Statistical Image Restoration via the Ising Model

Martin Charles Kandes[1,2] *
[1] Computational Science Research Center, San Diego State University, San Diego, California
[2] Department of Physics, San Diego State University, San Diego, California

Abstract

We present preliminary numerical results on a statistical mechanics approach to image restoration using the well-known Ising model. Both binary and grayscale image restoration are considered by employing the binary (or spin-1/2) Ising model in two and three dimensions, respectively. Using the Metropolis algorithm, we examine how external bias fields, nearest-neighbor interactions, and the lattice temperature impact the image restoration process. We find that the binary Ising model in both two and three dimensions can be useful for statistical image restoration when the lattice temperature is well below the critical temperature required for the onset of spontaneous magnetization in the field-free limit. In the case of binary image restoration, it is observed that nearest-neighbor interactions can play a significant role in the restoration process. However, for grayscale image restoration, their role is less clear and more complex to decipher. In general, we find that a quasi-two-dimensional Ising model is more suited for this task. In all cases, as expected, the quality of the restored image relies heavily upon the requirement to generate an accurate approximation of the original image from the degraded one. Despite the limited success of the preliminary results presented here, we argue that the generalized framework of the binary Ising model as it relates to the image restoration process is both quite unique and merits further study.

Keywords: image restoration, image degradation, image denoising, ising model, statistical mechanics

1 Introduction

The general problem of image restoration is to reconstruct the original, uncorrupted image of a scence given only the information available in a noisy, degraded version of the image. In general, all image restoration techniques rely on some a priori knowledge of the general statistical properties of images and how the various sources of noise may distort the pixels of an image when present.

The problem of statistical image restoration is two-fold. First, these methods often rely heavily on generating an accurate approximation of the original image using only information available in the degrade one. This is typically the most challenging aspect of these methods due to the astronomical number of the possible pixel configurations. The second issue is somewhat more tractable. Here, one must identify the type of noise that is degrading the image and estimate the magnitude of it's effect on the original image's pixels. When there exists a well-known model to describe a particular type of noise that is present in an image, the problem of denoising can be rather straightforward. However, if this task proves to be difficult, simple assumptions about the basic properties of images are often used instead. One such assumption commonly used is the assumption of smoothness in real world images. For example, consider an isolated white pixel located among a set of black ones. Statistically, this configuration is likely to have been caused by noise rather than to have existed in the original image and it should be suppressed. This set of local interactions and correlations among the neighboring pixels in an image is the basis for the use of spin-lattice models to attack the problem of statistical image restoration.

In recent years, spin-lattice-based models have been applied to the problem of statistical image restoration with some success [Tanaka 2002]. These models include the chiral Potts model [Carlucci and Inoue 1999] and the Q-Ising model [Inoue 2001; Inoue and Carlucci 2001; Tanaka et al. 2003], both of which are borrowed from the study of spin glasses in statistical mechanics. The mathematical justification for applying these advanced spin-lattice models to the image restoration process and how they are intended to operate on noisy images is deeply rooted in the theory of Bayesian statistics. A formal discussion of these advanced models is far beyond the scope of this work and we mention them here only to serve as motivation for the work to be presented here.

In this paper, we present preliminary numerical results on a similar statistical mechanics approach to image restoration using the well-known Ising model. Both binary and grayscale image restoration are considered by employing the binary (or spin-1/2) Ising model in two and three dimensions, respectively. Using the Metropolis algorithm, we examine how external bias fields, nearest-neighbor interactions, and the lattice temperature impact the image restoration process.

2 Ising Model

The Ising model is a mathematical model originally developed to simulate the physical and structural properties of ferromagnetic materials. Invented in 1925 by the physicist Ernst Ising [Ising 1925], this model has become a principal archetype in the field of statistical mechanics for studying the physics of phase transitions and critical phenomena. This popularity is largely derived from the fact that analytic techniques can be used to yield exact solutions of the Ising model in both one and two dimensions under certain conditions [Pathria 1996]. Additionally, the generalized framework of Ising model, in which interacting pairs of discrete objects may be used to exhibit macroscopic, collective behavior, has led it to be applied to a rich and diverse set of other types of problems over the years, including the study of lattice gases, binary alloys, neuron interactions, and, of course, image restoration.

In the two-dimensional square-lattice Ising model, we consider a finite set of discrete variables knowns as spins arranged over a regular square-lattice (see Figure 1). Associated with each lattice site is a spin variable S_{ij}, where the indicies on the spin variable represent it's location on the two-dimensional lattice. For the binary Ising model, each spin variable may only assume a value of either $+1$ or -1. If $S_{ij} = +1$, the ijth lattice site is said to have spin up, while if $S_{ij} = -1$, then it is said to have spin down. When the spins on the lattice are allowed interact with one another, nearest-neighbor interaction schemes (see Figure 1) are typically used to approximate their effects on the configuration of the spin-lattice.

*e-mail: mkandes@sciences.sdsu.edu

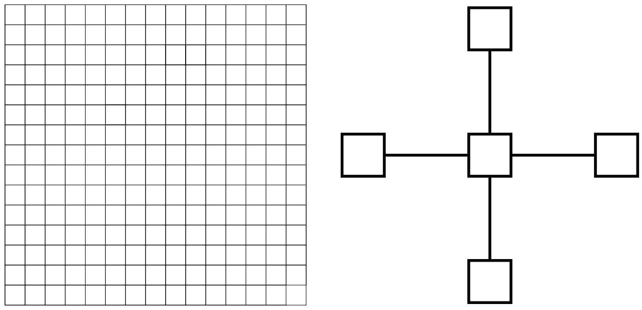

Figure 1: *(Left) A* 15×15 *regular square-lattice in two dimensions. The so-called lattice sites are located at the interestion points of the grid lines. Associated with eace lattice site is a single spin variable. When the interactions between spins on the lattice are first-order nearest-neighbor interactions, the grid lines connecting the lattice sites accurately depict this interaction scheme across the entire lattice. (Right) First-order nearest-neighbor interactions in two dimensions. A spin, centrally located on a lattice site, is coupled only to it's nearest horizontal and vertical neighbors.*

The total energy of the spin-lattice for a given configuration of spins is defined by

$$E = -\sum_{ij} \mu_{ij} H_{ij} S_{ij} - \sum_{\langle ijkl \rangle} J_{ijkl} S_{ij} S_{kl}, \tag{1}$$

where μ_{ij} is a coupling constant that sets the interaction strength between an individual spin and an externally applied magnetic field, H_{ij}, that may be acting on the system. J_{ijkl} is a coupling constant that sets the interaction strengths between locally coupled spins. The brackets $\langle\rangle$ over the sum of spin-spin interactions is used to designate that it extends only over nearest-neighbor pairs of spins. For each pair of spins, if $J_{ijkl} > 0$, then the spin-spin interaction is called ferromagnetic. If $J_{ijkl} < 0$, the interaction is referred to as antiferromagnetic. When $J_{ijkl} = 0$, there is no interaction between the two spins.

