Digital Image Processing Laboratory: MAP Image Restoration using JAX

September 18, 2024

1 Introduction

This laboratory explores the use of maximum *a posteriori* (MAP) estimation of images from noisy and blurred data using both Gaussian and non-Gaussian Markov random field (MRF) prior models. This estimation methodology is widely useful for problems of image restoration and reconstruction because it can be used to explicitly model both measurement noise and prior knowledge regarding the image to be estimated.

In order to use a MAP estimation framework, we will need three major components:

- *Prior model* - This is the probability distribution that we assume for the unknown image to be estimated. We will use and MRF prior model because it captures essential characteristics of images while remaining computationally tractable.
- *Measurement noise model* - We introduce a general model for the measurement which consists of a linear blurring operator followed by additive Gaussian noise.
- *Optimization method* - The MAP estimation framework results in a high dimensional optimization problem that must be solved. There are many ways to solve this problem, but we introduce a useful optimization technique known as iterative coordinate descent (ICD).

All solutions for this laboratory should be implemented using the JAX package in Python. It is important that students learn to program in JAX because it is efficient and highly portable, but it requires some experience to construct a high quality program that can be easily debugged.

1.1 Preparation for JAX

In this lab, you are required to use JAX. To help you get started, the GitHub repository `jax_startup` offers useful resources for setup. You can retrieve the `jax_startup` repository by running the following command:

```
git clone git@github.com:cabouman/jax_startup
```

Questions or comments concerning this laboratory should be directed to Prof. Charles A. Bouman, School of Electrical and Computer Engineering, Purdue University, West Lafayette IN 47907; (765) 494-0340; bouman@ecn.purdue.edu

Your first task is to install the necessary packages for this lab. You can follow the instructions in the README of the `jax_startup` repository to set up the conda environment and install the required packages.

It is also strongly recommended that you run and study the FIR demo provided in the repository. You can run the demo by following the instructions in the README. In the demo, you will learn how to use the `jax.vmap` function, which is essential for completing this lab.

2 MAP Estimation with Gaussian Priors

This first section of the lab introduces Markov random field (MRF) image models along with the basic techniques required for MAP estimation with a Gaussian prior. In this case, the optimization is quadratic.

2.1 Prior Model Formulation

For the purposes of this laboratory, the random field $\{X_s\}_{s\in S}$ denotes a image with continuously valued pixels taking values on the lattice S with N points. Furthermore, X is assumed to be an MRF with a strictly positive density; so by the Hammersly-Clifford theorem, its density must have the form of a Gibbs distribution. We also make the common assumption that the Gibbs distribution for X uses only pairwise interactions. In this case, the set of all cliques is given by

$$\mathcal{C} = \{\{i,j\} | \, i \in \partial j \text{ for } i,j \in S\} \ ,$$

where ∂j denotes the neighbors of j. For this laboratory, we use an 8-point neighborhood system.

In many practical applications, the Gibbs density of the MRF is assume to have the following form

$$p(x) = \frac{1}{z} \exp\left\{ - \sum_{\{i,j\}\in\mathcal{C}} b_{i,j}\rho(x_i - x_j) \right\} \ , \tag{1}$$

where $\rho(\cdot)$ is a real valued, positive and symmetric function and z is the normalizing constant known as the partition function. There are a wide variety of common choices for this function, but for this portion of the laboratory we focus on a class of MRF's known as generalized Gaussian MRF's with the form

$$p(x) = \frac{1}{z(g,p,\sigma)} \exp\left\{ -\frac{1}{p\sigma^p} \sum_{\{i,j\}\in\mathcal{C}} g_{i,j} \left| x_i - x_j \right|^p \right\} \ , \tag{2}$$

$$\tag{3}$$

where $z(g,p,\sigma)$ is the normalizing constant for the distribution, and we assume that the function $g_{i,j}$ is normalized so that

$$1 = \sum_{j\in\partial i} g_{i,j} \ . \tag{4}$$

It can be shown that for this distribution $z(g,p,\sigma) = z(g,p,1)\sigma^N$ where N is the number of lattice points. Using this result, we see that

$$p(x) = \frac{1}{z(g,p,1)\sigma^N} \exp\left\{ -\frac{1}{p\sigma^p} \sum_{\{i,j\}\in\mathcal{C}} g_{i,j} \left| x_i - x_j \right|^p \right\} \ . \tag{5}$$

