USC-SIPI REPORT # 126

Pyramid Implementation of Optimal Step Conjugate Search Algorithms for Some Computer Vision Problems

By

T. Simchony, R. Chellappa, H. Jinchi and Z. Lichtenstein

University of Southern California
Signal and Image Processing Institute
Department of Electrical Engineering-Systems
Powell Hall of Engineering
University Park/MC-0272
Los Angeles, CA 90089

Telephone No.: (213)743-8559

Primary author for correspondence:

Prof. Rama Chellappa at the above address.

Partially supported by the National Science Foundation - Grant No. : MIP-84-51010 and matching funds from IBM and AT&T. The first author is also supported by ECI Telecom.

Pyramid Implementation of Optimal Step Conjugate Search Algorithms for Some Computer Vision Problems

Abstract

Optimization of a cost function arises in several computer vision problems. The cost functions in these problems are usually derived from discretization of functionals obtained from regularization principles or stochastic estimation techniques using Markov random field models. In this paper we present a parallel implementation on a pyramid of the line search conjugate gradient algorithm for minimizing the cost functions mentioned above. By viewing the global cost function as a Gibbs energy function, we efficiently compute the gradients, inner products and the optimal step size using the pyramid. The global Gibbs energy of a given configuration is broken into its basic energy terms associated with different cliques. We let each low level processor sum the energies associated with its cliques. The local energy terms are added by the intermediate levels of the pyramid to the top, where the step size is determined by an efficient univariate search. Such implementation allows us to calculate the global energy in $O(\log n)$ operations, where n is the number of grid points in each direction. Implementation of this algorithm to shape from shading results in a multiresolution conjugate gradient algorithm. Robustness and efficiency of the algorithm are also demonstrated on edge detection using graduated non convexity algorithm, and estimation of gray level images degraded by multiplicative noise.

1 Introduction

Many physical phenomena that govern the processes that are of interest to research in computer vision, can be described by their local characteristics. They are formulated mathematically using differential or integral equations in the deterministic approach, or Markov fields in the stochastic formulation. Examples for the deterministic formulation are the image irradiance equation, the optical flow equation [1], etc. An example for the stochastic formulation is image restoration using Markov random fields models [2]. Computer vision research is usually focused on the inverse problem [3], i.e., given an observation (usually an image, or a sequence of images) find a dense depth map, the velocity field etc. These problems are in many cases ill posed in the sense of Hadamard [4].

In order to overcome the ill posedness, regularization methods have been suggested [3]. The original problems are reformulated as optimization problems. Instead of solving the original differential equation exactly, an integral consisting of two elements is constructed. The first element minimizes the mean square error between the observation and the corresponding calculated quantity (from the differential equation). The second term is a penalty term that favors smooth solutions. The desired solution is the minimum of this functional (integral). There are several ways to obtain the minimum of the functional. One can use calculus of variations to convert the optimization problem into a set of differential equations [5].

This variational approach was extensively explored for the shape from shading (SFS) problem [6], which deals with the reconstruction of the orientation map of a surface from a single image using the reflectance map equation. This formulation leads to nonlinear differential equations, which are not easy to solve unless the penalty term in the functional is sufficiently large (the

solution is far from the original problem). Alternatively, one can use direct optimization [5]. Direct methods discretize the integral in the functional and replace the partial derivatives by finite difference approximations to obtain cost functions of the different attributes' values at all grid points (which we call-configurations). In the SFS problem, one obtains a cost function on the orientation of the surface at all the grid points. The minimum of this cost function is the desired solution to the SFS problem [7]. Using standard calculus[7] , the configuration that sets the gradient of the cost function with respect to the orientation at each grid point to zero is searched for. The same methodology is used in [8], leading to a system of sparse nonlinear equations. The solution is obtained via a relaxation method in [7] or a direct method in [8]. The algorithms converge only when the penalty term is large, which yields an oversmoothed solution.

Cost functions also yield a suitable formulation for problems involving discontinuities in the observation or in the reconstructed attribute. An example for this formulation is Blakes' and Zissermans' weak membrane formulation [9] for the surface interpolation problem. They fit a weak membrane (weak membrane is a membrane that satisfies weak continuity constraints) to noisy data, and use it for edge detection, by marking the points where discontinuities in the membrane height occur. The cost function derived using this model is not convex. To find the global minimum of the cost function F it is suggested that the Graduated Non Convexity algorithm (GNC) may be used. The GNC algorithm starts with a convex approximation to the cost function F^*. Then a whole sequence of cost functions $F^{(P)}$, $0 \leq P \leq 1$ is constructed, so that $F^{(0)} = F$, the original nonconvex function, and $F^{(1)} = F^*$. In between, $F^{(P)}$ changes continuously, between $F^{(1)}$ and $F^{(0)}$. The GNC algorithm optimizes a whole sequence of $F^{(P)}$, for example $\{F^{(1)}, F^{(\frac{1}{2})}, F^{(\frac{1}{4})}, F^{(\frac{1}{8})}, F^{(\frac{1}{16})}\}$, one after the other, using the optimal solution of previous

optimization srep as the starting point for the next. The optimization step is then performed using the sequential over relaxation (SOR) method. The authors report a big degradation in the convergence rate of the SOR method when the processed image is corrupted with excessive noise.

The third cost functional formulation is derived from stochastic modeling. When one models a process by a Markov field, and exploits the Gibbs-Markov equivalence theorem, a Gibbs energy function is obtained for the different configurations. As an example, consider the problem of image estimation. We assume that the original image obeys a Gauss Markov Random Field (GMRF) model (note that GMRF models can be viewed as extended regularization models). The noise is assumed to be multiplicative and spatially independent. The objective is the maximum aposterioi distribution (MAP) solution, or the configuration that correspond to the minimum of the Gibbs energy function. This problem illustrates the equivalence between a Gibbs energy function, and the cost functions obtained by direct optimization. If the Gibbs energy function is not convex it can be minimized using simulated annealing [2][10], but the algorithm is computationally extensive.

In this paper we propose a unified and practical method for minimizing the cost functions resulting from the deterministic and stochastic formulations described above. The method exploits the equivalence between direct optimization of a deterministic problem, and the MAP solution using the Markov random fields, through the use of Gibbs energy function, and treats the two problems with the same algorithm. We chose an optimization algorithm [11] that is amenable to parallel computation to cope up with the computational complexity induced by the high dimensional cost functions. The special structure of Gibbs energy functions is utilized to map the algorithm to a highly parallel implementation. Since in general the cost function is not quadratic,

we use an extended version of the original conjugate gradient method- the line search extension proposed by Polak and Ribiere [11]. The algorithm is very efficient in solving high dimensional unconstrained optimization problems. The complete algorithm is :

- Step1. Given X_0 compute $G_0 = \nabla f(X_0)$ and set $D_0 = -G_0$.
- Step2. For $k = 0, 1, \cdots, n-1$:
  - a) Set $X_{k+1} = X_k + \alpha_k D_k$ where α_k minimizes $f(X_k + \alpha D_k)$.
  - b) Compute $G_{k+1} = \nabla f(X_{k+1})$.
  - c) Unless $k = n-1$, set $D_{k+1} = -G_{k+1} + \beta_k D_k$ where

$$\beta_k = \frac{(G_{k+1} - G_k)^T G_{k+1}}{G_k^T G_k}$$

- Step3. Replace X_0 by X_n and go back to Step 1.

The use of the algorithm requires computations of inner products in Step 2c, and finding the optimal step in Step 2a. The value of the optimal step depends on the global cost function. In order to efficiently calculate the global cost function and the inner products required for computing β_k, we propose to use a pyramid implementation. We view the global cost function as a Gibbs energy function corresponding to a given configuration. We break the energy function to its basic energy terms associated with different cliques. We let each low level processor sum the energies associated with its cliques, making sure that each clique's energy is summed only once. The local energy terms are added by the intermediate levels of the pyramid to the top, where the step size is determined by an efficient univariate search. Such an implementation allows us to calculate the global energy in $O(\log n)$ operations, where n is the number of grid points in

each direction. This search is bound to converge to a maximal point of the cost function. We assume that the cost function has only one extremum point corresponding to the desired solution. The additional component of the algorithm is the computation of the gradient in step 2b. This computation can be implemented in parallel on the bottom of the pyramid, because the gradients of the cost functions (for this class of problems) depend only on the function values in a small neighborhood of the pixel.

