An Efficient Parallel Algorithm for High Resolution Color Image Reconstruction

Michael. K. Ng
Department of Mathematics
The University of Hong Kong
Pokfulam Road, Hong Kong

Abstract

This paper studies the application of preconditioned conjugate gradient methods in high resolution color image reconstruction problems. The high resolution color images are reconstructed from multiple undersampled, shifted, degraded color frames with subpixel displacements. The resulting degradation matrices are spatially variant. The preconditioners are derived by taking the cosine transform approximation of the degradation matrices. The resulting preconditioning matrices allow the use of fast transform methods. We show how the methods can be implemented on parallel computers, and we demonstrate their parallel efficiency using experiments on a sixteen processor IBM SP-2.

1 Introduction

Using digital signal processing techniques to improve the spatial resolution of images, such as those obtained from satellites, is of great practical importance, see for instance [1, 10, 12, 13, 14]. In this paper, we consider the reconstruction of high resolution color images from multiple undersampled, shifted, degraded and noisy color images which are obtained by using multiple identical color image sensors shifted from each other by subpixel displacements. We remark that color can be regarded as a set of three images in their primary color components: red, green and blue.

Consider a sensor array with $L_1 \times L_2$ sensors, each sensor has $N_1 \times N_2$ sensing elements (pixels) and the size of each sensing element is $T_1 \times T_1$. Our aim is to reconstruct an image of resolution $M_1 \times M_2$, where $M_1 = L_1 \times N_1$ and $M_2 = L_2 \times N_2$. To maintain the aspect ratio of the reconstructed image, we consider the case where $L_1 = L_2 = L$ only. For simplicity, we assume that L is an even number in the following discussion.

In order to have enough information to resolve the high resolution image, there are subpixel displacements between the sensors. In the ideal case, the sensors are shifted from each other by a value proportional to $T_1/L \times T_2/L$. However, in practice there can be small perturbations around these ideal subpixel locations due to imperfection of the mechanical imaging system. Thus, for $l_1, l_2 = 0, 1, \cdots, L-1$ with $(l_1, l_2) \neq (0,0)$, the horizontal and vertical displacements $d^x_{l_1 l_2}$ and $d^y_{l_1 l_2}$ of the $[l_1, l_2]$-th sensor array with respect to the $[0,0]$-th reference sensor array are given by

$$d^x_{l_1 l_2} = \frac{T_1}{L}(l_1 + \epsilon^x_{l_1 l_2}) \quad \text{and} \quad d^y_{l_1 l_2} = \frac{T_2}{L}(l_2 + \epsilon^y_{l_1 l_2}).$$

Here $\epsilon^x_{l_1 l_2}$ and $\epsilon^y_{l_1 l_2}$ denote respectively the normalized horizontal and vertical displacement errors.

We remark that the parameters $\epsilon^x_{l_1 l_2}$ and $\epsilon^y_{l_1 l_2}$ can be obtained by manufacturers during camera calibration. We assume that

$$|\epsilon^x_{l_1 l_2}| < \frac{1}{2} \quad \text{and} \quad |\epsilon^y_{l_1 l_2}| < \frac{1}{2}, \qquad 0 \leq l_1, l_2 \leq L-1. \tag{1}$$

For if not, the low resolution images observed from two different sensor arrays will be overlapped so much that the reconstruction of the high resolution image is rendered impossible.

Let $f^{(r)}$, $f^{(g)}$ and $f^{(b)}$ be the original scene in red, green and blue channels respectively Then the observed low resolution image in the i-th ($i \in \{r, g, b\}$) channel $g^{(i)}_{l_1 l_2}$ for the (l_1, l_2)-th sensor is modeled by:

$$\begin{aligned} & g^{(i)}_{l_1 l_2}[n_1, n_2] \\ = & \sum_{j \in \{r,g,b\}} w_{ij} \left\{ \int_{T_2(n_2 - \frac{1}{2}) + d^y_{l_1 l_2}}^{T_2(n_2 + \frac{1}{2}) + d^y_{l_1 l_2}} \int_{T_1(n_1 - \frac{1}{2}) + d^x_{l_1 l_2}}^{T_1(n_1 + \frac{1}{2}) + d^x_{l_1 l_2}} \right. \\ & \left. f^{(j)}(x_1, x_2) dx_1 dx_2 \right\} + \eta^{(i)}_{l_1 l_2}[n_1, n_2], \end{aligned} \tag{2}$$

for $n_1 = 1, \ldots, N_1$ and $n_2 = 1, \ldots, N_2$. Here $\eta^{(i)}_{l_1 l_2}$ is the noise corresponding to the (l_1, l_2)-th sensor in the

0-7695-0571-6/00 $10.00 © 2000 IEEE

i-th channel, and w_{ii} and w_{ij} $(i \neq j)$ are the within-channel and the cross-channel degradation parameters. We note that

$$w_{ij} \geq 0,\ i,j \in \{r,g,b\} \text{ and } \sum_{j=r,g,b} w_{ij} = 1,\ i \in \{r,g,b\}. \tag{3}$$

To get the operator representation (4), we intersperse the low resolution images $g^{(i)}_{l_1 l_2}[n_1, n_2]$ to form an $M_1 \times M_2$ image by assigning

$$g^{(i)}[L(n_1-1)+l_1, L(n_2-1)+l_2] = g^{(i)}_{l_1 l_2}[n_1, n_2],$$

$(i \in \{r,g,b\})$. The image $g^{(i)}$ so formed is called the *observed high resolution image from the i-th channel.* Similarly, we define $\eta^{(i)}$. Using a column by column ordering for $g^{(i)}$, $f^{(i)}$ and $\eta^{(i)}$, (2) becomes

$$g^{(i)} = \sum_{j \in \{r,g,b\}} w_{ij} \mathcal{H} f^{(j)} + \eta^{(i)}, \quad i \in \{r,g,b\}. \tag{4}$$

1.1 The Discrete Model

The above continuous image model can be discretized by the rectangular rule and approximated by a discrete image model. Because (2) is a blurring process, the boundary values of $g^{(i)}$ are also affected by the values of $f^{(r)}$, $f^{(g)}$ and $f^{(b)}$ outside the scene. Thus in order to find f, we need some assumptions on the values of $f^{(r)}$, $f^{(g)}$ and $f^{(b)}$ outside the scene. Usual assumptions are the periodic boundary condition and the zero boundary condition. In [2], we proposed to use the Neumann boundary condition. It assumes that the scene immediately outside is a reflection of the original scene at the boundary. For grey-level image reconstruction problems, it gives better reconstructed images than that by the zero or periodic boundary conditions, see [2, 9].

For $i \in \{r,g,b\}$, let $\mathbf{g}^{(i)}$ and $\mathbf{f}^{(i)}$ be respectively the discretization of $g^{(i)}$ and $f^{(i)}$ using a column by column ordering. Let

$$\mathbf{g} = [\mathbf{g}^{(r)}\ \mathbf{g}^{(g)}\ \mathbf{g}^{(b)}]^t \quad \text{and} \quad \mathbf{f} = [\mathbf{f}^{(r)}\ \mathbf{f}^{(g)}\ \mathbf{f}^{(b)}]^t.$$

Under the Neumann boundary condition assumption, the degradation matrices $\mathbf{H}_{l_1 l_2}(\epsilon) = \mathbf{H}^x_{l_1 l_2}(\epsilon) \otimes \mathbf{H}^y_{l_1 l_2}(\epsilon)$ in each channel, where $\mathbf{H}^x_{l_1 l_2}(\epsilon)$ is a banded Toeplitz-plus-Hankel matrices with bandwidth $2L-1$:

$$\frac{1}{L}\begin{pmatrix} 1 & \cdots & 1 & h^{x+}_{l_1 l_2} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ 1 & \ddots & \ddots & \ddots & \ddots & h^{x+}_{l_1 l_2} \\ h^{x-}_{l_1 l_2} & \ddots & \ddots & \ddots & \ddots & 1 \\ & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & & h^{x-}_{l_1 l_2} & 1 & \cdots & 1 \end{pmatrix} +$$

$$\frac{1}{L}\begin{pmatrix} 1 & \cdots & 1 & h^{x-}_{l_1 l_2} & & 0 \\ \vdots & \ddots & \ddots & & & \\ 1 & \ddots & & & & h^{x+}_{l_1 l_2} \\ h^{x-}_{l_1 l_2} & & & & \ddots & 1 \\ & & & \ddots & \ddots & \vdots \\ 0 & & h^{x+}_{l_1 l_2} & 1 & \cdots & 1 \end{pmatrix}. \tag{5}$$

$\mathbf{H}^y_{l_1 l_2}(\epsilon)$ is defined similarly. The degradation matrix for the whole sensor array is made up of degradation matrices from each sensor:

$$\mathbf{H}_L(\epsilon) = \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbf{D}_{l_1 l_2} \mathbf{H}_{l_1 l_2}(\epsilon), \quad i,j \in \{r,b,g\}. \tag{6}$$

Here $\mathbf{D}_{l_1 l_2}$ are diagonal matrices with diagonal elements equal to 1 if the corresponding component of the observed low resolution image comes from the (l_1, l_2)-th sensor and zero otherwise, see [1] for more details. We have the same matrix $\mathbf{H}_L(\epsilon)$ within the channels and across the channel, therefore the overall degradation matrix is given by

$$\mathbf{A}_L(\epsilon) = \begin{pmatrix} w_{rr} & w_{rg} & w_{rb} \\ w_{gr} & w_{gg} & w_{gb} \\ w_{br} & w_{bg} & w_{bb} \end{pmatrix} \otimes \mathbf{H}_L(\epsilon) \equiv \mathbf{W} \otimes \mathbf{H}_L(\epsilon). \tag{7}$$

Since the matrix $\mathbf{A}_L(\epsilon)$ is ill-conditioned, a regularization procedure should be imposed to obtain a reasonable estimate of the original image.

1.2 Regularization

Since the system (4) is ill-conditioned and generally not positive definite, we solve it by using a minimization and regularization technique. In [5, 6], Galatsanos et al. have proposed the following weighted discrete Laplacian matrix $\mathbf{R}$ as the regularization matrix:

$$\begin{aligned} & [\mathbf{Rf}]_{r,j,k} \\ = \ & 6[\mathbf{f}^{(r)}]_{j,k} - [\mathbf{f}^{(r)}]_{j-1,k} - [\mathbf{f}^{(r)}]_{j+1,k} - [\mathbf{f}^{(r)}]_{j,k-1} - \\ & [\mathbf{f}^{(r)}]_{j,k+1} - \frac{||\tilde{\mathbf{f}}^{(r)}||_2}{||\tilde{\mathbf{f}}^{(g)}||_2}[\mathbf{f}^{(g)}]_{j,k} - \frac{||\tilde{\mathbf{f}}^{(r)}||_2}{||\tilde{\mathbf{f}}^{(b)}||_2}[\mathbf{f}^{(b)}]_{j,k}, \end{aligned}$$

for $1 \leq j \leq M_1$ and $1 \leq k \leq M_2$. The entries $[\mathbf{Rf}]_{g,j,k}$ $[\mathbf{Rf}]_{b,j,k}$ can be defined similarly. Here $||\tilde{\mathbf{f}}^{(r)}||_2$, $||\tilde{\mathbf{f}}^{(g)}||_2$ and $||\tilde{\mathbf{f}}^{(b)}||_2$ are the estimates of the $||\mathbf{f}^{(r)}||_2$, $||\mathbf{f}^{(g)}||_2$ and $||\mathbf{f}^{(b)}||_2$ respectively and are assumed to be nonzero. The cross-channel weights of this regularization matrix capture the changes of reflectivity across the channels.