For simplicity, we assume that S is a rectangular lattice, that the function g is shift invariant so that $g_{i,j} = g_{i-j} = g_{j-i}$, and we use a circular boundary condition. With these assumptions, (4) holds whenever

$$1 = \sum_{j\in\partial 0} g_{0-j} \ .$$

In this case, it can be shown that

$$\begin{aligned} p(x_i|x_j \, j \neq i) &= p(x_i|x_j, j \in \partial i) \qquad (6) \\ &= \frac{1}{z} \exp\left\{ -\frac{1}{p\sigma^p} \sum_{j\in\partial i} g_{i-j} \left| x_i - x_j \right|^p \right\} \ . \qquad (7) \end{aligned}$$

For the special case of $p = 2$, the GGMRF is Gaussian. In this case, the Gibbs distribution of (5) can be written in Matrix form as

$$p(x) = \frac{1}{(2\pi\sigma)^{(N/2)}} |B|^{1/2} \exp\left\{ -\frac{1}{2\sigma^2} x^t B x \right\} \tag{8}$$

where x is a vectorized form of the image and B is a symmetric positive definite matrix with entries

$$B_{i,j} = \delta_{i-j} - g_{i,j} \ .$$

For a GMRF, the conditional density of a pixel is given by

$$p(x_i|x_j \, j \neq i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2} \left(x_i - \sum_{j\in\partial i} g_{i-j} x_j \right)^2 \right\} \ . \tag{9}$$

So we can see that the conditional expectation and variance are given by

$$\begin{aligned} E\left[x_i|x_j, j \neq i\right] &= \sum_{j\in\partial i} g_{i-j} x_j \qquad (10) \\ Var\left[x_i|x_j, j \neq i\right] &= \sigma^2 \ . \qquad (11) \end{aligned}$$

Using the expression of (5), it can be shown that the maximum likelihood estimator for the scale parameter σ of the GGMRF is given by

$$\hat{\sigma}^p = \frac{1}{N} \sum_{\{i,j\}\in\mathcal{C}} g_{i,j} \left| x_i - x_j \right|^p \ . \tag{12}$$

For the following section problems, the image img04.tif will be modeled using a GGMRF with a circular boundary conditions with parameters g_i to be given by the following 2-D array of values

1/12	1/6	1/12
1/6	0	1/6
1/12	1/6	1/12

(13)

where the $g_0 = 0$ term is at the center of the array. Furthermore, the pixel values of the image fall in the range $[0, 255]$ and should **not** be rescaled.

Section Problems:

1. Show that for the distribution of (2) the normalizing constant has the form $z(g, p, \sigma) = z(g, p, 1)\sigma^N$ where N is the number of lattice points (i.e. pixels).

2. Use the result of equation (5) to derive the ML estimate of σ^p shown in (12).

3. Compute the noncausal prediction error for the image img04.tif

$$e_i = x_i - \sum_{j \in \partial i} g_{i-j} x_j$$

   and display it as an image by adding an offset of 127 to each pixel. Clip any value which is less than 0 or greater than 255 after adding the offset of 127.

4. Compute $\hat{\sigma}_{ML}$ the ML estimate of the scale parameter σ for values of p in the range $0.1 \leq p \leq 2$. Do not include cliques that fall across boundaries of the image. Plot $\hat{\sigma}_{ML}$ (not $\hat{\sigma}^p_{ML}$) versus p for p ranging from 0.1 to 2.

2.2 MAP Restoration with Additive Noise and Gaussian Prior

In this section, you will use a Gaussian MRF to compute the MAP estimate of an image that has been corrupted using additive Gaussian noise. Assume that the noisy image Y is related to the noiseless image X by

$$Y = X + W \ ,$$

where W is i.i.d. Gaussian noise with mean 0 and variance $\sigma_W^2 = 4^2$. Furthermore, assume that X is a Gaussian MRF with noncausal prediction variance of σ_x^2 and a noncausal prediction filter of g_i shown in (13).