An additional improvement can be obtained for the SFS problem if one searches for the minimum of the discrete cost function using a multi resolution conjugate gradient algorithm. The rate of convergence of the multi resolution search is an order of magnitude faster than the single resolution version. This improvement is due to better propagation of low frequency information on the coarse grid. Determination of the higher frequency modes occurs as the grid gets finer and finer. This multi resolution search utilizes the same properties that make multigrid method so successful [12]. We feel that one can use conjugate gradient search as the relaxation step in the multigrid relaxation method.

We have also studied the use of the extended conjugate gradient search to replace the SOR step in Blake's and Zisserman's [9] graduated non convexity (GNC) algorithm. Once again we use the Gibbs formulation in order to obtain an efficient and parallel implementation. The results we obtain using the conjugate gradient algorithm require a smaller number of iterations than the SOR method [9]. The convergence rate of the algorithm is much less sensitive to noise. This example demonstrates the robustness of the algorithm in comparison to other methods.

The algorithm is also applied to the MAP estimation of gray level images degraded by multiplicative noise. Although the cost function is not convex, we obtained good estimates for all the

different initial conditions that we tried. The resulting algorithm required much less computation than simulated annealing.

The organization of the paper is as follows. In section 2, the SFS problem is briefly discussed, and the multi resolution conjugate gradient algorithm for reconstructing the orientation of the surface is presented. In section 2.1 a synthetic sphere image is experimented with both a single and multi resolution versions of the algorithm to show the reduction in computation time obtained with the multi resolution implementation. Section 3 discusses estimation of gray level images corrupted by multiplicative noise. We present experiments to show that although the cost function is not convex good results are obtained using the deterministic algorithm for both 5dB and 0dB noise. In section 4, we present the GNC algorithm with a modified optimization step, based on the conjugate gradient search. Experimental results for edge detection are presented in section 4.1. The experiments indicate that the algorithm is more robust than SOR.

2 Shape From Shading

2.1 Discrete Cost Function for SFS and Gibbs Representation

The SFS problem is the problem of reconstructing the surface orientation from the observed image intensity. Let $E(x,y)$ be the observed image of intensity related to a surface $Z(x,y)$ as

$$E(x,y) = R(p,q,\beta,l,\rho) \tag{1}$$

where β is the illumination direction vector, l is the vector from the surface to the camera, ρ is the albedo term, and $p = Z_x$ and $q = Z_y$ are the surface slopes. In the case of a Lambertian

surface, one can further write (1) as,

$$E(x,y) = \frac{\rho\beta \cdot (-p,-q,1)}{(1+p^2+q^2)^{\frac{1}{2}}} \tag{2}$$

The discrete set up for the SFS problem was first suggested by Strat [7], using the following cost function:

$$\epsilon^2 \sum_{i=1}^{n}\sum_{j=1}^{m}(E_{ij} - R(p_{ij},q_{ij}))^2 + \frac{\lambda}{\epsilon^2}\sum_{i=1}^{n}\sum_{j=1}^{m} e_{ij}^2 \tag{3}$$

where the first term corresponds to the irradiance error. The second term e_{ij} is the integrability penalty term which corresponds to an estimate for the integral around an elementary square path in the counter-clockwise direction, with the picture cell (i,j) in the lower left corner, i.e.

$$e_{ij} = \frac{\epsilon}{2}[p_{i,j} + p_{i+1,j} + q_{i+1,j} + q_{i+1,j+1} - p_{i+1,j+1} - p_{i,j+1} - q_{i,j+1} - q_{i,j}] \tag{4}$$

In our work we use Strat's integrability term. A different discrete cost function was suggested by Lee [8] :

$$\mu = \epsilon^2 \sum_{i=1}^{n}\sum_{j=1}^{m}(E_{ij} - R(f_{ij},g_{ij}))^2 + \kappa\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} \tag{5}$$

where f and g are the surface orientation in stereographic coordinates and r_{ij} is a smoothing penalty term:

$$r_{ij} = [(f_{i+1,j} - f_{i,j})^2 + (f_{i,j+1} - f_{i,j})^2 + (g_{i+1,j} - g_{i,j})^2 + (g_{i,j+1} - g_{i,j})^2]/h^2$$

In our formulation we work with p and q and use two different cost functions. The first is a straight forward modification of (5) to p and q coordinates,and the second adds the integrability constraint described in (3) and (4) to obtain:

$$\epsilon^2 \sum_{i=1}^{n}\sum_{j=1}^{m}(E_{ij} - R(p_{ij},q_{ij}))^2 + \frac{\lambda}{\epsilon^2}\sum_{i=1}^{n}\sum_{j=1}^{m} e_{ij}^2 + \kappa\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} \tag{6}$$

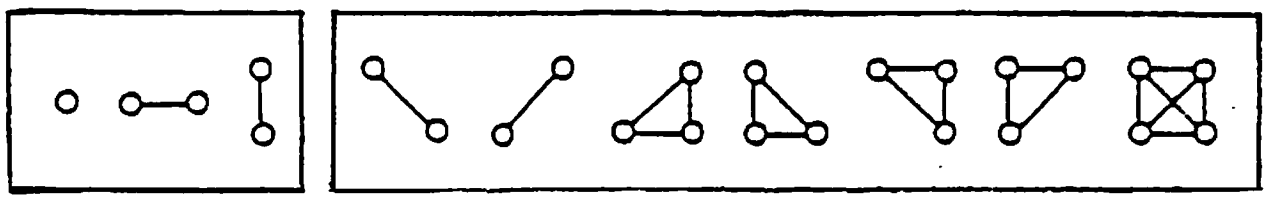
We now explain the Gibbs formulation for the cost function.

Let $S = \{s_1, s_2, \ldots, s_N\}$ be the set of sites (grid points), $\mathcal{G} = \{\mathcal{G}_s, s \in S\}$ be a neighborhood system for S, meaning any collection of subsets of S for which 1) $s \notin \mathcal{G}_s$ and 2) $s \in \mathcal{G}_r \iff r \in \mathcal{G}_s$. $\mathcal{G}_s$ is the set of neighbors of s and $\{S, \mathcal{G}\}$ is a graph in the usual way. A subset $C \subset S$ is a clique if every pair of distinct sites in C are neighbors. $\mathcal{C}$ denotes the set of cliques [2]. In our problem S is the set of pixel sites (grid points) of the image intensity, $N = n^2$ where n is the number of grid points in each direction. It is easier to view s_m as a two dimensional vector $s_m = \{(i,j) \mid i,j \in [1 \cdots n]\}$. One can look into a homogeneous neighborhood system of the form

$$\mathcal{G} = \mathcal{F}_c = \{\mathcal{F}^c_{i,j}, (i,j) \in S\}$$

$$\mathcal{F}^c_{i,j} = \{(k,l) \in S \mid 0 < (k-i)^2 + (l-j)^2 < c\}$$

Notice that the sites near the boundary have fewer neighbors than the interior ones. The cliques for $c = 1$ are all of the form $\{(i,j)\}$,$\{(i,j),(i,j+1)\}$, $\{(i,j),(i+1,j)\}$ as shown in Figure 1A, and for $c = 2$ we have the above and the additional cliques described in Figure 1B.



(1 A) (1 B)

Figure 1: Gibbs Energy Cliques.

1 A: First Order MRF Cliques (c=1); 1 B: Second Order MRF Cliques (c=2);

The cost function:

$$U(p,q) = \sum_{i,j} \epsilon^2 (E_{ij} - R(p_{ij}, q_{ij}))^2 + \kappa[(p_{i,j+1} - p_{i,j})^2 + (p_{i+1,j} - p_{i,j})^2 + (q_{i,j+1} - q_{i,j})^2 + (q_{i+1,j} - q_{i,j})^2] \tag{7}$$

can be broken into a sum of energy terms corresponding to a nearest neighbor system. The only cliques we need to consider correspond to $c = 1$. Thus, we can write:

$$U(p,q) = \sum_{C \in \mathcal{C}} V_C(p,q)$$

Where $\mathcal{C}$ denote the set of Cliques for $\mathcal{G} = \mathcal{F}_1$. In general the stencil we use for finite difference approximation to differential operators determines the spatial dependence of the cost function. It is reflected in the type of cliques we have to use when we break the cost function into it's primal components. Thus, for (6) the cliques consist of one or two elements. In (7), the spatial dependence corresponds to $c = 2$ and the additional cliques consist of two elements at most. Note, that to each pixel corresponds a two element vector of unknowns: $(p_{i,j}, q_{i,j})$. The unknowns are coupled together in the Gibbs energy function. This energy function is very similar to energy functions of coupled Markov Processes [2]. Minimizing our cost function is totally equivalent to solving for the configuration with maximum probability. A deterministic algorithm for solving this problem was suggested in [13], in which Besag minimizes the cost function for a grid point (i,j) given the values of it's neighbor set. The algorithm uses only local computation, and can be implemented in parallel. The problem with using the algorithm when the unknowns can take values in a continuous set, is that the computations involved in minimizing the energy of the conditional distribution is not trivial. Thus, this algorithm is not practical for our problem. All the deterministic algorithms have one limitation in common-they all guarantee convergence only to a local minimum. To avoid the difficulty, we assumed that the minimum is unique. This is not

always true, as discussed in [14]. We can use our formulation to deal with the multiple minima case. An algorithm for finding the global minimum of a Gibbs energy function was suggested by Geman and Hwang in [15], using an algorithm based on a simulated diffusion process. The algorithm performs a gradient search combined with a Brownian motion component, which is gradually attenuated to zero. This search can be implemented efficiently, on the architecture suggested in this paper. The only modification required, is adding a white noise component to each element of the computed gradient.