In the context of statistical mechanics, the principal problem that must be solved for the Ising model is to find the ground state spin-lattice configuration, i.e. to find the lowest energy configuration of the lattice at a specific inverse lattice temperature, β, given some initial spin-lattice configuration $\{S_{ij}\}$, the sets of coupling constants $\{\mu_{ij}\}$ and $\{J_{ijkl}\}$, and an externally applied magnetic field $\{H_{ij}\}$. Notice that when the interactions between two spins are ferromagnetic, it is energetically favorable for the spins to be aligned with one another. In fact, when all of the spin-spin interactions in the lattice are ferromagnetic, we find that above a certain critical inverse lattice temperature, β_c, the spin-lattice undergoes a so-called phase transition. Here, the ground state spin-lattice stablizes and large, locally connected regions of the lattice begin to align with one another due to collective interactions arising from the long range correlations between the spins. In the absence of an externally applied magnetic field, i.e. $\{H_{ij} = 0\}$, this cooperative phenomenon is known as spontaneous magnetization. Hoever, more importantly, this collective alignment of different regions of the lattice is the precise property we aim to exploit in the binary and grayscale image restoration process. Thus, we expect to the image restoration process to occur for inverse lattice temperatures well above this critical temperature.

Computing the ground state spin-lattice configuration is typically a non-trivial task. Like the large number problem of pixel configurations in statistical image restoration, the number of possible configurations for the spin-lattice is rather daunting to consider. To efficiently sample the configuration space and find an approximate ground state of the spin-lattice, we use the so-called Metropolis Monte Carlo algorithm [Metropolis et al.]. The algorithm is as follows: (1) Choose a spin on the lattice at random. (2) Flip the spin. Spin up goes to spin down. Spin down goes to spin up. (3) Compute the energy difference between the previous configuration of the spin-lattice and the new configuration, that is compute

$$\Delta E = E' - E, \tag{2}$$

where E' is the energy of the new lattice configuration and E is the energy of the previous lattice configuration. (4) If $\Delta E \leq 0$, then keep the new, lower energy spin-lattice configuration. (5) If $\Delta E \geq 0$, then draw a random number R between 0 and 1. (6) Accept the new spin-lattice configuration if and only if

$$R \leq e^{-\beta \Delta E}. \tag{3}$$

(7) Otherwise, keep the old spin-lattice configuration and repeat the process for the next randomly chosen spin. The key idea behind the Metropolis algorithm is that as we randomly walk through the spin-lattice configuration space, we systematically reject the majority of steps that increase the energy of the lattice configuration, allowing a relatively quick convergence to the ground state solution. In Appendix B, the FORTRAN source code of the programs written to find the approximate ground state configurations of the two- and three-dimensional Ising models via the Metroplois algorithm are given.

In the context of binary image restoration, we propose to use the two-dimensional binary Ising model as follows. Consider only unit interactions on the lattice, such that the sets of coupling constants are $\{\mu_{ij} = 1\}$ and $\{J_{ijkl} = 1\}$, and let the degraded binary image be the initial configuration of the spin-lattice $\{S_{ij}\}$. We then set the externally applied magnetic field term, $\{H_{ij}\}$, to be some approximation of the original image obtained by other means. By using the approximated original image as the external bias field, it follows that it is energtically favorable for the spins of the degraded image to become aligned with this approximated original image. Via spin-spin interactions, the information in the approximated bias field can then also be distributed in a correlated way among the spins on the lattice. It is our hope that these interactions will compensate for the undoubtably imperfect approximation of the original image. The final restored image is the approximated ground state spin-lattice configuration found using the Metropolis algorithm.

For grayscale images, the restoration process proposed is similar. However, instead of using the two-dimensional Ising model, we employ the three-dimensional binary Ising model to couple together the eight two-dimensional bit-planes that represent the pixels of the grayscale image.

3 Numerical Results

For the numerical results on binary and grayscale image restoration using the two- and three-dimensional binary Ising model presented here, the emphasis was to study the general mechanics of the image restoration process and the resulting qualitative features of the restored images. As such, we have not focused on how to construct an efficient and accurate approximation of the original image from the degraded ones to be used as the external bias field. Instead, we consider only the two extremum cases when the external bias field is taken to be the exact, original binary or grayscale image and when the external bias field is taken to be the degraded image itself. The idea for using the degraded image as the external bias field assumes that a large fraction of the pixels are uncorrupted and thus retain some information about the original image the spin-spin interactions may use in the restoration process.

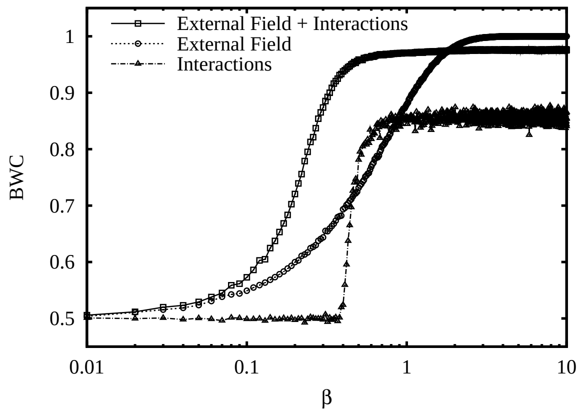

Figure 2: *Binary image restoration of the degraded binary image of Lena ($\sigma^2 = 0.1$) versus the inverse lattice temperature. In this case, the original binary image of Lena was used as the external bias field. Hence, the near perfect image reconstruction under certain conditions when the bias field is applied.*