The MAP estimate is given by

$$\begin{aligned}\hat{x} &= \arg\max_x \left\{ p_{x|y}(x|y) \right\} \\ &= \arg\max_x \left\{ \log p(y|x) + \log p(x) \right\} \ ,\end{aligned}$$

where

$$p(y|x) = \frac{1}{(2\pi\sigma_W^2)^{N/2}} \exp\left\{ -\frac{1}{2\sigma_W^2} \sum_{i \in S} (y_i - x_i)^2 \right\} \ ,$$

and

$$p(x) = \frac{1}{z\sigma_x^N} \exp\left\{ -\frac{1}{2\sigma_x^2} \sum_{\{i,j\} \in \mathcal{C}} g_{i,j}(x_i - x_j)^2 \right\} \ .$$

So the MAP estimate of x is given by

$$\hat{x} = \arg\min_x \left\{ \frac{1}{2\sigma_W^2} \sum_{i \in S} (y_i - x_i)^2 + \frac{1}{2\sigma_x^2} \sum_{\{i,j\} \in \mathcal{C}} g_{i,j}(x_i - x_j)^2 \right\} \ , \tag{14}$$

and the cost function being minimized is given by

$$c(x) = \frac{1}{2\sigma_W^2} \sum_{i \in S} (y_i - x_i)^2 + \frac{1}{2\sigma_x^2} \sum_{\{i,j\} \in \mathcal{C}} g_{i,j} (x_i - x_j)^2 \ . \tag{15}$$

In some cases, the measurement noise variance, σ_W^2, may also be unknown. In this case, σ_W^2 is sometimes referred to as a nuisance parameter because its value may not be of direct interest, but it is require in order to get an accurate estimate of x. One approach to the estimation of σ_W^2 is to jointly optimize the posterior distribution over x and σ_W^2.

$$\begin{aligned} (\hat{x}, \hat{\sigma}_W^2) &= \arg \max_{x, \sigma_W^2} \left\{ p_{x|y}(x|y, \sigma_W^2) \right\} \qquad (16) \\ &= \arg \max_{x, \sigma_W^2} \left\{ \log p(y|x, \sigma_W^2) + \log p(x) \right\} \end{aligned}$$

It can be shown that the optimization with respect to σ_W^2 results in

$$\hat{\sigma}_W^2 = \frac{1}{N} \sum_{i \in S} (y_i - x_i)^2 \ . \tag{17}$$

Substituting this into the expression for the joint optimization results in

$$\hat{x} = \arg \min_x \left\{ \frac{N}{2} \log \left(\sum_{i \in S} (y_i - x_i)^2 \right) + \frac{1}{2\sigma_x^2} \sum_{\{i,j\} \in \mathcal{C}} g_{i,j} (x_i - x_j)^2 \right\} \ , \tag{18}$$

which results in a new cost function given by

$$c(x) = \frac{N}{2} \log \left(\sum_{i \in S} (y_i - x_i)^2 \right) + \frac{1}{2\sigma_x^2} \sum_{\{i,j\} \in \mathcal{C}} g_{i,j} (x_i - x_j)^2 \ . \tag{19}$$

In practice, direct optimization of the cost function in equation (18) can be difficult, so it is often easier to iteratively minimize with respect to x and then σ_W^2. This can be accomplished by repeatedly applying the following pair of update equations.

$$\begin{aligned} \hat{\sigma}_W^2 &\leftarrow \frac{1}{N} \sum_{i \in S} (y_i - \hat{x}_i)^2 \qquad (20) \\ \hat{x} &\leftarrow \arg \min_x \left\{ \frac{1}{2\hat{\sigma}_W^2} \sum_{i \in S} (y_i - x_i)^2 + \frac{1}{2\sigma_x^2} \sum_{\{i,j\} \in \mathcal{C}} g_{i,j} (x_i - x_j)^2 \right\} \qquad (21) \end{aligned}$$

Section Problems:

1. Show that the cost function of (15) is strictly convex.

2. Derive equations (17) and (18) for the joint MAP estimation of x and σ_W^2.

3. Does the application of update (20) followed by (21) reduce the cost function of (19)? Why?

4. Let $(\hat{x}, \hat{\sigma}_W^2)$ denote the exact solution to the optimization of equation (16). Is $\hat{\sigma}_W^2$ the ML estimate of σ_W^2? Is it the MAP estimate? Justify your answer.