2.2 The Multi Resolution Conjugate Gradient Algorithm Implemented on a Pyramid

We first calculate the gradient corresponding to (5):

$$\frac{\partial \mu}{\partial p_{i,j}} = 2\epsilon^2(E_{i,j} - R_{i,j})\frac{\partial R_{i,j}}{\partial p_{i,j}} + \kappa(4p_{i,j} - \hat{p}_{i,j}) \tag{8}$$

where

$$\hat{p}_{i,j} = p_{i,j+1} + p_{i,j-1} + p_{i+1,j} + p_{i-1,j}$$

and

$$\frac{\partial \mu}{\partial q_{i,j}} = 2\epsilon^2(E_{i,j} - R_{i,j})\frac{\partial R_{i,j}}{\partial q_{i,j}} + \kappa(4q_{i,j} - \hat{q}_{i,j}) \tag{9}$$

where

$$\hat{q}_{i,j} = q_{i,j+1} + q_{i,j-1} + q_{i+1,j} + q_{i-1,j}$$

for the Lambertian Reflectance map, and light source located at the camera, (8) and (9) become:

$$\frac{\partial \mu}{\partial p_{i,j}} = 2\epsilon^2(E_{i,j} - \frac{1}{(1 + p_{i,j}^2 + q_{i,j}^2)^{\frac{1}{2}}})\frac{p_{i,j}}{(1 + p_{i,j}^2 + q_{i,j}^2)^{\frac{3}{2}}} + \kappa(4p_{i,j} - \hat{p}_{i,j})$$

$$\frac{\partial \mu}{\partial q_{i,j}} = 2\epsilon^2(E_{i,j} - \frac{1}{(1+p_{i,j}^2+q_{i,j}^2)^{\frac{1}{2}}})\frac{q_{i,j}}{(1+p_{i,j}^2+q_{i,j}^2)^{\frac{3}{2}}} + \kappa(4q_{i,j} - \hat{q}_{i,j}) \tag{10}$$

Note that the spatial communication required is only with the nearest neighbors. Next, we note that if the boundary and initial conditions are trivial, the gradient is zero-we are at a local maximum of the energy function. We can now use the above example to illustrate the importance of enforcing integrability in the gradient search. One can see that if we start with an initial condition that satisfies ($p = q$) and the condition holds on the boundary, the search is then limited to the subspace ($p = q$), because the gradient is the same for p and q in every iteration. This problem is solved when the integrability term is added to obtain:

$$\begin{aligned}
\frac{\partial \mu}{\partial p_{i,j}} &= 2\epsilon^2[E_{i,j} - \frac{1}{(1+p_{i,j}^2+q_{i,j}^2)^{\frac{1}{2}}}]\ \frac{p_{i,j}}{(1+p_{i,j}^2+q_{i,j}^2)^{\frac{3}{2}}} + \\
&\kappa(4p_{i,j} - \hat{p}_{i,j}) + \frac{\lambda}{\epsilon^2}(e_{i,j} + e_{i,j-1} - e_{i-1,j-1} - e_{i-1,j}) \\
\frac{\partial \mu}{\partial q_{i,j}} &= 2\epsilon^2[E_{i,j} - \frac{1}{(1+p_{i,j}^2+q_{i,j}^2)^{\frac{1}{2}}}]\ \frac{q_{i,j}}{(1+p_{i,j}^2+q_{i,j}^2)^{\frac{3}{2}}} + \\
&\kappa(4q_{i,j} - \hat{q}_{i,j}) + \frac{\lambda}{\epsilon^2}(e_{i,j-1} + e_{i-1,j-1} - e_{i,j} - e_{i-1,j})
\end{aligned} \tag{11}$$

where $e_{i,j}$ was defined in (4). We now give the algorithm:

1. set the initial resolution to $l = 2^k$ and choose an initial solution on the coarse grid.

2. Given the initial configuration p^0, q^0 compute $G_0 = \nabla\mu(p^0, q^0)$ and set $D_0 = -G_0$.

3. calculate the optimal step for the given D_k using the global energy function (at the top of the pyramid) as described in the next paragraph.

4. perform the descent step

$$p^{k+1} = p^k + \alpha_k D_k^p$$

$$q^{k+1} = q^k + \alpha_k D_k^q$$

5. calculate the gradient vector G_{k+1} for the given resolution (at the bottom of the pyramid)

6. calculate a new conjugate direction $D_{k+1} = -G_{k+1} + \beta_k D_k$ where

$$\beta_k = \frac{(G_{k+1} - G_k)^T G_{k+1}}{G_k^T G_k}$$

7. repeat steps 3-6 till either the reduction in the global energy is below a threshold or a fixed number of iteration have been completed.

8. stop if you have reached the desired resolution

9. refine the resolution ($l = 2^{k+1}$), perform a local coarse to fine extension of the solution to obtain an initial configuration for the finer level.

10. go back to step 2

Step 3 of the algorithm is implemented in the following manner. The top of the pyramid performs univariate minimization using an iterative algorithm, such as Golden Section search [16], to determine the optimal step. The algorithm uses the values of the cost function corresponding to different step sizes. In each iteration, a new value for the step size is propagated down the pyramid to the bottom layer, so that the processors on the bottom can calculate the updated values of p and q and then sum the energy terms associated with the cliques that correspond to their location. In order to avoid repeated terms in the sum, each energy term is weighted by a reciprocal of the number of elements that construct the corresponding clique.Then, the intermediate levels of the pyramid sum the energy terms corresponding to their "sons", in order

to obtain the global energy for the configuration at the top. When the optimal step is determined at the top, it is propagated down to the bottom layer, and step 4 is performed.

2.3 Experimental Results

The experiments were performed on a synthetic sphere image illuminated from above. The two gradients corresponding to the cost functions (6) , (7) were used to obtain a reconstruction of the surface, assuming a Lambertian reflectance map, and constant albedo. λ and κ were chosen experimentally. We found that the rate of convergence is highly effected by the choice of λ and κ. Setting the smoothing part to small values, slowed down the convergence rate significantly. As expected if smoothing dominated, the results started to differ from the original surface , because we have solved for a different problem. The results for (7) were obtained using 10 or less iterations at each resolution, starting with a 4×4 grid and finishing at 64×64 and 128×128 grids. The L_2 error is $O(10^{-2})$. When we decreased the smoothing part, better results were obtained and the accuracy improved to $O(10^{-3})$.

We repeated the first experiment with a single resolution conjugate gradient search. The results are summarized in the table below. The experiment supports the assumption that multiple resolutions increase the rate of convergence significantly.

Table 1. Comparison of Single and Multi Resolution Algorithms to Shape from Shading

grid size	resolution	iteration	energy	L_2 error $\times 10^{-2}$
64	multi	40	2.960476	2.12969
	single	104	2.888647	2.11380
128	multi	50	2.876788	1.15690
	single	252	2.851924	2.09900

3 MAP Estimation of Gray Level Images Corrupted by Multiplicative Noise Using Line Search Conjugate Methods

In this example we consider a Markov random field model for the gray level image. The optimal cost function is defined in terms of a probability distribution. We adopt a Bayesian approach and use the Gaussian Markov random field (GMRF) model for computing the MAP estimate of an image degraded by multiplicative or Poisson noise. The role of posterior density in estimation of degraded images is well known [17][2]. The MAP method can be generalized to nonlinear image models and to noise models different from additive noise.