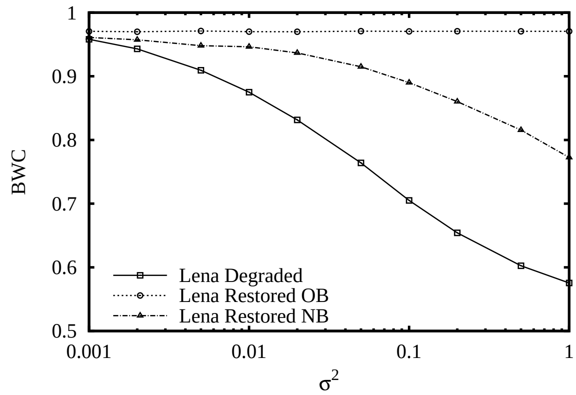

Figure 3: *Binary image restoration of the degraded binary images of Lena at an inverse lattice temerature of $\beta = 1$ versus the magnitude of the gaussian noise variance, σ^2, applied to create the degraded images. The solid line and open squares are the measured quality of the degraded images of Lena with respect to the original image. The dotted line and open circles are the measured quality of the restored images generated by using the original binary image as the external bias field. The dashed-dotted line and the open triangles are measured quality of the restored images generated by using the degraded images as the external bias fields. Note that both the external bias field and spin-spin interactions were used in the restoration process.*

The degraded images to be restored were generated by applying varying degrees of gaussian noise to an original grayscale image as measured by the gaussian noise variance, σ^2. To generate the noisy binary images, the degraded grayscale images were then converted to binary format using a threshold function on the grayscale pixel values. The MATLAB scripts written to prepare the degraded binary and grayscale images are available in Appendix A.

The quality of each restored binary or grayscale image was measured by computing a bitwise correlation (BWC) between the bit sequences of the restored and original image, i.e.

$$BWC = \frac{Number\ of\ bits\ that\ are\ the\ same}{Total\ number\ of\ bits} \tag{4}$$

The FORTRAN program written to compute this bitwise comparison between bit sequences is available in Appendix B.

3.1 Binary Image Restoration

Figure 1 shows the quality of binary image restoration obtained via the two-dimensionl Ising model for different values of the inverse lattice temperature, β. Three different types restoration were considered: (1) when an external bias field is applied and spin-spin interactions are present; (2) when o nly an external bias field is applied, i.e. when there are no spin-spin interactions on the lattice; and (3) when only spin-spin interactions are present, i.e. when no external bais field is applied. As expected, we find that the quality of the restored image increases considerably above the critical inverse temperature required for the onset of spontaneous magnetization. Furthermore, we see that even when no external field is applied and only spin-spin interactions are considered, a large fraction of the bits realign themselves with those of the original image.

Figure 2 shows the quality of binary image restoration that is obtained at an inverse temperature of $\beta = 1$ for varying degrees of noise degraded images when both an external bias field and spin-spin interactions are used in the restoration process. Here, we see that when the original image is used as the external bias field (OB), near perfect image restoration is possible for all levels of noise degraded images. In constrast, when the noisy, degraded image is used as the external bias field (NB), the restoration is always less than perfect, with the quality of the restored image decreasing with increasing noise level. However, even in this case, the quality of the restored images with respect to the original image are greater than that of the degraded ones.

In Figure 3, the qualitative features of the binary image restoration process are depicted in several sample images. Notice that even under the ideal case when the original binary image is used as the external bias field, the image restoration is not perfect. This is likey due to an averaging out effect arising from the interactions, casuing the high frequency components in the image to be lost.

3.2 Grayscale Image Restoration

Figure 5 shows the the quality of grayscale image restoration obtained via the three-dimensional and quasi-two-dimensional Ising models for different values of the inverse lattice temperature. By eliminating the inter-bit-plane spin coupling, we found that the quasi-two-dimensional significantly improves the overall quality of the image restoration process. However, neither obtain near perfect image reconstruction when interactions are taken into consideration, even when the original image is used as the external bias field. This problem arises primarily from the significant degradation of the lower bit-planes in the image due the noise. Only when there are no spin interactions on the lattice due we obtain near perfect image reconstruction under these ideal conditions. Of course, however, this mode of restoration is of little practical use.

Figure 6 shows the quality of the grayscale image restoration obtained via the three-dimensional and quasi-two-dimensional Ising models at an inverse temperature of $\beta = 1$ for varying degrees of noise degraded images when both an external bias field and spin-spin interactions are used in the restoration process. Here, in both cases, the limiting ceiling placed on the quality of image restoration due to the degradation in the lower bit-planes is quite obvious. Even in these limits, we find the quality of grayscale image restoration is rather poor compared to the results for binary image

Figure 4: *Binary image restoration of Lena via the two-dimensional Ising model. (Upper Left) Original binary image of Lena. (Upper Right) Degraded binary image of Lena ($\sigma^2 = 1$). (Lower Left) Restored image of Lena using the original binary image of Lena as the external bias field. (Lower Right) Restored image of Lena using the degraded image of Lena as the external bias field.*

restoration above. In Figure 7 and 8, the qualitative features of the grayscale image restoration process for the three-dimensional and quasi-two-dimensional Ising models are depicted in several sample images.

4 Conclusion

Despite the limited success of the preliminary results presented here, we argue that the generalized framework of the binary Ising model as it relates to the image restoration process is both quite unique and merits further study. All in all, We have really only considered a relatively basic, straightforward approch to using the Ising model. It is likely that this process can be significantly improved by considering not only simple techniques such as preconditioning the degraded image with smoothing filters, but also through more advanced, complex methods that exploit non-uniform spin-field and spin-spin coupling schemes. In addition, it is quite possible that many of the issues observed in the simple approches we have considered are fixed by using higher-order nearest-neighbor interactions. Whatever the case may be, these advanced coupling schemes and higher-order nearest-neighbor interactions would undoubtably carry over to higher dimensional binary Ising models that may be useful for color image restoration. Of course, however, all such speculations require further numerical and theoretical study.

Acknowledgements

Thanks Dr. Morris! It was great semester. I hope you have a good holiday.

References

CARLUCCI, D. M., AND INOUE, J.-I. 1999. Image restoration using the chiral Potts spin glass. *Phys. Rev. E 60*, 3, 2547–2553.