2.3 Iterative Coordinate Descent Optimization (ICD)

The iterative coordinate decent (ICD) algorithm works by sequentially minimizing the cost function of (14) with respect to each pixel. The update of pixel x_i is given by the following expression

$$x_i \leftarrow \arg\min_u \left\{ \frac{1}{2\sigma_W^2}(y_i - u)^2 + \frac{1}{2\sigma_x^2} \sum_{j \in \partial i} g_{i,j}(u - x_j)^2 \right\}$$

Differentiating this expression and solving for the minimum results in the ICD update equation

$$x_i \leftarrow \frac{y_i + \frac{\sigma_W^2}{\sigma_x^2} \sum_{j \in \partial i} g_{i,j} x_j}{1 + \frac{\sigma_W^2}{\sigma_x^2}}$$

In many problems, it is known for physical reasons that the image x must be positive (e.g. reflected or emitted light energy must be positive). In this case, it is useful to impose a positivity constraint on the solution since this constraint can improve the restoration quality. This can be done very simply by constraining the ICD updates to be positive.

$$x_i \leftarrow \max \left\{ 0, \frac{y_i + \frac{\sigma_W^2}{\sigma_x^2} \sum_{j \in \partial i} g_{i,j} x_j}{1 + \frac{\sigma_W^2}{\sigma_x^2}} \right\}$$

In this case, the standard ICD algorithm then has the form

Standard ICD Algorithm:

```
1. Set $K$ = desired number of iterations
2. Select desired values of $\sigma_x$ and $\sigma_W$
3. For each $i \in S$ /* Initialize with ML estimate */
       $x_s \leftarrow y_i$
4. For $k = 0$ to $K - 1$
   (a) For each $i \in S$
           $x_i \leftarrow \max \left\{ 0, \frac{y_i + \frac{\sigma_W^2}{\sigma_x^2} \sum_{j \in \partial i} g_{i,j} x_j}{1 + \frac{\sigma_W^2}{\sigma_x^2}} \right\}$
```

Section Problems:

1. Show that the costs resulting from ICD updates forms a monotone decreasing sequence that is bounded below.

2. Show that any local minimum of the cost function of (15) is also a global minimum.

3. Use the monochrome image img04.tif as x and produce a noisy image y by adding i.i.d. Gaussian noise with mean zero and $\sigma_W^2 = 16^2$. Approximate y by truncating the pixels to the range $[0, \cdots 255]$. Print out the image Y.

4. Compute the MAP estimate of X using 20 iterations of ICD optimization. Use $\sigma_x^2 = \hat{\sigma}_x^2$ the ML estimate of the scale parameter computed for $p = 2$, and $\sigma_W^2 = 16^2$. Print out the resulting MAP estimate.

5. Plot the cost function of (14) as a function of the iteration number for the experiment of step 4.

6. Repeat step 4 for $\sigma_x^2 = 5 * \hat{\sigma}_x^2$, and $\sigma_x^2 = (1/5) * \hat{\sigma}_x^2$.

2.4 MAP Restoration from Blurred/Noisy Image with Gaussian Prior

In this section, you will restore an image that has been blurred and then corrupted by additive Gaussian noise. For this problem, it is best to think of the images Y and X as being N dimensional column vectors formed by listing the pixels in raster order. In this case, a linear space-invariant filter can be represented by a $N \times N$ matrix transform H. So Y is given by

$$Y = H\,X + W$$

where H is a circulant-block-circulant matrix that implements a 2-D shift invariant filter applied with circular boundary conditions that has an impulse response of