Given the distribution of the corrupting noise we write the conditional density of the degraded image conditioned on the original image. The degradation we consider in this work is multiplicative Gaussian. We assume that the original image is generated by a 2-D noncausal GMRF model [18][19] with unknown parameters. This assumption enables us to write the probability density function of the original image as a function of the parameters of the GMRF model. Assuming that a prototype of the original image is available, we use the maximum likelihood technique

[20][21][22] to obtain the estimates of GMRF model parameters. Thus, the MAP algorithm is written in terms of the GMRF model parameter estimates.

Obtaining the MAP estimate involves maximization of a very complicated function. For a 64×64 image with gray levels in the range $1 - 256$, the dimensionality of the solution space is extremely large, making an exhaustive search for the MAP estimate computationally extensive. The search is implemented using the Polak-Ribiere conjugate gradient method. Experiments show that although theoretically it can converge to a local maximum, this method yielded good results for any initial condition we tried.

3.1 GMRF Models

Suppose that the original image $\{y(s)\}$ is defined on a finite lattice $\Omega = \{s = (i,j) : 1 \leq i,j \leq M\}$. We partition this finite lattice Ω into mutually exclusive and totally inclusive subsets Ω_B, the boundary set, and Ω_I, the interior set:

$$\begin{aligned} \Omega_B &= \{s = (i,j) : s \in \Omega \text{ and } (s+r) \notin \Omega \\ &\qquad \text{for at least one } r \in N\} \\ \Omega_I &= \Omega - \Omega_B \end{aligned}$$

where N is a symmetric neighbor set related to the asymmetric neighbor set N_s as $N = \{s : s \in N_s\} \bigcup \{-s : s \in N_s\}$. Specific assumption on $\{y(s)\,,\ s \in \Omega_B\}$ leads to finite lattice representations [23].

Assume that the original image $\{y(s)\}$ is wide sense stationary, with mean $y_m(s) = E[y(s)]$, and obeys the GMRF model [18][19] defined on Ω_I.

$$y(s) - y_m(s) = \Theta^t(\mathbf{y}_s - \mathbf{y}_{m,s}) + e(s) \tag{12}$$

where $\Theta = col[\theta_r, r \in N_s]$, $\mathbf{y}_s = col[y(s+r)+y(s-r), r \in N_s]$, and $\mathbf{y}_{m,s} = col[y_m(s+r)+y_m(s-r), r \in N_s]$. In (12), the stationary Gaussian noise sequence $e(s)$ has the following correlation properties:

$$\begin{aligned} E[e(s)e(r)] &= -\nu\theta_{s-r} \quad (s-r) \in N \\ &= \nu \qquad s = r \\ &= 0 \qquad \text{otherwise} \end{aligned} \tag{13}$$

where ν is the variance of $e(s)$. Using (12) and (13) one can show that [18], $\{\mathbf{y}(s), s\varepsilon\Omega\}$ obeys a 2-D Markov property, i.e. $p(y(s)|$ all $y(r), r \neq s) = p(y(s)|\ y(s+r)\ ,\ r \in N)$.

The GMRF model considered in this section is noncausal, i.e., the intensity $y(s)$ at site s is a function of the neighbors of s in all directions. The GMRF models are characterized by conditional densities of the form [18][19]

$$\begin{aligned} p(y(s)|\mathbf{y}_s) &= \frac{1}{\sqrt{2\pi\nu}} exp\{-\frac{1}{2\nu}[y(s) - y_m(s) \\ &\quad -\Theta^T(\mathbf{y}_s - \mathbf{y}_{m,s})]^2\} \end{aligned} \tag{14}$$

Using the vector-matrix notation, (12) and (13) can then be written as

$$B(\Theta)(\mathbf{y} - \mathbf{y}_m) = \mathbf{e} \tag{15}$$

$$E[\mathbf{e}\mathbf{e}^t] = \nu B(\Theta) \tag{16}$$

respectively, where $B(\Theta)$ is a block-Toeplitz matrix and $\mathbf{y}_m = col[y_m(s), s \in \Omega_I]$. Assuming $B(\Theta)^{-1}$ exists, from (15) and (16) we obtain the covariance matrix of the original image as

$$Q_y = E[(\mathbf{y} - \mathbf{y}_m)(\mathbf{y} - \mathbf{y}_m)^t] = \nu B(\Theta)^{-1} \tag{17}$$

The probability density function of $\mathbf{y}$ can then be written as

$$p(\mathbf{y}) = \{\frac{1}{(2\pi)^{M^2} det(\nu B(\Theta)^{-1})}\}^{\frac{1}{2}} exp\{-\frac{1}{2\nu}(\mathbf{y} - \mathbf{y}_m)^t B(\Theta)(\mathbf{y} - \mathbf{y}_m)\} \tag{18}$$

3.2 Deterministic Relaxation Using Conjugate Gradient Method

If we know the statistical properties of the original image y and the degradations, then we can construct the *aposteriori* density function $p(\mathbf{y}|\mathbf{x})$ from the observation x. Thus Bayes law leads to the description of the *aposteriori* density

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} \tag{19}$$

where y is the original image we wish to estimate. The MAP estimate is obtained by maximizing the *aposteriori* density function $p(\mathbf{y}|\mathbf{x})$ to find the most probable value of y. Since the observation x is given, $p(\mathbf{x})$ can be treated as a constant, we have

$$p(\mathbf{y}|\mathbf{x}) = K p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \tag{20}$$

where constant K is equal to $1/p(\mathbf{x})$.

An image corrupted by multiplicative Gaussian noise can be expressed as

$$x(s) = y(s)n(s) \tag{21}$$

where $n(s)$ is a signal-independent white Gaussian noise of known mean n_m and unknown variance σ_n^2. For a specified y, the variations in x are given by $\mathbf{n} = col[n(s) = x(s)/y(s), s \in \Omega_I]$. Since $n(s)$ is a white Gaussian noise variable, we can express the conditional density function $p(\mathbf{x}|\mathbf{y})$ as

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \frac{1}{[(2\pi)^{M^2}|Q_n|]^{\frac{1}{2}}|J|} exp\{-\frac{1}{2}(\mathbf{n}-\mathbf{n}_m)^t {Q_n}^{-1}(\mathbf{n}-\mathbf{n}_m)\} \\ &= \frac{1}{(2\pi{\sigma_n}^2)^{\frac{M^2}{2}}\prod_s y(s)} exp\{-\frac{1}{2{\sigma_n}^2}\sum_s [\frac{x(s)}{y(s)} - n_m]^2\} \end{aligned} \tag{22}$$

where $Q_n = \sigma_n^2 I$ and $|J|$ is the Jacobian of the transformation from n to x. Since the image y is assumed to be generated by a GMRF model, using $p(\mathbf{y})$ from (18) and taking logarithm, and

noting that $|J| = \prod_s y(s)$ we get

$$\begin{aligned}\ln\, p(\mathbf{y}|\mathbf{x}) = \ln\, K &- \frac{1}{2}\ln\,\{(2\pi)^{M^2} det[\nu B(\Theta)^{-1}]\} \\ &- \frac{1}{2\nu}(\mathbf{y}-\mathbf{y}_m)^t B(\Theta)(\mathbf{y}-\mathbf{y}_m) - \frac{1}{2}\ln\,(2\pi{\sigma_n}^2)^{M^2} \\ &- \frac{1}{2{\sigma_n}^2}\sum_s [\frac{x(s)}{y(s)} - n_m]^2 - \sum_s \ln y(s) \end{aligned} \tag{23}$$

The MAP estimate is found by locating the maximum of the right-hand side in (23). By dropping the constants and the negative sign, the following function is equivalently minimized

$$\Psi(\mathbf{y}) = \frac{1}{2}(\mathbf{y}-\mathbf{y}_m)^t Q_y^{-1}(\mathbf{y}-\mathbf{y}_m) + \frac{1}{2{\sigma_n}^2}\sum_s [\frac{x(s)}{y(s)} - n_m]^2 \; + \; \sum_s \ln y(s) \tag{24}$$

The gradient of Ψ is given as

$$\nabla\Psi(\mathbf{y}) = Q_y^{-1}(\mathbf{y}-\mathbf{y}_m) - \mathbf{v} + \mathbf{w} \tag{25}$$

where

$$\mathbf{w} = col[w(s) = \frac{1}{y(s)}\;,\; s \in \Omega_I]$$

and

$$\mathbf{v} = col[v(s) = \frac{x(s)}{{\sigma_n}^2 y(s)^2}(\frac{x(s)}{y(s)} - n_m), s \in \Omega_I] \tag{26}$$