CHANDLER, D. 1987. *Introduction to Modern Statistical Mechanics*. Oxford University Press, Inc.

HUANG, K. 1987. *Statistical Mechanics*, Second ed. John Wiley and Sons, Inc.

INOUE, J.-I., AND CARLUCCI, D. M. 2001. Image restoration using the Q-Ising spin glass. *Phys. Rev. E 64*, 036121.

INOUE, J.-I. 2001. Application of the quantum spin glass theory to image restoration. *Phys. Rev. E 63*, 046114.

ISING, E. 1925. Beitrag zur Theorie des Ferro-und Paramgnetismus. *Z. Phys. 31*, 253.

METROPOLIS, N., ROSENBLUTH, A. W., TELLER, A. H., AND TELLER, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.*.

PATHRIA, R. K. 1996. *Statistical Mechanics*, Second ed. Elsevier Butterworth-Heinemann.

TANAKA, K., INOUE, J.-I., AND TITTERINGTON, D. M. 2003. Probabilistic image processing by means for the Bethe approximation for the Q-Ising model. *J. Phys. A 36*, 11023–11035.

TANAKA, K. 2002. Statistical-mechanical approach to image processing. *J. Phys. A 35*, R81–R150.

Appendix A: MATLAB M-Files

```
% ==========================================================================
% Math 336 - Image Processing - Final Project
%
% This MATLAB script is used for generating the noisy binary image data in
% a format suitable to be read-in by the FORTRAN program ising2.x.
%
% Marty Kandes
% Computational Science Research Center / Department of Physics
% San Diego State University
% Fall 2008
% --------------------------------------------------------------------------
clear all; close all; clc;              % Clear MATLAB workspace.
f = imread('lena.jpg');                 % Read image from graphics file.
f = imresize(f,[256,256]);              % Resize image.
f = im2double(f);                       % Convert image to double.
f = mat2gray(f);                        % Renormalize image's grayscale.
mean = 0;                               % Set noise mean.
variance = 0.001;                       % Set noise variance.
n = imnoise(f,'gaussian',mean,variance); % Add noise to image.
b = im2bw(n,graythresh(n));             % Convert image to binary.
imwrite(b,'lenagn0001.png','png');      % Write image to graphics file.
[M N] = size(b);                        % Get dimensions of binary image.
fid = fopen('lenagn0001.dat','w');      % Open file to write out data.
for J = 1:N                             %
    for I = 1:M                         %
        fprintf(fid,'%1i \n',b(I,J));   % Write binary image data to file.
    end;                                %
end;                                    %
status = fclose(fid);                   % Close file.
% ==========================================================================
% Math 336 - Image Processing - Final Project
%
% This MATLAB script is used for reading in the binary image data from the
% output of the FORTRAN program ising2.x to construct the restored binary
% image.
%
% Marty Kandes
% Computational Science Research Center / Department of Physics
% San Diego State University
% Fall 2008
% --------------------------------------------------------------------------
clear all; close all; clc;           % Clear MATLAB workspace.
M = 256; N = 256;                    % Set image dimensions.
load lenarestored.dat;               % Load binary image data.
K = 0;                               % Initialize loop counter.
for J = 1:N                          %
    for I = 1:M                      %
        K = K + 1;                   %
        b(I,J) = lenarestored(K,1);  % Read binary image data into array b.
    end;                             %
end;                                 %
imwrite(b,'lenarestored.png','png'); % Write image to graphics file.
% ==========================================================================
% Math 336 - Image Processing - Final Project
%
% This MATLAB script is used for generating binary bit plane data of a
% noisy grayscale image in a format suitable to be read-in by the FORTRAN
% program ising3.x.
%
% Marty Kandes
% Computational Science Research Center / Department of Physics
```

```
% San Diego State University
% Fall 2008
% -----------------------------------------------------------------------
clear all; close all; clc;                 % Clear MATLAB workspace.
f = imread('lena.jpg');                    % Read image from graphics file.
f = imresize(f,[256,256]);                 % Resize image.
mean = 0;                                  % Set noise mean.
variance = 0.001;                          % Set noise variance.
n = imnoise(f,'gaussian',mean,variance);   % Add noise to image.
imwrite(n,'lenagn0001.png','png');         % Write image to graphics file.
n = double(n);                             % Convert image to double.
[M,N] = size(n); P = 8;                    % Set num bit planes and dims.
b = zeros(M,N,P);                          % Preallocate bit plane array.
b(:,:,1) = logical(mod(n,2));              % Compute bit planes. Level 1. LSB.
b(:,:,2) = logical(mod(floor(n/2),2));     % Level 2.
b(:,:,3) = logical(mod(floor(n/4),2));     % Level 3.
b(:,:,4) = logical(mod(floor(n/8),2));     % Level 4.
b(:,:,5) = logical(mod(floor(n/16),2));    % Level 5.
b(:,:,6) = logical(mod(floor(n/32),2));    % Level 6.
b(:,:,7) = logical(mod(floor(n/64),2));    % Level 7.
b(:,:,8) = logical(mod(floor(n/128),2));   % Level 8. MSB.
fid = fopen('lenagn0001.dat','w');         % Open file to write out data.
for K = 1:P                                %
    for J = 1:N                            %
        for I = 1:M                        %
            fprintf(fid,'%1i \n',b(I,J,K)); % Write bit plane data to file.
        end;                               % LSBP to MSBP.
    end;                                   %
end;                                       %
status = fclose(fid);                      % Close file.
% =======================================================================
% Math 336 - Image Processing - Final Project
%
% This MATLAB script is used for reading in the binary bit plane data of a
% output by the FORTRAN program ising3.x to construct the restored image.
%
% Marty Kandes
% Computational Science Research Center / Department of Physics
% San Diego State University
% Fall 2008
% -----------------------------------------------------------------------
clear all; close all; clc;             % Clear MATLAB workspace.
M = 256; N = 256; P = 8;               % Set num bit planes and dims.
b = zeros(M,N,P);                      % Preallocate bit plane array.
load lenarestored.dat;                 % Load bit planes.
L = 0;                                 % Initialize loop counter.
for K = 1:P                            %
    for J = 1:N                        %
        for I = 1:M                    %
            L = L + 1;                 %
            b(I,J,K) = lenarestored(L,1); % Read bit plane data into array.
        end;                           %
    end;                               %
end;                                   % Compute restored image from bit planes.
f = uint8(2*(2*(2*(2*(2*(2*(2*b(:,:,8)+b(:,:,7))+b(:,:,6))+b(:,:,5))+b(:,:,4))+b(:,:,3))+b(:,:,2))+b(:,:,1));
imwrite(f,'lenarestored.png','png'); % Write image to graphics file.
% =======================================================================
```