1/81	2/81	3/81	2/81	1/81
2/81	4/81	6/81	4/81	2/81
3/81	6/81	9/81	6/81	3/81
2/81	4/81	6/81	4/81	2/81
1/81	2/81	3/81	2/81	1/81

(22)

where the 9/81 term is at the center of the filter. In this case, the cost functional for MAP estimation has the form

$$c(x) = \frac{1}{2\sigma_W^2}||y - Hx||^2 + \frac{1}{2\sigma_x^2}\sum_{\{i,j\}\in\mathcal{C}} g_{i,j}(x_i - x_j)^2 \tag{23}$$

For this type of problem, the ICD iterations should be computed using a state recursion to reduce computation. First, we express the MAP estimate using matrix notation

$$\hat{x} = \arg\min_{x\geq 0}\left\{\frac{1}{2\sigma_W^2}||y - Hx||^2 + \frac{1}{2\sigma_x^2}\sum_{\{i,j\}\in\mathcal{C}} g_{i,j}(x_i - x_j)^2\right\}\ , \tag{24}$$

where $x \geq 0$ indicates that the optimization is constrained to have positive pixel values. By defining the error vector $e = y - Hx$, the ICD update for each pixel x_i then has the form

$$\begin{aligned} v &\leftarrow x_i && (25) \\ x_i &\leftarrow \arg\min_{u \geq 0} \left\{ \frac{1}{2\sigma_W^2} ||e - H_{*,i}(u - v)||^2 + \frac{1}{2\sigma_x^2} \sum_{j \in \partial i} g_{i,j}(u - x_j)^2 \right\} && (26) \\ e &\leftarrow e - H_{*,i}(x_i - v) && (27) \end{aligned}$$

where $H_{*,i}$ denotes the i^{th} column of the matrix H. The update of (26) can be more efficiently expressed by exploiting the quadratic structure of the data term, and using the relationship

$$x_i \leftarrow \arg\min_{u \geq 0} \left\{ \theta_1(u - v) + \frac{1}{2}\theta_2(u - v)^2 + \frac{1}{2\sigma_x^2} \sum_{j \in \partial i} g_{i,j}(u - x_j)^2 \right\} \qquad (28)$$

where

$$\begin{aligned} \theta_1 &= -\frac{e^t H_{*,i}}{\sigma_W^2} && (29) \\ \theta_2 &= \frac{||H_{*,i}||^2}{\sigma_W^2} && (30) \end{aligned}$$

This results in the following ICD algorithm when the prior distribution is Gaussian and a blurring filter is included in the forward model.

```
ICD Algorithm with Blurring Filter:
  1. Set $K$ = desired number of iterations
  2. Select desired values of $\sigma_x$ and $\sigma_W$
  3. Initialize $x \leftarrow y$
  4. Initialize $e \leftarrow y - Hx$
  5. For $k = 0$ to $K - 1$
     (a) For each $i \in S$
           $v \leftarrow x_i$
           $\theta_1 \leftarrow -\frac{e^t H_{*,i}}{\sigma_W^2}$
           $\theta_2 \leftarrow \frac{||H_{*,i}||^2}{\sigma_W^2}$
           $x_i \leftarrow \max\left\{0, \frac{\theta_2 v - \theta_1 + \frac{1}{\sigma_x^2}\sum_{j \in \partial i} g_{i,j} x_j}{\theta_2 + \frac{1}{\sigma_x^2}}\right\}$
           $e \leftarrow e - H_{*,i}(x_i - v)$
```

Section Problems:

1. Use the monochrome image img04.tif as X and produce a blurred and noisy image Y. To do this, first apply the blurring filter of (22) with circular boundary conditions; then add noise with a variance of $\sigma_W^2 = 4^2$. Approximate Y by truncating the pixels to the range $[0, \cdots, 255]$. Print out the image Y.

2. Compute the MAP estimate of X using 20 iterations for coordinate decent optimization with $\sigma_x^2 = \hat{\sigma}_x^2$ the ML estimate of σ_x for $p = 2$, and $\sigma_W^2 = 4^2$. Print out the resulting MAP estimate.