Substitution of (17) into (24) yields $\nabla\Psi$ in terms of the GMRF model parameters as

$$\nabla\Psi(\mathbf{y}) = \frac{1}{\nu}B(\Theta)(\mathbf{y}-\mathbf{y}_m) - \mathbf{v} + \mathbf{w} = \frac{1}{\nu}\mathbf{e} - \mathbf{v} + \mathbf{w} \tag{27}$$

Equation (27) can also be expressed in point-by-point form as

$$\nabla\Psi(y(s)) = \frac{1}{\nu}[(y(s) - y_m(s)) - \Theta^t(\mathbf{y}_s - \mathbf{y}_{m,s})] - \frac{x(s)}{{\sigma_n}^2 y(s)^2}(\frac{x(s)}{y(s)} - n_m) + \frac{1}{y(s)} \tag{28}$$

The MAP estimate must satisfy the nonlinear equation $\nabla\Psi(\mathbf{y}) = 0$. Since a direct solution to this equation is not possible, one may use a search procedure such as the Polak-Ribiere [11] line search extension of the conjugate gradient method for finding the minimum of $\Psi(\mathbf{y})$. Since the conjugate gradient descent step

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \alpha^{(k)} D_k \tag{29}$$

involves $\alpha^{(k)}$- the optimal step in the k iteration, we determine $\alpha^{(k)}$ using the univariate search at the top of the pyramid. In order to construct the algorithm we need to specify the energy terms corresponding to the different cliques that sum to the global energy function. Using the Gibbs-MRF equivalence, one can easily note that $p(\mathbf{y}|\mathbf{x})$ is Gibbs[2][19]. One can then write $p(\mathbf{y}|\mathbf{x})$ in the form

$$p\,(\mathbf{y}|\mathbf{x}) \;=\; \frac{e^{-U^P(\mathbf{y}|\mathbf{x})}}{Z} \tag{30}$$

where Z is a normalizing factor and $U^P(\mathbf{y}|\mathbf{x})$ is the global energy function that can be broken into a sum of energy terms corresponding to different cliques as

$$U^P(\mathbf{y}|\mathbf{x}) \;=\; \sum_{c \epsilon C} V_c(\mathbf{y}|\mathbf{x})$$

In the case of Gaussian distribution the cliques consist of one or two elements. For the first order GMRF model the cliques are:

$$\{(i,j)\;(i,j+1)\;,\;(i,j)\;(i+1,j)\}.$$

For the second order model we have

$$\{(i,j)\;(i,j+1)\;,\;(i,j)\;(i+1,j)\;,\;(i,j)\;(i+1,j+1)\;,\;(i,j)\;(i+1,j-1)\}$$

For the multiplicative Gaussian noise degradation model in (21), we can express the conditional density function $p(x(s)|y(s))$ as

$$p(x(s)|y(s)) = \frac{1}{\sqrt{2\pi\sigma_n^2 y(s)}} exp\{-\frac{1}{2\sigma_n^2}[\frac{x(s)}{y(s)} - n_m]^2\} \tag{31}$$

Using (14) and (31), we have

$$\begin{aligned} p(y(s)|x(s), \mathbf{y}_s) &= \frac{K}{2\pi\sqrt{\nu\sigma_n^2 y(s)}} exp\{-[\frac{1}{2\nu}((y(s) - y_m(s) \\ &- \Theta^T(\mathbf{y}_s - \mathbf{y}_{m,s}))^2 + \frac{1}{2\sigma_n^2}(\frac{x(s)}{y(s)} - n_m)^2]\} \\ &= \frac{K}{2\pi\sqrt{\nu\sigma_n^2 y(s)}} exp\{-\hat{\mathcal{E}}(y(s))\} \end{aligned} \tag{32}$$

where

$$\begin{aligned} \hat{\mathcal{E}}(y(s)) &= \frac{1}{2\nu}((y(s) - y_m(s) - \Theta^T(\mathbf{y}_s - \mathbf{y}_{m,s}))^2 \\ &+ \frac{1}{2\sigma_n^2}(\frac{x(s)}{y(s)} - n_m)^2 \end{aligned}$$

Thus the energy function is:

$$U^P(y(s)|x(s), \mathbf{y}_s) = \mathcal{E}(y(s)) = \hat{\mathcal{E}}(y(s)) + \ln(y(s)) \tag{33}$$

From (32) one can note that the zeroth order energy terms in the expansion of the energy function $U^P(\mathbf{y}|\mathbf{x})$ (including additive constant terms) are of the form

$$\frac{1}{2\nu}(y(s) - y_m(s))^2 + \frac{1}{2\sigma_n^2}\left(\frac{x(s)}{y(s)} - n_m\right)^2 + \ln(y(s)) \ , \ \forall s \in \Omega_I$$

The energy terms corresponding to the first and second order cliques are

$$\frac{1}{\nu}\Theta^T(\mathbf{y}_s - \mathbf{y}_{m,s})\ y(s) \ , \ \forall s \in \Omega_I$$

We now perform the line search conjugate gradient algorithm similar to the one presented in the former section but with a fixed resolution.

3.3 Experimental Results

The experiments in this section were performed on USC girl image in Figure 1, corrupted by multiplicative Gaussian noise. When we used Polak-Ribiere method on this problem, we realized that the search vectors got very small and the gradients lost orthogonality. In order to overcome the problem, we looked into a different line search extension of conjugate gradient method that includes a restart procedure, as suggested by Powell [24]. The algorithm can be summarized as follows:

- Step1. For $k = 1$
  - a)Given X_1 compute $G_1 = \nabla f(X_1)$
  - b)Set $t = 1$,$D_1 = -G_1$
  - c)Choose ζ (orthogonality parameter).
  - d)Go to step 3
- Step2. For $k = 2, \cdots, n$:
  - a) Compute $G_k = \nabla f(X_k)$.
  - b) If $\frac{|G_{k-1}^T G_k|}{|G_k^T G_k|} \geq \zeta$ then $t = k - 1$.
  - c) Set $\beta_k = \frac{G_k^T (G_k - G_{k-1})}{D_{k-1}^T (G_k - G_{k-1})}$
  - d) If $k = t + 1$ then

    set $\gamma_k = 0$ compute D_k, where $D_k = -G_k + \beta_k D_{k-1} + \gamma_k D_t$

    go to step 3

else

compute γ_k and D_k where $\gamma_k = \frac{G_k^T(G_{t+1}-G_t)}{D_t^T(G_{t+1}-G_t)}$

(now check that the search direction D_k is sufficiently downhill)

if $D_k^T G_k \leq -0.8 G_k^T G_k$ continue, else set $t = k-1$ and go to 2.d

- Step 3

  - a) Set $X_{k+1} = X_k + \alpha_k D_k$ where α_k minimizes $f(X_k + \alpha D_k)$.

  - b) If the terminating condition holds stop, else go back to Step 2

Note that the calculation of the optimal step in Powell algorithm is performed exactly as in the former method. Powell algorithm converged in 20% less iterations than the Polak-Ribiere method. As the energy function includes a log term, in order to avoid numerical problems near zero, optimization was performed on a shifted version of the original image (one gray level up). The solutions were also constrained during the search to the compact set $1 \leq x_i \leq 256$. In order to compute the GMRF parameters the global estimate of the mean was subtracted from the original image. The least square (LS) estimates [25] of a second order GMRF model with $N_S = \{(0,1), (1,0), (-1,1), (1,1)\}$ were then computed. It was found that the LS estimates $\hat{\Theta}$ were unstable, i.e. , some of the eigenvalues of the transformation matrix $B(\hat{\Theta})$ were found to be negative. The LS estimates were rescaled by dividing the vector $\hat{\Theta}$ by $\sum_{r \in N_s} \hat{\Theta}_r$. The rescaled estimates were then used as the initial conditions for the maximum likelihood estimation algorithm [20][21][22] .

Since the restoration with the estimated parameters smoothed important details in the reconstructed image, we used an estimated noise variance smaller than the actual one. It gives better subjective results, because important details like edges are not smeared. We tried several initial conditions and the results were the same. Figure 2.B is the initial conditions for both experiments. The original image corrupted by multiplicative Gaussian noise with variances 0.0712 and 0.2217 (corresponding to additive noise of 5dB and 0dB), are in Figures 2.C and 2.D respectively. Figure 2.E is the result after 200 iterations for the 5dB image, using the estimated noise variance of 0.024 (instead of 0.0712). Figure 2.F is the result after 200 iterations for the 0dB image, using the estimated noise variance of 0.07 (instead of 0.2217).

