \tag{12}$$

This nonlinear function $\mathbf{F}$ is maximized by using an extension of the Newton–Raphson method. This new method first generates a *search direction* ϑ^k by solving the system

$$[\nabla^2 F(\Theta_k)]_{(r+1)\times(r+1)} [\vartheta^k]_{(r+1)\times 1} = -[\nabla F(\Theta_k)]_{(r+1)\times 1}. \tag{13}$$

Note that this method works well when $\nabla^2 F(\Theta_k)$ is a symmetric, positive-definite Hessian matrix. We then *maximize* the step in the search direction, yielding an approximation to λ_k that attains the local maximum of $F(\Theta_k + \lambda\vartheta)$ and also satisfies the constraints that each of the M^2 values in the logarithm term for $\mathbf{F}$ is positive. Finally, an *optimality test* is performed. We set $\Theta_{k+1} = \Theta_k + \lambda\vartheta$, and if Θ_{k+1} is sufficiently close to Θ_k, the procedure terminates. We give the first and second derivatives of $\mathbf{F}$ with respect to Θ_k and ν in [1].

For a rapid convergence of the Newton–Raphson method, it must be initialized with a good estimate of parameters close to the global maximum. We use the least squares estimate given in Section III-A as Θ_0, the starting value of the parameters.

TABLE II
Θ PARAMETERS FOR TREE BARK TEXTURE

Parameter	LSE	MLE	Parameter	LSE	MLE
(1,0)	0.590927	0.568643	(0,3)	0.024942	0.015561
(0,1)	0.498257	0.497814	(4,0)	-0.019122	-0.006186
(1,1)	-0.281546	-0.272283	(0,4)	-0.009040	-0.003748
(-1,1)	-0.225011	-0.219671	(-2,1)	0.045105	0.036778
(0,2)	-0.125950	-0.128427	(1,-2)	0.031217	0.040860
(2,0)	-0.203024	-0.162452	(1,2)	0.061537	0.067912
(2,2)	-0.014322	-0.017466	(2,1)	0.067865	0.055445
(2,-2)	-0.002711	-0.007541	ν	22205.84	65.45
(3,0)	0.060477	0.034623	$F(\Theta)$	-266147.34	-264245.13

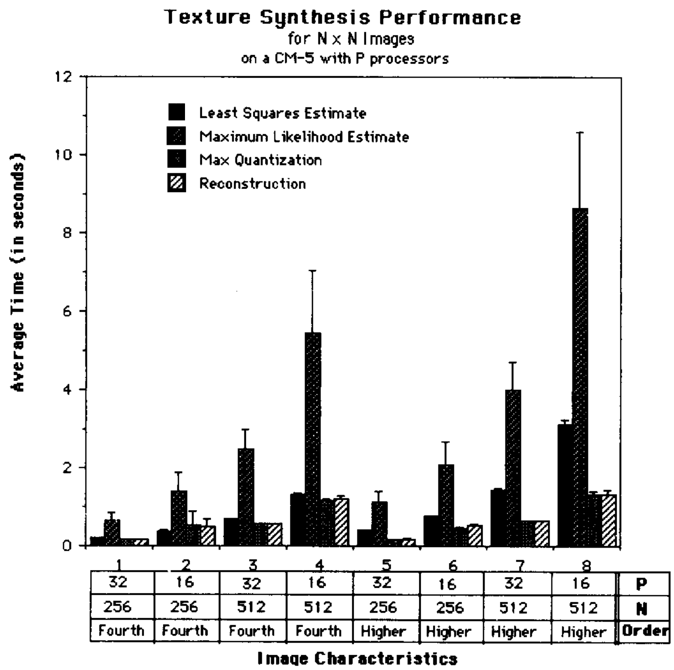

Fig. 6. Timings for parallel image processing techniques.

In Fig. 5, we show the synthesis using least squares and maximum likelihood estimates for tree bark obtained from standard textures library. Table II shows the respective parameters for both the LSE and MLE and give their log-likelihood function values. This example shows that the maximum likelihood estimate improves the parameterization. In addition, CM-5 timings for these estimates varying machine size, image size, and neighborhood models can be found in Fig. 6 for both fourth and higher order models on this selection of real-world textured images. The value plotted is the mean time over 13 diverse images, and errors bars give the standard deviation. More explicit tables, as well as CM-2 timings, for these estimates can be found in [1].

IV. CONCLUSIONS

We have presented efficient data parallel algorithms for texture analysis and synthesis based on Gibbs or Markov random field models. A complete software package running on the Connection Machine model CM-2 and the Connection Machine model CM-5 implementing these algorithms is available for distribution to interested parties. Please see http://www.umiacs.umd.edu/~dbader for additional information. The experimental data strongly support the analysis concerning the scalability of our algorithms. The same type of algorithms can be used to handle other image processing algorithms such as image estimation [8], [9], texture segmentation [5], and integration of early vision modules. We are currently examining several of these extensions.

The parameters for the 256 × 256 image of tree bark texture in Fig. 5 are given in Table II.

REFERENCES

[1] D. A. Bader, J. JáJá, and R. Chellappa, "Scalable data parallel algorithms for texture synthesis and compression using Gibbs random fields," UMIACS and Elec. Eng., Univ. of Maryland, College Park, MD, Tech. Rep. CS-TR-3123 and UMIACS-TR-93-80, Aug. 1993.

[2] J. E. Besag and P. A. P. Moran, "On the estimation and testing of spacial interaction in Gaussian lattice processes," *Biometrika*, vol. 62, pp. 555–562, 1975.