3. Plot the cost function of equation 23 as a function of iteration number.

3 MAP Estimation with Non-Gaussian Prior

In this section, we introduce some of the basic methods for computing the MAP estimate with non-Gaussian prior models. A key component of these methods is the use of computational efficient algorithms for solving the line-search problems that commonly arrise in non-quadratic optimization.

3.1 Basic Techniques for MAP Restoration with non-Gaussian Prior

In this section, you will compute the MAP estimate of a image with linear blurring and noise as in the previous section, but in addition, you will use a non-Gaussian prior model formed by a GGMRF with $p = 1.20$. For this problem, the cost function to be minimized has the form

$$c(x) = \frac{1}{2\sigma_W^2}||y - Hx||^2 + \frac{1}{p\sigma_x^p} \sum_{\{i,j\}\in\mathcal{C}} g_{i,j}|x_i - x_j|^p \tag{31}$$

and the MAP estimate is given by

$$\hat{x} = \arg\min_{x\geq 0} \left\{ \frac{1}{2\sigma_W^2}||y - Hx||^2 + \frac{1}{p\sigma_x^p} \sum_{\{i,j\}\in\mathcal{C}} g_{i,j}|x_i - x_j|^p \right\} , \tag{32}$$

In this case, the update of each pixel does not have a closed form, so it can only be expressed in terms of a minimization with respect to a scalar value u. This results in the following ICD algorithm with blurring filter H and GGMRF prior.

ICD Algorithm with Blurring Filter and GGMRF Prior:

```
1. Set $K =$ desired number of iterations
2. Select desired values of $\sigma_x$ and $\sigma_W$
3. Initialize $x \leftarrow y$
4. Initialize $e \leftarrow y - Hx$
5. For $k = 0$ to $K - 1$
   (a) For each $i \in S$
         $v \leftarrow x_i$
         $\theta_1 \leftarrow -\frac{e^t H_{*,i}}{\sigma_W^2}$
         $\theta_2 \leftarrow \frac{||H_{*,i}||^2}{\sigma_W^2}$
         $x_i \leftarrow \arg\min_{u \geq 0} \left\{ \theta_1(u-v) + \frac{1}{2}\theta_2(u-v)^2 + \frac{1}{p\sigma_x^p} \sum_{j \in \partial i} g_{i,j} |u - x_j|^p \right\}$
         $e \leftarrow e - H_{*,i}(x_i - v)$
```

The key to implementing these equations is the computation of the minimization in equation (33). This can be done by rooting the derivative of the equation.

$$\theta_1 + \theta_2(u-v) + \frac{1}{\sigma_x^p} \sum_{j \in \partial i} g_{i,j} |u - x_j|^{p-1} \mathrm{sign}(u - x_j) = 0 \tag{33}$$

When the weights $g_{i,j}$ are positive, then the solution of this equation must fall in the range $[low, high]$ where

$$low = \min \left\{ x_j \text{ for } j \in \partial i, \left(v - \frac{\theta_1}{\theta_2} \right) \right\} \tag{34}$$

$$high = \max \left\{ x_j \text{ for } j \in \partial i, \left(v - \frac{\theta_1}{\theta_2} \right) \right\} \tag{35}$$

Using these bounds, the solution to (33) can be computed using a half interval search method. The routines *solve.c* and *solve.h* are included in the C-code provided for this laboratory. These routines can be used to root any expression given initial *low* and *high* starting points. The syntax of the *solve* routine is:

```
double  solve(
  double (*f) (double x, void * pblock),
  void * pblock, /* pointer to structure containing function parameters */
  double a,       /* low bound on solution */
  double b,       /* upper bound on solution */
  double err,     /* accuarcy of solution */
  int *code       /* error code */
  )
```

The variable `f` is a function pointer which should point to the a function that evaluates expression to be rooted; in this case, the left-hand-side of (33). The function `f` should be written so that it is a function of a single argument x and a pointer to structure containing any additional parameters. For this problem, the structure should contain the parameters θ_1 along θ_2 with the neighboring pixel values. The variables `a` and `b` are used to give *low* and *high* values to start the search, and `err` is a variable which specifies the precision of the rooting operation. The variable `code` is used to return error codes when the search is not successful.