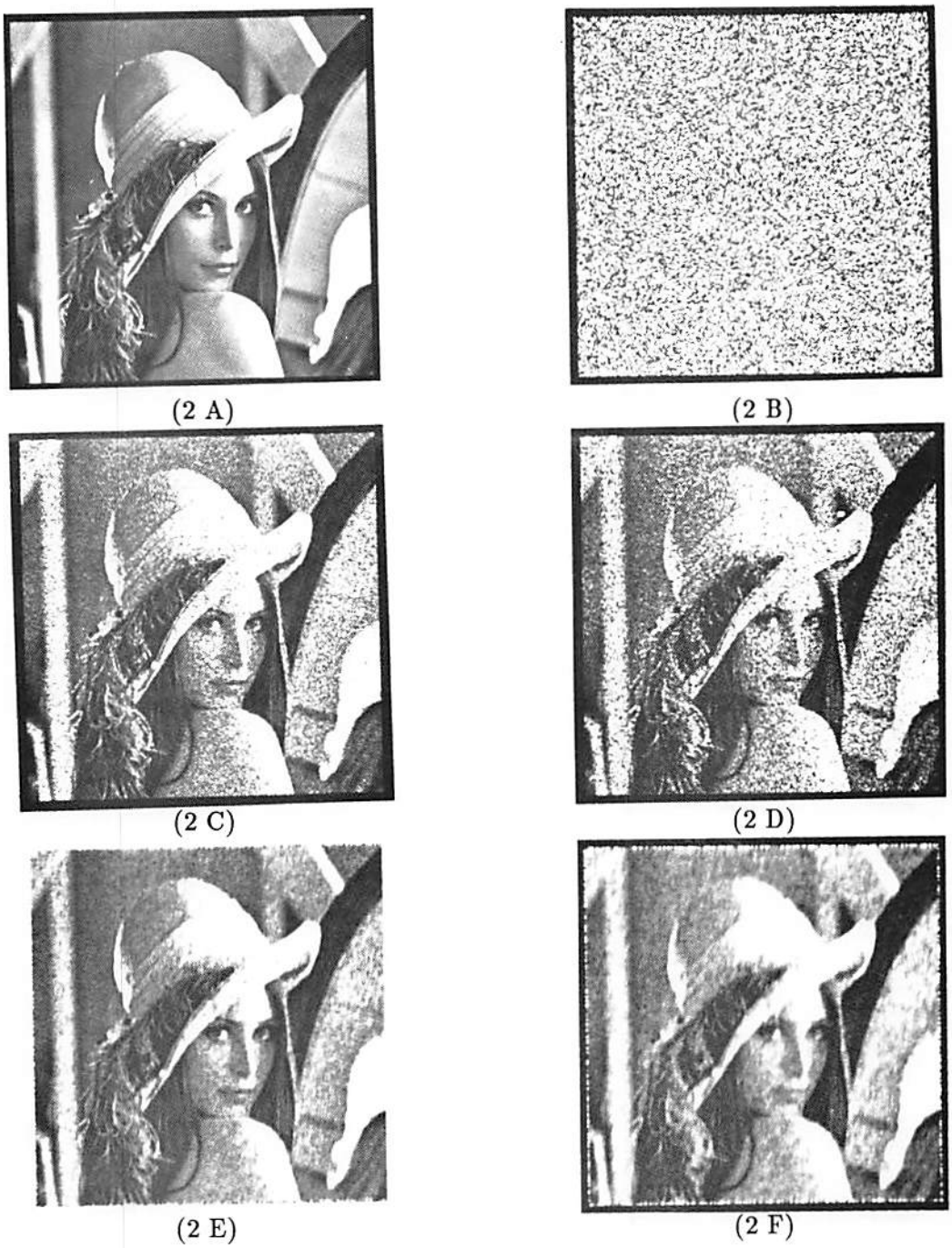


Figure 2: MAP estimation of multiplicative Gaussian noisy images using CG algorithm.

2 A: Original USC girl image ;
2 B: 0 dB noise (used as initial conditions) ;
2 C: Image corrupted by 5 dB multiplicative noise ;
2 D: Image corrupted by 0 dB multiplicative noise ;
2 E: CG estimate for the 5 dB noise ;
2 F: CG estimate for the 0 dB noise ;

4 Edge Detection Using Blakes' Graduated Non-Convexity Algorithm

Edge detection can be viewed as the problem of fitting a weak membrane (that is, an elastic membrane under weak continuity constrains) to a surface [9]. The location of discontinuities in the weak membrane correspond to discontinuities in the intensity (step edges). The problem has the following mathematical form: Find the configuration that minimizes the following functional corresponding to a weak membrane with a simple line process:

$$F = \int \{(u-d)^2 + \lambda^2(\nabla u)^2\} dA + \alpha \int dl \tag{34}$$

The first integral is evaluated over the area in which the data d is defined, and the second along the length of all discontinuities. α is a penalty per unit length of discontinuity. λ is a characteristic length for smoothing the continuous portions of the data and is also a characteristic distance for interaction between discontinuity. Since the data is only available on grid points, and the problem does not have a closed form solution, we need to discretize the problem. Blake [9] suggests minimizing the cost function.

$$F = D + \sum_{ij} h_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}, l_{ij}) + \sum_{ij} h_{\alpha,\lambda}(u_{i,j} - u_{i,j+1}, m_{ij}) \tag{35}$$

where

$$D = \sum_{ij} (u_{i,j} - d_{i,j})^2 \tag{36}$$

l_{ij} activates the line process in the northernly direction and

$$h_{\alpha,\lambda}(t,l) = \lambda^2(t)^2(1-l) + \alpha l$$

m_{ij} activates the line process in the easternly direction and

$$h_{\alpha,\lambda}(t,m) = \lambda^2(t)^2(1-m) + \alpha m$$

The problem is thus reduced to the following optimization problem.

$$\min_{\{u_{ij}\}} \left(D + \min_{\{l_{ij}\}} \left(\sum_{ij} h_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}, l_{ij}) \right) + \min_{\{l_{ij}\}} \left(\sum_{ij} h_{\alpha,\lambda}(u_{i,j} - u_{i,j+1}, m_{ij}) \right) \right)$$

As D does not involve l_{ij}, m_{ij}, minimization over l_{ij}, m_{ij} can be performed and one is then left with minimization with respect to u_{ij}:

$$\min_{\{u_{ij}\}} F = \min_{\{u_{ij}\}} \left(D + \sum_{ij} g_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}) + \sum_{ij} g_{\alpha,\lambda}(u_{i,j} - u_{i,j+1}) \right)$$

where

$$g_{\alpha,\lambda}(t) = \min_{l \in \{0,1\}} h_{\alpha,\lambda}(t,l) = \min(\lambda^2(t)^2, \alpha)$$

By applying a convex approximation to F Blake obtains:

$$F^* = D + \sum_{ij} g^*_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}) + \sum_{ij} g^*_{\alpha,\lambda}(u_{i,j} - u_{i,j+1}) \tag{37}$$

where

$$g^*_{\alpha,\lambda}(t) = \begin{cases} \lambda^2(t)^2, & |t| < q \\ \alpha - c^*(|t| - r)^2/2, & q \le |t| < r \\ \alpha, & |t| \ge r \end{cases}$$

where

$$r^2 = \alpha\left(\frac{2}{c^*} + \frac{1}{\lambda^2}\right), \quad q = \frac{\alpha}{\lambda^2 r} \text{ and } c^* = \frac{1}{4} \text{ for membrane}$$

A one parameter family of cost functions $F^{(P)}$ is then defined by replacing g^* in (37) by $g^{(P)}$. $g^{(P)}$ is similar to g^* except that c^* is replaced by a variable c, that varies with P.

$$F^{(P)} = D + \sum_{ij} g^{(P)}_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}) + \sum_{ij} g^{(P)}_{\alpha,\lambda}(u_{i,j} - u_{i,j+1}) \tag{38}$$

with

$$g^{(P)}_{\alpha,\lambda}(t) = \begin{cases} \lambda^2(t)^2, & |t| < q \\ \alpha - c(|t| - r)^2/2, & q \le |t| < r \\ \alpha, & |t| \ge r \end{cases}$$

$$\text{where } c = \frac{c^*}{P}, \quad r^2 = \alpha\left(\frac{2}{c} + \frac{1}{\lambda^2}\right), \quad \text{and } q = \frac{\alpha}{\lambda^2 r}$$