Appendix B: FORTRAN Source Code

```
! =======================================================================
! Math 336 - Image Processing - Final Project
!
! Binary Image Restoration using the 2D Ising Model
!
! Marty Kandes
! Computational Science Research Center / Department of Physics
! San Diego State University
! Fall 2008
! -----------------------------------------------------------------------
      PROGRAM ISING2
      IMPLICIT NONE

      INTEGER, PARAMETER :: NITS = 6553600  ! Num Monte Carlo iterations.
      INTEGER, PARAMETER :: NINV = 1000     ! Num inverse temps.
      INTEGER, PARAMETER :: NPIX = 256      ! Num pixels in X.
      INTEGER, PARAMETER :: NPIY = 256      ! Num pixels in Y.

      INTEGER :: FILENUM                    ! Input / output file number(s).
      INTEGER :: I, J                       ! Loop counters.
      INTEGER :: U, V                       ! Integer spin coordinates.
      INTEGER :: DELTAE                     ! Energy diff in spin configs.

      INTEGER, ALLOCATABLE :: B(:,:)        ! Original binary image.
      INTEGER, ALLOCATABLE :: H(:,:)        ! External magnetic field.
      INTEGER, ALLOCATABLE :: S(:,:)        ! Spin lattice / restore image.

      REAL, PARAMETER :: DELTAB = 0.01E0    ! Diff between inverse temps.

      REAL :: X, Y                          ! Real spin coordinates.
      REAL :: BETA                          ! Inverse temperature.
      REAL :: ETA                           ! Random number between 0 & 1.

      FILENUM = 998                         ! Initialize file number(s).

      ALLOCATE(B(NPIX,NPIY))                ! Allocate arrays. Orig image.
      ALLOCATE(H(NPIX,NPIY))                ! External field.

      DO J = 1, NPIY                        ! Read in external field.
         DO I = 1, NPIX                     !
            READ(FILENUM,*) H(I,J)          !
            IF (H(I,J) == 0) THEN           ! Convert 0 bits to -1.
               H(I,J) = -1                  ! Need -1 bits for calculations.
            ENDIF                           !
         ENDDO                              !
      ENDDO                                 !
      FILENUM = FILENUM + 1                 ! Increment file number.

      DO J = 1, NPIY                        ! Read in original, noisy
         DO I = 1, NPIX                     ! binary image to be restored.
            READ(FILENUM,*) B(I,J)          ! Convert 0 bits to -1.
            IF (B(I,J) == 0) THEN           !
               B(I,J) = -1                  !
            ENDIF                           !
         ENDDO                              !
      ENDDO                                 !
      FILENUM = FILENUM + 1                 ! Increment file number.

      !H = 0                                 ! Uncomment for no ext field.

      DO J = 1, NINV                        ! Loop over inv temps.
         BETA = FLOAT(J)*DELTAB             ! Compute current inv temp.
         CALL RANDOM_SEED()                 ! Seed random number generator.
         S = B                              ! Init spin lattice to orig im.
         DO I = 1, NITS                     ! Loop over num MC iterations.
            CALL RANDOM_NUMBER(X)           ! Random X spin coordinate.
            CALL RANDOM_NUMBER(Y)           ! Random Y spin coordinate.
            CALL RANDOM_NUMBER(ETA)         ! Random number between 0 & 1.
            U = CEILING(FLOAT(NPIX)*X)      ! Integer X spin coordinate.
            V = CEILING(FLOAT(NPIY)*Y)      ! Integer Y spin coordinate.
            DELTAE = H(U,V)                 ! Compute difference in the
            IF (U.NE.1) THEN                ! configuration energy of the
               DELTAE = DELTAE + S(U-1,V)   ! spin lattice if we were to
            ENDIF                           ! flip the current spin under
            IF (U.NE.NPIX) THEN             ! consideration. Here, we use
               DELTAE = DELTAE + S(U+1,V)   ! non-periodic boundary
            ENDIF                           ! conditions.
            IF (V.NE.1) THEN                !
               DELTAE = DELTAE + S(U,V-1)   ! Comment out these if
            ENDIF                           ! statements to consider Ising
            IF (V.NE.NPIY) THEN             ! model without local
               DELTAE = DELTAE + S(U,V+1)   ! interactions.
            ENDIF                           !
            DELTAE = 2*S(U,V)*DELTAE        ! Final energy difference.
            IF (DELTAE.LE.0) THEN           ! Use Metropolis algorithm
               S(U,V) = -S(U,V)             ! to decide if we flip the
            ELSEIF ((-BETA*FLOAT(DELTAE)).LT.LOG(ETA)) THEN ! current spin
               S(U,V) = S(U,V)              ! under consideration or keep
            ELSE                            ! it's present state.
               S(U,V) = -S(U,V)             !
            ENDIF                           !
         ENDDO                              !
         DO V = 1, NPIY                     !
            DO U = 1, NPIX                  !
               IF (S(U,V) == -1) THEN       ! Convert -1 bits to 0 bits.
                  S(U,V) = 0                ! Need 0 bits for image file.
               ENDIF                        !
               WRITE(FILENUM,*) S(U,V)      ! Write restored image to file.
            ENDDO                           !
         ENDDO                              !
         FILENUM = FILENUM + 1              ! Increment file number.
      ENDDO                                 !

      DEALLOCATE(B,H,S)                     ! Deallocate arrays.

      STOP
      END
! =======================================================================
! Math 336 - Image Processing - Final Project
!
! Grayscale Image Restoration using the 3D Ising Model
!
! Marty Kandes
! Computational Science Research Center / Department of Physics
! San Diego State University
! Fall 2008
! -----------------------------------------------------------------------
      PROGRAM ISING3
      IMPLICIT NONE

      INTEGER, PARAMETER :: NITS = 52428800 ! Num Monte Carlo iterations.