Using Tikhonov regularization, our discretization problem becomes:

$$(\mathbf{A}_L(\epsilon)^t \Upsilon \mathbf{A}_L(\epsilon) + \mathbf{R}^t \mathbf{R})\mathbf{f} = \mathbf{A}_L(\epsilon)^t \Upsilon \mathbf{g}, \tag{8}$$

where $\mathbf{A}_L(\epsilon)$ is given in (7),

$$\Upsilon = \begin{pmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{pmatrix} \otimes \mathbf{I} \equiv \Omega \otimes \mathbf{I},$$

and α_r, α_g and α_b are the regularization parameters which are assumed to be positive scalars.

In the case of gray-level image reconstruction, we have already developed a fast algorithm that is based on the preconditioned conjugate gradient method with cosine transform preconditioners, see [2, 11]. In particular, we have shown that when the L_2 or H_1 norm regularization functional is used, the spectra of the preconditioned normal systems are clustered around one and hence the conjugate gradient method converges very quickly.

The main aim of this paper is to extend our results in [11] from grey-level images to color images which are vector-valued gray-level images. We will extend our fast and stable gray-level image processing algorithm with cosine transform preconditioners to the color image reconstruction problems. We also show how the method can be implemented on parallel computers, and we demonstrate their parallel efficiency using experiments on a sixteen process IBM SP-2.

The outline of the paper is as follows. In Section 2, we introduce our cosine transform preconditioners. In Section 3, the parallel implementation of the algorithm is given. Finally, numerical results are presented to demonstrate the effectiveness of our method on parallel computers in Section 4.

2 Cosine Transform Based Preconditioners

The linear system (8) will be solved by using the preconditioned conjugate gradient method. In this section, we construct the cosine transform preconditioner of $\mathbf{A}_L(\epsilon)$ which exploits the banded and block structures of the matrix.

Let $\mathbf{C}_n$ be the $n \times n$ discrete cosine transform matrix, i.e., the (i,j)-th entry of $\mathbf{C}_n$ is given by

$$\sqrt{\frac{2-\delta_{i1}}{n}} \cos\left(\frac{(i-1)(2j-1)\pi}{2n}\right), \qquad 1 \le i,j \le n,$$

where δ_{ij} is the Kronecker delta. Note that the matrix-vector product $\mathbf{C}_n\mathbf{z}$ can be computed in $O(n \log n)$ operations for any vector $\mathbf{z}$, see [13, pp. 59–60]. For an $M_1 \times M_1$ block matrix $\mathbf{H}_L(\epsilon)$ with the size of each block equal to $M_2 \times M_2$, the cosine transform preconditioner $c(\mathbf{H}_L(\epsilon))$ of $\mathbf{H}_L(\epsilon)$ is defined to be the matrix $(\mathbf{C}_{M_1}^t \otimes \mathbf{C}_{M_2}^t)\Phi(\mathbf{C}_{M_1} \otimes \mathbf{C}_{M_2})$ that minimizes

$$\|(\mathbf{C}_{M_1}^t \otimes \mathbf{C}_{M_2}^t)\Phi(\mathbf{C}_{M_1} \otimes \mathbf{C}_{M_2}) - \mathbf{H}_L(\epsilon)\|_F$$

over all diagonal matrices Φ, where $\|\cdot\|_F$ is the Frobenius norm, see [4]. Clearly, the cost of computing $c(\mathbf{H}_L(\epsilon))^{-1}\mathbf{y}$ for any vector $\mathbf{y}$ is $O(M_1M_2 \log M_1M_2)$ operations. Since $\mathbf{H}_L(\epsilon)$ in (6) is a banded matrix with $(2L-1)^2$ non-zero diagonals and is of size $M_1M_2 \times M_1M_2$, the cost of constructing $c(\mathbf{H}_L(\epsilon))$ is of $O(L^2M_1M_2)$ operations only, see [3].