The GNC algorithm begins by minimizing $F^{(P=1)} = F^*$. Then P is decreased from 1 to 0, which makes $g^{(P)}$ change steadily from g^* to g. For every value of P we minimize $F^{(P)}$ starting with the last configuration corresponding to the previous P (local minimum of $F^{(2P)}$). We suggest that minimization of $F^{(P)}$ can be performed efficiently using the optimal step conjugate gradient algorithm described in section 2. The conjugate gradient algorithm takes fewer number of iterations than SOR, but each iteration requires sub-steps to determine the optimal step. The conjugate gradient search is not sensitive to the noise in the processed image. The number of iterations increases only by 40% in the presence of 0 dB noise, while the SOR algorithm requires double number of iterations. We view $F^{(P)}$ as a Gibbs energy function for the global configuration. The energy function is then broken into terms that correspond to cliques of a nearest neighbor system. The zero order terms are of the form:

$$(u_{ij} - d_{ij})^2$$

The energy terms corresponding to cliques of two elements e.g. $(i,j)(i,j+1)$ is of the form:

$$g^{(P)}_{\alpha,\lambda}(t) = \begin{cases} \lambda^2(t)^2, & |t| < q \\ \alpha - c(|t| - r)^2/2, & q < |t| < r \\ \alpha, & |t| > r \end{cases}$$

Note that the energy term is a nonlinear function of the cliques' elements. We proceed by calculating the gradient of $F^{(P)}$ for points in the interior:

$$\begin{aligned}\frac{\partial F^{(P)}}{\partial u_{ij}} &= 2(u_{i,j}-d_{i,j})+g^{(P)'}_{\alpha,\lambda}(u_{i,j}-u_{i,j-1}) \\ &+ g^{(P)'}_{\alpha,\lambda}(u_{i,j}-u_{i-1,j})+g^{(P)'}_{\alpha,\lambda}(u_{i,j}-u_{i,j+1}) \\ &+ g^{(P)'}_{\alpha,\lambda}(u_{i,j}-u_{i+1,j})\end{aligned} \tag{39}$$

where

$$g^{(P)'}_{\alpha,\lambda}(t)=\begin{cases}2\lambda^2(t), & |t|<q \\ -c(|t|-r)\text{sign}(t), & q\le|t|<r \\ 0, & |t|\ge r\end{cases}$$

On the boundary we need a modification, for the corner $i=0, j=0$:

$$\begin{aligned}\frac{\partial F^{(P)}}{\partial u_{00}} &= 2(u_{0,0}-d_{0,0})+g^{(P)'}_{\alpha,\lambda}(u_{0,0}-u_{0,1}) \\ &+ g^{(P)'}_{\alpha,\lambda}(u_{0,0}-u_{1,0})\end{aligned} \tag{40}$$

and similarly at the other corners

For the boundary side $i=0,\ j=1,\cdots,N-1$:

$$\begin{aligned}\frac{\partial F^{(P)}}{\partial u_{ij}} &= 2(u_{0,j}-d_{0,j})+g^{(P)'}_{\alpha,\lambda}(u_{0,j}-u_{0,j-1}) \\ &+ g^{(P)'}_{\alpha,\lambda}(u_{0,j}-u_{0,j+1})+g^{(P)'}_{\alpha,\lambda}(u_{0,j}-u_{1,j})\end{aligned} \tag{41}$$

and similarly for the other sides.

We now have all the building blocks required for our algorithm, which is now presented:

1. choose λ, h_0 (scale and sensitivity).

2. set $\alpha = h_0^2\lambda/2$.

3. set $P = 1.0$

4. given the initial configuration u^0 compute $G_0 = \nabla F^P(u^0)$ and set $D_0 = -G_0$.

5. calculate (using a univariate search) the optimal step for the given D_k using the global energy function (at the top of the pyramid).

6. perform the descent step

$$u^{(k+1)} = u^{(k)} + \alpha_k D_k$$

7. calculate the gradient vector $G_{k+1} = \nabla F^P(u^{(k+1)})$ (at the bottom of the pyramid)

8. calculate a new conjugate direction $D_{k+1} = -G_{k+1} + \beta_k D_k$ where

$$\beta_k = \frac{(G_{k+1} - G_k)^T G_{k+1}}{G_k^T G_k}$$

9. repeat steps 5-8 until $\max_{i,j} |u_{i,j}^{(k+1)} - u_{i,j}^{(k)}| < \epsilon$, or the energy reduction is below some level.

10. if $P > \frac{c^*}{\lambda}$ then $P = P/2$ $u^0 = u^{(k+1)}$ go back to step 4.

11. calculate the edges location using the following rules:

$$l_{i,j} = \begin{cases} 1 & \text{if } |u_{i,j} - u_{i-1,j}| > r \\ 0 & \text{if } |u_{i,j} - u_{i-1,j}| < q \\ \text{ambiguous} & \text{otherwise} \end{cases}$$

$$m_{i,j} = \begin{cases} 1 & \text{if } |u_{i,j} - u_{i,j+1}| > r \\ 0 & \text{if } |u_{i,j} - u_{i,j+1}| < q \\ \text{ambiguous} & \text{otherwise} \end{cases}$$

$l_{i,j}$ and $m_{i,j}$ are the edge indicator functions in the northernly and easternly directions respectively.

4.1 Experimental Results

4.1.1 Edge detection using Conjugate Gradient

The experiments in this section were performed on a 128 × 128 Airport image. We used the Polak-Ribiere conjugate gradient search. Edge detection was performed on the original image and the orginal image corrupted by additive gaussian noise. The conjugate gradient method proved to be robust to noise in the sense that the convergence properties were not affected considerably by the amount of noise in the processed image. In each step of the GNC algorithm the termination criterion used was : stop if $f(x_k) - f(x_{k+1}) \leq 100$. We prefered this test over the one : stop when $\| u_{k+1} - u_k \|_\infty \leq a$ (where a is some constant), because it requires a smaller number of iterations without degradation in the computed edges. The GNC steps were performed till $p = \frac{c^*}{\lambda}$.

Experimental results are summerized in Figure 3. Figures 3.A and 3.B are the original Airport image and the result obtained with $h = 10$, $\lambda = 4$ after 157 iterations. Figures 3.C and 3.D are the image with 5dB noise ($\sigma_n = 12$), and the result obtained with $h = 16$, $\lambda = 4$, where the number of iteration increased to 198. Figures 3.E and 3.F are the image with 0dB noise ($\sigma_n = 22$), and the result obtained with $h = 20$, $\lambda = 4$ with the algorithm terminating after 224 iterations.

4.1.2 Comparison between Conjugate Gradient and SOR

We first compared the sequential SOR and the Black and Red (parallel) SOR. The number of iterations and the final results were the same for these two algorithms. In order to

compare the SOR and the conjugate gradient we used several images with different noise levels. The number of iterations in the SOR increases considerably when the noise level is increased, while the number of iterations in the conjugate gradient method is not so sensitive to the noise. The number of iterations in the SOR is greater than that of the conjugate gradient, but in each conjugate gradient step we perform more computations. The final results for these two methods are the same for all the noise levels. In the following Table we summarize the results.

Table 2. Comparison between Conjugate Gradient and SOR Algorithms

image	algorithm	iterations	final energy$\times 10^6$	undefined $l_{i,j}$
airport	SOR	328	1.3101	336
	CG	157	1.3088	336
airport	SOR	477	4.1830	279
with 5dB noise	CG	198	4.1814	253
airport	SOR	617	8.9788	244
with 0dB noise	CG	224	8.9722	227

In Figure 4 we present the comperison results for the two algorithms. Figures 4.A and 4.B are the results from the 5dB and 0dB images using SOR algorithm, where Figures 4.C and 4.D were obtained by using the conjugate gradient algorithm.