      INTEGER, PARAMETER :: NINV = 1000     ! Num inverse temps.
      INTEGER, PARAMETER :: NPIX = 256      ! Num pixels in X.
      INTEGER, PARAMETER :: NPIY = 256      ! Num pixels in Y.
      INTEGER, PARAMETER :: NBPZ = 8        ! Num bit planes.

      INTEGER :: FILENUM                    ! Input / output file number(s).
      INTEGER :: I, J, K                    ! Loop counters.
      INTEGER :: U, V, W                    ! Integer spin coordinates.
      INTEGER :: DELTAE                     ! Energy diff in spin configs.

      INTEGER, ALLOCATABLE :: B(:,:,:)      ! Original, noisy binary image.
      INTEGER, ALLOCATABLE :: H(:,:,:)      ! External magnetic field.
      INTEGER, ALLOCATABLE :: S(:,:,:)      ! Spin lattice / restore image.

      REAL, PARAMETER :: DELTAB = 0.01E0    ! Diff between inverse temps.

      REAL :: X, Y, Z                       ! Real spin coordinates.
      REAL :: BETA                          ! Inverse temperature.
      REAL :: ETA                           ! Random number between 0 & 1.

      FILENUM = 998                         ! Initialize file number(s).

      ALLOCATE(B(NPIX,NPIY,NBPZ))           ! Allocate arrays. Orig image.
      ALLOCATE(H(NPIX,NPIY,NBPZ))           ! External field.
      ALLOCATE(S(NPIX,NPIY,NBPZ))           ! Spin lattice / rest image.

      DO K = 1, NBPZ                        !
         DO J = 1, NPIY                     !
            DO I = 1, NPIX                  !
               READ(FILENUM,*) H(I,J,K)     ! Read in external field.
               IF (H(I,J,K) == 0) THEN      ! Convert 0 bits to -1.
                  H(I,J,K) = -1             ! Need -1 bits for cals.
               ENDIF                        !
            ENDDO                           !
         ENDDO                              !
      ENDDO                                 !
      FILENUM = FILENUM + 1                 ! Increment file number.
```

```
      DO K = 1, NBPZ                         !
         DO J = 1, NPIY                      ! Read in original, noisy
            DO I = 1, NPIX                   ! binary image to restore.
               READ(FILENUM,*) B(I,J,K)      ! Convert 0 bits to -1.
               IF (B(I,J,K) == 0) THEN       !
                  B(I,J,K) = -1              !
               ENDIF                         !
            ENDDO                            !
         ENDDO                               !
      ENDDO                                  !
      FILENUM = FILENUM + 1                  ! Increment file number.

      !H = 0                                  ! Uncomment for no ext field.

      DO J = 1, NINV                         ! Loop over inv temps.
         BETA = FLOAT(J)*DELTAB              ! Compute current inv temp.
         CALL RANDOM_SEED()                  ! Seed random number generator.
         S = B                               ! Init spin lattice to orig im.
         DO I = 1, NITS                      ! Loop over num MC iterations.
            CALL RANDOM_NUMBER(X)            ! Random X spin coordinate.
            CALL RANDOM_NUMBER(Y)            ! Random Y spin coordinate.
            CALL RANDOM_NUMBER(Z)            ! Random Z spin coordinate.
            CALL RANDOM_NUMBER(ETA)          ! Random number between 0 & 1.
            U = CEILING(FLOAT(NPIX)*X)       ! Integer X spin coordinate.
            V = CEILING(FLOAT(NPIY)*Y)       ! Integer Y spin coordinate.
            W = CEILING(FLOAT(NBPZ)*Z)       ! Integer Z spin coordinate.
            DELTAE = H(U,V,W)                ! Compute difference in the
            IF (U.NE.1) THEN                 ! configuration energy of the
               DELTAE = DELTAE + S(U-1,V,W) ! spin lattice if we were to
            ENDIF                            ! flip the current spin under
            IF (U.NE.NPIX) THEN              ! consideration. Here, we use
               DELTAE = DELTAE + S(U+1,V,W) ! non-periodic boundary
            ENDIF                            ! conditions.
            IF (V.NE.1) THEN                 !
               DELTAE = DELTAE + S(U,V-1,W) ! Comment out these if
            ENDIF                            ! statements to consider Ising
            IF (V.NE.NPIY) THEN              ! model without interactions.
               DELTAE = DELTAE + S(U,V+1,W) !
            ENDIF                            !
            IF (W.NE.1) THEN                 !
               DELTAE = DELTAE + S(U,V,W-1) !
            ENDIF                            !
            IF (W.NE.NBPZ) THEN              !
               DELTAE = DELTAE + S(U,V,W+1) !
            ENDIF                            !
            DELTAE = 2*S(U,V,W)*DELTAE       ! Final energy difference.
            IF (DELTAE.LE.0) THEN            ! Use Metropolis algorithm
               S(U,V,W) = -S(U,V,W)          ! to decide if we flip the
            ELSEIF ((-BETA*FLOAT(DELTAE)).LT.LOG(ETA)) THEN ! current spin
               S(U,V,W) = S(U,V,W)           ! under consideration or keep
            ELSE                             ! it's present state.
               S(U,V,W) = -S(U,V,W)          !
            ENDIF                            !
         ENDDO                               !
         DO W = 1, NBPZ                      !
            DO V = 1, NPIY                   !
               DO U = 1, NPIX                !
                  IF (S(U,V,W) == -1) THEN  ! Convert -1 bits to 0 bits.
                     S(U,V,W) = 0            ! Need 0 bits for image file.
                  ENDIF                      !
                  WRITE(FILENUM,*) S(U,V,W) ! Write restored image out.
               ENDDO                         !
            ENDDO                            !
         ENDDO                               !
         FILENUM = FILENUM + 1               ! Increment file number.
      ENDDO                                  !

      DEALLOCATE(B,H,S)                      ! Deallocate arrays.

      STOP
      END
! ======================================================================
! Math 336 - Image Processing - Final Project