An example named `SolveExample` is provided in along with the C-code repository that shows how to use the `solve` function for this section.

Section Problems:

1. Use the noisy and blurred image from Section 2.4 as the image y, and compute the MAP estimate of X using 20 iterations of coordinate decent optimization with $p = 1.20$, $\sigma_x^{1.2} = \hat{\sigma}_x^{1.2}$ (i.e. the ML estimate of σ for $p = 1.2$), $\sigma_W^2 = 4^2$, and `err = 1e-7`. Print out the resulting MAP estimate.

2. Produce two restorations using 20 ICD iterations with the parameters of problem 1 above, but with $\sigma_x = 5*\hat{\sigma}_x$ and $\sigma_x = (1/5)*\hat{\sigma}_x$. Print out the resulting MAP estimates.

3. Plot the cost function of equation 31 as a function of iteration number.

3.2 MAP Restoration using Majorization to Optimize non-Gaussian Cost Function

In this section you will restore the same image that you restored in the previous section, except this time you will use majorization techniques to solve the associated optimization problem. You should find that using majorization makes your program run quite a bit faster while converging to the same solution.

For this part of the lab, we will be using the **q-Generalized GMRF (QGGMRF)** as our prior model rather than the GGMRF. The cost function to be minimized when using the QGGMRF prior is given by:

$$c(x) = \frac{1}{2\sigma_W^2}||y - Hx||^2 + \frac{1}{p\sigma_x^p}\sum_{\{i,j\}\in\mathcal{C}} g_{i,j}|x_i - x_j|^p \left(\frac{\left|\frac{x_i - x_j}{T\sigma_x}\right|^{q-p}}{1 + \left|\frac{x_i - x_j}{T\sigma_x}\right|^{q-p}} \right) \tag{36}$$

Observe that the prior term in the above cost function is not generally quadratic. However, if there was some way to convert the minimization of such a function into the minimization of a quadratic function (similar the the form of the minimization in Section 2.4), we could abandon the `solve` line search routine in favor of a simpler (and faster) method. It turns out that we can find a quadratic function to *substitute* for equation 36. With this goal in mind, we turn to the **symmetric bound** method of majorization in order to find this quadratic substitute function. We begin by noting that each potential function of the form

$$p(x) = \frac{1}{z}\left\{ - \sum_{\{s,r\}\in\mathcal{P}} b_{s,r}\rho(x_s - x_r) \right\} \tag{37}$$

is upper bounded by a surrogate function that is a symmetric and quadratic function of Δ, where Δ is defined as $x_s - x_r$. Using this assumption, such a surrogate function has the form

$$\rho(\Delta;\Delta') = \frac{a_2}{2}\Delta^2 \tag{38}$$

where a_2 is a function of Δ'. In order to determine a_2, we match the gradients of $\rho(\Delta)$ and $\rho(\Delta;\Delta')$ at $\Delta = \Delta'$ to yield

$$a_2 = \frac{\rho'(\Delta')}{\Delta'} \tag{39}$$

which results in the following symmetric bound surrogate function,

$$\rho(\Delta;\Delta') = \begin{cases} \frac{\rho'(\Delta')}{2\Delta'} & \text{if } \Delta' \neq 0 \\ \\ \frac{\rho''(0)}{2} & \text{if } \Delta' = 0 \end{cases}.$$

It can subsequently be shown that the MAP surrogate cost function is given by

$$\begin{aligned} Q(x;\,x') &= \frac{1}{2\sigma_W^2}||y - Hx||^2 + \sum_{s,r\in\mathcal{P}} \frac{b_{s,r}\rho'(x'_s - x'_r)}{2(x'_s - x'_r)}(x_s - x_r)^2 && (40) \\ &= \frac{1}{2\sigma_W^2}||y - Hx||^2 + \sum_{s,r\in\mathcal{P}} \tilde{b}_{s,r}(x_s - x_r)^2 && (41) \end{aligned}$$