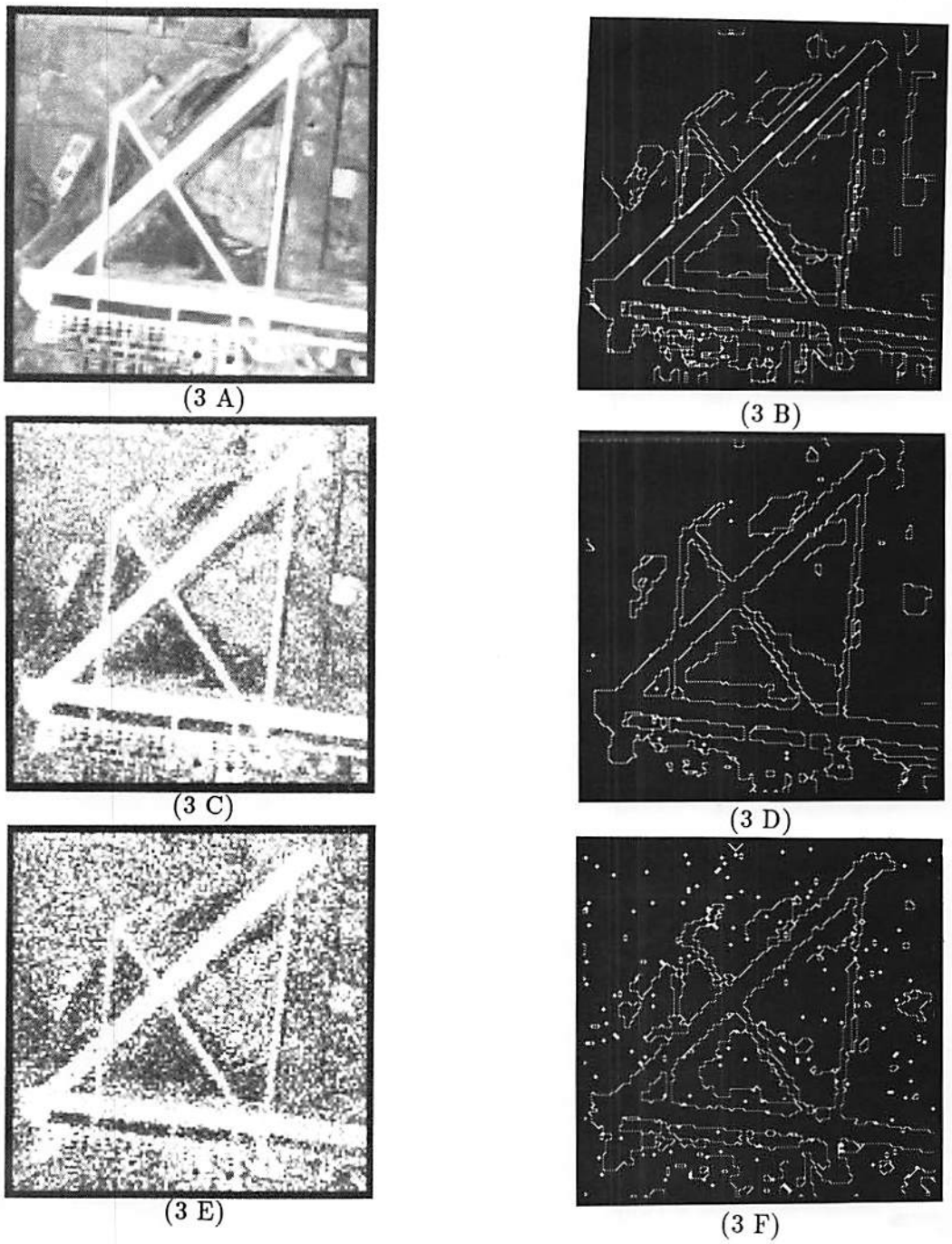
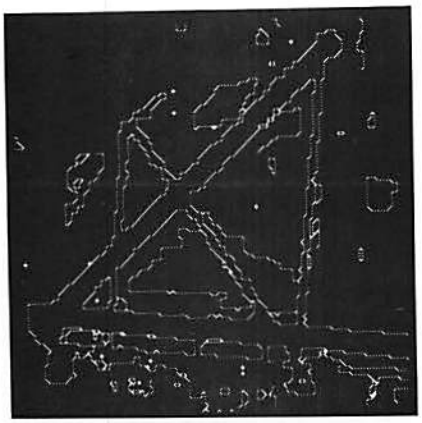
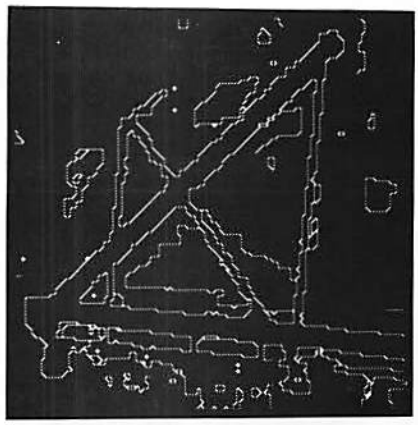


Figure 3: Edge Detection Using Conjugate Gradient Based GNC Algorithm.

3 A: Original airport image ;
3 B: Edge estimate on the original airport image.
3 C: Airport image corrupted by 5 dB additive noise;
3 D: Edge estimate on the 5 dB noisy airport image.
3 E: Airport image corrupted by 0 dB additive noise;
3 F: Edge estimate on the 0 dB noisy airport image.



(4 A)



(4 B)

(4 C)

(4 D)

Figure 4: Comparison Between SOR and Conjugate Gradient GNC Results.

4 A: SOR edge estimate on 5dB image 3C ; 4 B: CG edge estimate on 5dB image 3C ;
4 C: SOR edge estimate on 0dB image 3E ; 4 D: CG edge estimate on 0dB image 3E ;

References

[1] B. K. P. Horn, *"Robot Vision"*, The MIT Press, Cambridge,Massachusetts, 1986.

[2] S. Geman and D. Geman, "Stochastic Relaxation ,Gibbs Distributions ,and Bayesian Restoration of Images", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol.6, pp. 721–741, November 1984.

[3] T. Poggio and V. Torre, "Ill posed Problems and Regularization Analysis in Early Vision", *Artificial Intelligence Lab , M.I.T*, A.I.M 773, 1984.

[4] J. Hadamard, *"Lectures on the Cauchy Problem in Linear Partial Differential Equations"*, Yale University Press, New Haven, 1923.

[5] R. Courant and D. Hilbert, *"Methods of Mathematical Physics"*, Volume vol. 1, Interscience, New York, 1953.

[6] M. J. Brooks and B. K. P. Horn, Shape and Source From Shading, In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 932–936, Los Angeles,California, August 1985.

[7] M. Strat, *"A Numerical Method for Shape From Shading from a Single Image"*, M.S. thesis, M.I.T, Department of Elect. Engrg. and Comp. Sci., 1979.

[8] D. Lee, "A Provably Convergent Algorithm for Shape from Shading", In *Proc. DARPA Image Undestanding Workshop*, pages 489–496, Miami Beach,Florida, December 1985.

[9] A. Blake and A. Zisserman, *"Visual Reconstruction"*, MIT Press, Cambridge, 1987.

[10] T. Simchony et al, "MAP Estimation of Gray Level Images Corrupted by Multiplicative and Poisson Noise", (Submitted for Publication).

[11] D. Luenberger, *"Linear and Nonlinear Programming"*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

[12] D. Terzopoulos, "Image Analysis Using Multigrid Relaxation Methods", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol.8, pp. 129–139, March 1986.

[13] J. Besag, "On the Statistical Analysis of Dirty Pictures", *Journal of Royal Statistic Society B*, vol. 48 No. 6, pp. 259–302, 1986.

[14] T. Simchony and R. Chellappa, "Direct Analytical Methods for Solving Poisson Equations in Computer Vision Problems", In *IEEE Computer Society Workshop on Computer Vision Problems*, Miami Beach, Florida, November 1987.

[15] S. Geman and C. Hwang, "Diffusions For Global Optimization", *SIAM Journal of Control and Optimization*, vol.24 no.5, pp. 1031–1043, September 1986.

[16] L.E. Scales, *"Introduction to Non-Linear Optimization"*, Spinger-Verlag, New York, 1985.

[17] B. R. Hunt, "Bayesian methods in nonlinear digital image restoration ", *IEEE Trans. Computers*, vol.CAS-26, pp. 219–229, 1977.

[18] J. W. Woods, "Two-dimensional discrete Markovian fields", *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 232–240, March 1972.

[19] J. Besag, "Spatial Interaction and the Statistical Analysis of Latice Systems", *Journal of Royal Statistic Society B*, No. 2, pp. 192–236, 1974.

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

[21] H. Kunsch, "Thermodynamics and statistical analysis of Gaussian random fields", *Z. Wahr. Ver. Geb.*, vol. 58, pp. 401–421, November 1981.

[22] G. Sharma and R. Chellappa, "A model-based approach for estimation of two-dimensional maximum entropy power spectra", *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 90–99, January 1985.

[23] R.L. Kashyap, "Random field models on finite lattices for finite images", In *Proc. of the Conf. on information sciences and systems*, pages 215–220, Johns Hopkins University, March 1981.

[24] M.J.D Powell, "Restart Procedures for the Conjugate Gradient Method", *Mathematical Programing*, vol.12, pp. 241–254, 1977.

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