!
! Bitwise image comparison between original, uncorrupted image and the
! restored Ising images. Computes either the binary or grayscale
! degradation-restoration coefficient. i.e. NBPZ = 1 for C_{BDR} and
! NBPZ = 8 for C_{GDR}.
!
! Marty Kandes
! Computational Science Research Center / Department of Physics
! San Diego State University
! Fall 2008
! ----------------------------------------------------------------------
      PROGRAM BITCHECK
      IMPLICIT NONE

      INTEGER, PARAMETER :: NSEQ = 1000     ! Num bit sequences / images.
      INTEGER, PARAMETER :: NPIX = 256      ! Num pixels in X.
      INTEGER, PARAMETER :: NPIY = 256      ! Num pixels in Y.
      INTEGER, PARAMETER :: NBPZ = 1        ! Num bit planes.

      INTEGER :: I, J, K, L                  ! Loop count.
      INTEGER :: FILENUM                     ! Input file number(s).
      INTEGER :: BITCOUNT                    ! Similar bit count.

      INTEGER, ALLOCATABLE :: IM(:,:,:)     ! Original image bit sequence.
      INTEGER, ALLOCATABLE :: RE(:,:,:)     ! Restored image bit sequence.

      REAL, PARAMETER :: DELTAB = 0.01E0    ! Diff between inv temps.

      REAL :: BETA                           ! Inverse temperature.

      FILENUM = 999                          ! Initialize file numbers.

      ALLOCATE(IM(NPIX,NPIY,NBPZ))           ! Allocate arrays. Orig image.
      ALLOCATE(RE(NPIX,NPIY,NBPZ))           ! Restored image(s).

      DO K = 1, NBPZ                         !
         DO J = 1, NPIY                      !
            DO I = 1, NPIX                   !
               READ(FILENUM,*) IM(I,J,K)    ! Read in original image data.
            ENDDO                            !
         ENDDO                               !
      ENDDO                                  !
      FILENUM = FILENUM + 1                  ! Increment file number.

      DO L = 1, NSEQ                         !
         BITCOUNT = 0                        ! Initialize bit count.
         BETA = FLOAT(L)*DELTAB              ! Compute current inv temp.
         DO K = 1, NBPZ                      !
            DO J = 1, NPIY                   !
               DO I = 1, NPIX                !
                  READ(FILENUM,*) RE(I,J,K) ! Read restored image(s).
               ENDDO                         !
            ENDDO                            !
         ENDDO                               !
         DO K = 1, NBPZ                      !
            DO J = 1, NPIY                   !
               DO I = 1, NPIX                !
                  IF (RE(I,J,K) == IM(I,J,K)) THEN
                     BITCOUNT = BITCOUNT + 1 ! Num bits same.
                  ENDIF                      !
               ENDDO                         !
            ENDDO                            !
         ENDDO                               ! Frac of bits same.
         WRITE(6,*) BETA, FLOAT(BITCOUNT)/FLOAT(NPIX*NPIY*NBPZ)
         FILENUM = FILENUM + 1               ! Increment file number.
      ENDDO

      DEALLOCATE(IM,RE)

      STOP
      END
! ======================================================================
! Math 336 - Image Processing - Final Project
!
! Applys a median binary filter to either binary or grayscale image date
! to build an approximate external field, h_{ij} or h_{ijk}, for use in the
! Ising image restoration process. Note that for the case of grayscale
! image data, filtering is not applied between different bit planes.
!
! Marty Kandes
! Computational Science Research Center / Department of Physics
! San Diego State University
! Fall 2008
! ----------------------------------------------------------------------
      PROGRAM BINMEDIAN
      IMPLICIT NONE

      INTEGER, PARAMETER :: NPIX = 256      ! Num pixels in X.
      INTEGER, PARAMETER :: NPIY = 256      ! Num pixels in Y.
      INTEGER, PARAMETER :: NBPZ = 8        ! Num bit planes.

      INTEGER :: I, J, K, L                  ! Loop count.
      INTEGER :: MEDIAN                      ! Median count.

      INTEGER, ALLOCATABLE :: NI(:,:,:)     ! Noisy image.
      INTEGER, ALLOCATABLE :: MF(:,:,:)     ! Median filtered image.

      ALLOCATE(NI(NPIX,NPIY,NBPZ))           ! Allocate arrays. Orig image.
      ALLOCATE(MF(NPIX,NPIY,NBPZ))           ! Median filtered image.

      DO K = 1, NBPZ                         !
         DO J = 1, NPIY                      !
            DO I = 1, NPIX                   !
               READ(5,*) NI(I,J,K)           ! Read in original image data.
            ENDDO                            !
         ENDDO                               !
      ENDDO                                  !

      DO K = 1, NBPZ                         !
         DO I = 1, NPIX                      !
            MF(I,1,K) = NI(I,1,K)            !
         ENDDO                               !
         DO J = 2, NPIY-1                    !
            MF(1,J,K) = NI(1,J,K)            !
            DO I = 2, NPIX-1                 ! Apply median filter.
               MEDIAN = NI(I,J,K)+NI(I-1,J,K)+NI(I+1,J,K)+NI(I,J-1,K)+NI(I,J+1,K)
               IF (MEDIAN.GE.3) THEN         !
                  MF(I,J,K) = 1              !
               ELSE                          !
                  MF(I,J,K) = 0              !
               ENDIF                         !
            ENDDO                            !
            MF(NPIX,J,K) = NI(NPIX,J,K)      !
         ENDDO                               !
         DO I = 1, NPIX                      !
            MF(I,NPIY,K) = NI(I,NPIY,K)      !
         ENDDO                               !
      ENDDO

      DO K = 1, NBPZ                         !
         DO J = 1, NPIY                      !
            DO I = 1, NPIX                   !
               WRITE(6,*) MF(I,J,K)          ! Write filtered image data.
            ENDDO                            !
         ENDDO                               !
      ENDDO

      DEALLOCATE(NI,MF)                      ! Deallocate arrays.

      STOP
      END
! ======================================================================
```