[3] R. Chellappa, "Two-dimensional discrete Gaussian Markov random field models for image processing," in *Progress in Pattern Recognition*, vol. 2, L. N. Kanal and A. Rosenfeld, Eds. New York: Elsevier, 1985, pp. 79–112.

[4] R. Chellappa and R. L. Kashyap, "Synthetic generation and estimation in random field models of images," in *Proc. IEEE Comp. Soc. Conf. Patt. Recog. Image Processing*, Dallas, TX, Aug. 1981, pp. 577–582.

[5] F. S. Cohen, "Markov random fields for image modeling and analysis," in *Modelling and Applications of Stochastic Processes*, U. Desai, Ed. Boston, MA: Kluwer, 1986, pp. 243–272, ch. 10.

[6] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern. Anal. Machine Intell.*, vol. PAMI-5, pp. 25–39, Jan. 1983.

[7] R. C. Dubes and A. K. Jain, "Random field models in image analysis," *J. Applied Statist.*, vol. 16, pp. 131–164, 1989.

[8] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, Nov. 1984.

[9] F. C. Jeng, J. W. Woods, and S. Rastogi, "Compound Gauss–Markov random fields for parallel image processing," in *Markov Random Fields: Theory and Application*, R. Chellappa and A. K. Jain, Eds. Boston, MA: Academic, 1993, chap. 2, pp. 11–38.

[10] R. L. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spacial interaction models of images," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 60–72, Jan. 1983.

Linear Filtering of Images Based on Properties of Vision

V. Ralph Algazi, Gary E. Ford, and Hong Chen

***Abstract*— The design of linear image filters based on properties of human visual perception has been shown to require the minimization of criterion functions in both the spatial and frequency domains. In this correspondence, we extend this approach to continuous filters of infinite support. For lowpass filters, this leads to the concept of an ideal lowpass image filter that provides a response that is superior perceptually to that of the classical ideal lowpass filter.**

I. INTRODUCTION

THE use of hard cutoff (ideal) lowpass filters in the suppression of additive image noise is known to produce ripples in the response to sharp edges. For high contrast edges, human visual perception fairly simply determines acceptable filter behavior. Ripples in the filter response are visually masked by the edge, so that the contrast sensitivity of the visual system decreases at sharp transitions in image intensity and increases somewhat exponentially as a function of the spatial distance from the transition.

Manuscript received November 5, 1993; revised December 21, 1994. This work was supported by the University of California MICRO Program, Grass Valley Group, Pacific Bell, Lockheed, and Hewlett Packard. The associate editor coordinating the review of this paper and approving it for publication was Prof. Nikolas P. Galatsanos.

The authors are with the CIPIC, Center for Image Processing and Integrated Computing, University of California, Davis, Davis, CA 95616 USA.

IEEE Log Number 9413843.

Algorithmic procedures using properties of human vision have been described for over 20 years [1]. The development of adaptive methods of image enhancement and restoration, based on the use of a masking function, measure spatial detail to determine visual masking [2], [3]. In active regions of the image, visual masking is high, relative noise visibility is low, and the filter applied is allowed to pass more noise until the subjective visibility is equal to that in flat areas.

Whether the filter is adaptive or not, the design of the linear filter to be applied is a critical issue. Hentea and Algazi [4] have demonstrated that the first perceptible image distortions due to linear filtering occur at the major edges and thus, worst case design for visual appearance should be based on edge response. They developed a filter design approach based on the minimization of a weighted sum of squared-error criterion functions in both the spatial and frequency domains. In the spatial domain, the weighting is by a visibility function, representing the relative visibility of spatial details as a monotonically increasing function of the distance from an edge. This visibility function, determined experimentally from the visibility of a short line positioned parallel to an edge, was also found experimentally to predict satisfactorily the visibility of ripples due to linear filters [4].

In the following, we extend the work of Hentea and Algazi by considering the design and properties of one-dimensional continuous filters of infinite support (two-dimensional filters are generated by 1 D to 2 D transformations). We obtain a new formal result on the lowpass filter of infinite support that is optimal for images. It establishes the limiting performance that digital filters of finite complexity can only approximate.

II. DESIGN OF ONE-DIMENSIONAL FILTERS FOR IMAGES

The basic tradeoff in the design approach of Algazi and Hentea [4] is maintaining image quality while reducing unwanted artifacts or noise. The image quality is measured by spatial domain criterion function for the visibility of ripples in the vicinity of edges

$$I_1 = \int_{-\infty}^{\infty} w_1^2(x)[\hat{u}(x) - u(x)]^2\,dx \tag{1}$$

where $u(x)$ is a unit step input producing the filter response $\hat{u}(x) = u(x) * h(x)$, where $h(x)$ is the point spread function of the filter, $*$ denotes convolution and $w_1(x)$ is a spatial weighting function, chosen to be the visibility function

$$w_1(x) = 1 - a^{|x|}. \tag{2}$$

The frequency domain criterion function for the reduction of unwanted artifacts and noise is

$$I_2 = \int_{-\infty}^{\infty} W_2^2(f)|H(f) - H_d(f)|^2\,df \tag{3}$$

where $H_d(f)$ is the desired filter frequency response and $W_2(f)$ is the frequency-domain weighting function. Hentea and Algazi minimized I_1 under a constraint on I_2, but we now minimize the equivalent criterion $J(\alpha) = \alpha I_1 + (1-\alpha)I_2$ where α controls the relative weights of the two criteria, with $0 \le \alpha \le 1$.

To develop the optimality condition, (1) is expressed in the frequency domain using Parseval's relation, the transform of a zero-mean step is used, and calculus of variations is applied to the criterion