The MAP estimate can then be computed through repeated minimization of equation 41. It is important to note that the expression for $\tilde{b}$ differs based on the choice of prior model. Recall that for this part of the lab, we will be using the QGGMRF as our prior model. The QGGMRF specification for $\tilde{b}$ is given below:

$$\tilde{b}_{s,r} \leftarrow \begin{cases} \frac{b_{s,r}}{p\sigma_x^p} & x'_s - x'_r = 0\ ;\ p = 1 \\ \\ b_{s,r}\frac{|x'_s - x'_r|^{p-2}}{2\sigma_x^p}\frac{\left|\frac{x'_s - x'_r}{T\sigma_x}\right|^{q-p}\left(\frac{q}{p} + \left|\frac{x'_s - x'_r}{T\sigma_x}\right|^{q-p}\right)}{\left(1 + \left|\frac{x'_s - x'_r}{T\sigma_x}\right|^{q-p}\right)^2} & else \end{cases}.$$

It turns out that a common choice for the q parameter is $q = 2$, and as such we will require $q = 2$ for all parts of this section. An effective pseudocode specification of the algorithm for such a minimization is specified on the following page. The above QGGMRF specification for $\tilde{b}$ is the one you will use in the first step of the algorithm's inner loop. The routines *get_btilde.c* and *get_btilde.h* are included in the C-code provided for this laboratory. These routines can be used to compute and return the value of $\tilde{b}$ for a given $\Delta = x'_s - x'_r$. The syntax of the *get_btilde* routine is:

```
double  get_btilde(
  double delta,        /* difference between pixel x_s and its neighbor x_r */
  double b,            /* neighborhood relationship coefficient */
  double sigma_x,      /* scale parameter */
  double p,            /* shape parameter of qggmrf */
  double q,            /* "       "       "       " */
  double T             /* threshold parameter of qggmrf */
  )
```

Also note that $\tilde{b}_{s,r}$ must be computed for each neighbor of the current pixel x_s in order to compute θ_1 and θ_2. Please see Chapter 7 (particularly Section 7.3, p. 147) of Dr. Bouman's MBIP book for more details and theory.

ICD Algorithm with Majorization of GGMRF Prior:

```
1. Set $K$ = desired number of iterations
2. Select desired values of $\sigma_x$ and $\sigma_W$
3. Initialize $x \leftarrow y$
4. Initialize $e \leftarrow y - Hx$
5. For $k = 0$ to $K - 1$
    (a) For each $s \in S$
            $\tilde{b}_{s,r} \leftarrow b_{s,r} \frac{\rho'(x'_s - x'_r)}{2(x'_s - x'_r)} \quad \forall r \in \partial i$
            $\theta_1 \leftarrow -\frac{e^t H_{*,s}}{\sigma_W^2} + \sum_{r \in \partial s} 2\tilde{b}_{s,r}(x_s - x_r)$
            $\theta_2 \leftarrow \frac{||H_{*,s}||^2}{\sigma_W^2} + \sum_{r \in \partial s} 2\tilde{b}_{s,r}$
            $\alpha^* \leftarrow \text{clip}\left\{ \frac{-\theta_1}{\theta_2}, [-x_s, \infty) \right\}$
            $x_s \leftarrow x_s + \alpha^*$
            $e \leftarrow e - H_{*,s}\alpha^*$
```

Section Problems:

1. Use the same noisy and blurred image from Sections 2.4 and 3.1 as the image y, and compute the MAP estimate of X using 20 iterations of coordinate decent optimization with symmetric bound majorization as specified above. Use the parameters $p = 1.2$, $q = 2$, $\sigma_x^{1.2} = \hat{\sigma}_x^{1.2}$ (i.e. the ML estimate of σ for $p = 1.2$), and $\sigma_W^2 = 4^2$. Print out the resulting MAP estimate.

2. Produce two restorations using 20 ICD iterations with the parameters of problem 1 above, but with $\sigma_x = 5 * \hat{\sigma}_x$ and $\sigma_x = (1/5) * \hat{\sigma}_x$. Print out the resulting MAP estimates.

3. Plot the cost function of equation 36 as a function of iteration number.