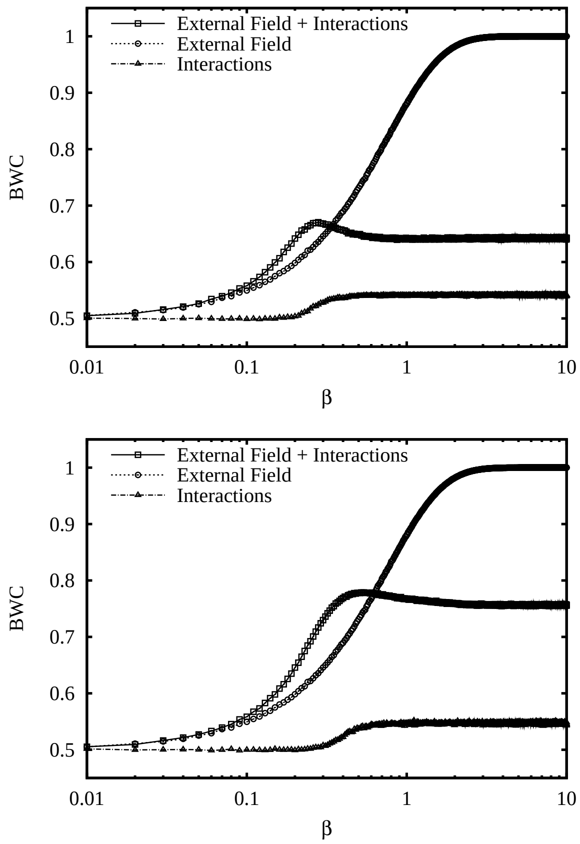

Figure 5: *(Top) Grayscale image restoration of the degraded grayscale image of Lena ($\sigma^2 = 1$) versus the inverse lattice temperature using the three-dimensional Ising model. In this case, original grayscale image of Lena was used as the external bias field. (Bottom) Grayscale image restoration of the degraded grayscale image of Lena ($\sigma^2 = 1$) versus inverse lattice temperature using the quasi-two-dimensional Ising model. Again, the original grayscale image of Lena was used here as the external bias field.*

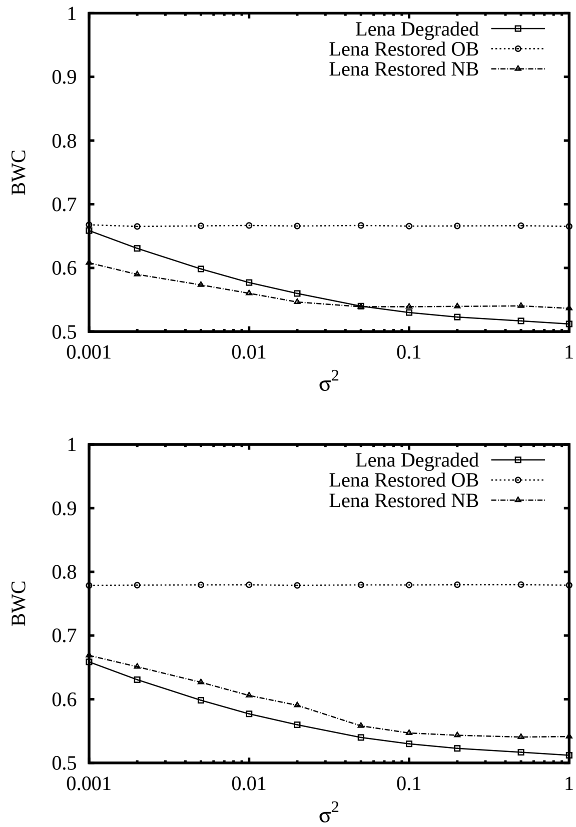

Figure 6: *(Top) Grayscale image restoration of the degraded grayscale images of Lena at an inverse temperature of $\beta = 1$ versus the magnitude of the gaussian noise variance, σ^2, using the three-dimensional Ising model. (Bottom) Grayscale image restoration of the degraded grayscale images of Lena at an inverse temperature of $\beta = 1$ versus the magnitude of the gaussian noise variance using the quasi-two-dimensional Ising model. In both cases, the solid lines and open squares are the measured quality of the degraded images of Lena with respect to the original image. The dotted lines and open circles are the measured quality of the restored images generated by using the original grayscale image as the external bias field. The dashed-dotted lines and the open triangles are the measured quality of the restored images generated by using the degraded images as the external bias field. Note that both the external bias fiel and spin-spin interactions were used in the restoration process.*

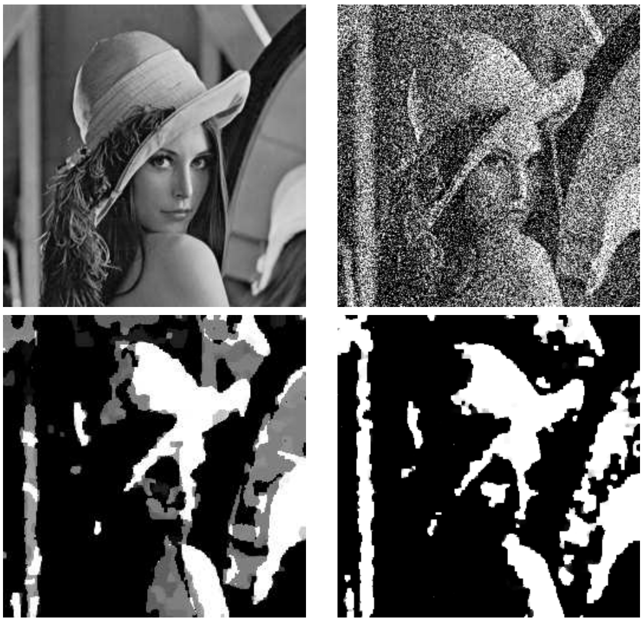

Figure 7: *Grayscale image restoration of Lena via the three-dimensional Ising model. (Upper Left) Original grayscale image of Lena. (Upper Right) Degraded grayscale image of Lena ($\sigma^2 = 1$). (Lower Left) Restored image of Lena using the original grayscale image of Lena as the external bias field. (Lower Right) Restored image of Lena using the degraded image of Lena as the external bias field.*

Figure 8: *Grayscale image restoration of Lena via the quasi-two-dimensional Ising model. (Upper Left) Original grayscale image of Lena. (Upper Right) Degraded grayscale image of Lena ($\sigma = 1$). (Lower Left) Restored image of Lena using the original grayscale image of Lena as the external bias field. (Lower Right) Restored image of Lena using the degraded image of Lena as the external bias field.*