We will employ the cosine transform preconditioner $c(\mathbf{H}_L(\epsilon))$ for $\mathbf{H}_L(\epsilon)$. More precisely, we solve the following preconditioned system

$$\begin{aligned} & [\mathbf{W}^t\Omega\mathbf{W} \otimes c(\mathbf{H}_L(\epsilon))^t c(\mathbf{H}_L(\epsilon)) + \mathbf{R}^t\mathbf{R}]^{-1} \\ & [\mathbf{W}^t\Omega\mathbf{W} \otimes \mathbf{H}_L(\epsilon)^t\mathbf{H}_L(\epsilon) + \mathbf{R}^t\mathbf{R}]\mathbf{f} \\ = \; & [\mathbf{W}^t\Omega \otimes \mathbf{H}_L(\epsilon)^t]\mathbf{g}. \end{aligned} \tag{9}$$

In [2, 11], we have proposed to use cosine transform preconditioners to precondition the linear systems arising from the grey-level image reconstruction. Numerical results have shown that these preconditioners are effective.

Regarding the cost per iteration, the main work in each iteration for the conjugate gradient method is the matrix-vector multiplication, see for instance Golub and van Loan [7]. Since $\mathbf{W}$ is a 3-by-3 matrix and $\mathbf{H}_L(\epsilon)$ is a banded matrix, the matrix-vector multiplication $\mathbf{y} = [\mathbf{W}^t\Omega\mathbf{W} \otimes \mathbf{H}_L(\epsilon)^t\mathbf{H}_L(\epsilon) + \mathbf{R}^t\mathbf{R}]\mathbf{v}$ can be done very fast. Recall that if we use the Neumann boundary condition for both $\mathbf{L}$, then the matrices $\mathbf{L}$ and $c(\mathbf{H}_L(\epsilon))$ can be diagonalized by discrete cosine transform matrices. The corresponding system is a block-diagonalized system of M_1M_2 decoupled subsystems. The solution vector for the preconditioner can be computed by solving a set of M_1M_2 decoupled 3-by-3 matrix equations by a permutation. The total cost per iteration is therefore of $O(M_1M_2 \log M_1M_2)$ operations.

3 Parallel Implementation

The algorithm developed in Section 2 are well adapted for parallel computation in both shared memory and message-passing environments. We have implemented it on IBM SP-2 computer assuming that the number of processors does not exceed M_1 or M_2. In this section, we describe the implementation. For ease of illustration, we describe our implementation under the assumption that there are $p = 4$ processors. The pixel values are partitioned among processors as in Figure 1; if M_1 is divisible by p then each processor contains M_1/p blocks of $3M_2$ pixel values oriented along the x-y plane. If M_1 is not divisible by p, the

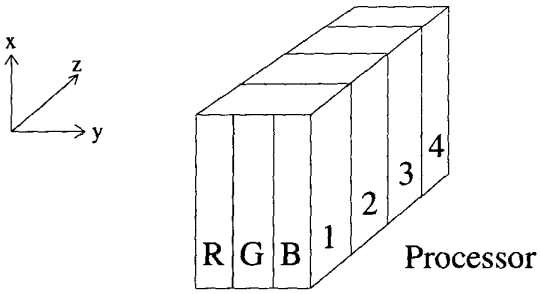

Figure 1: Partitioning of the pixel values in RGB channels among 4 processors.

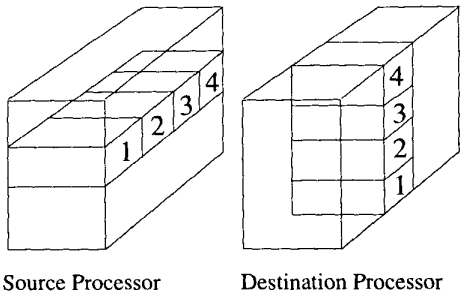

Figure 2: Source and destination processors for data movement, from the perspective of Processor 2.

number of x-y planes assigned to each processor differs by at most one.

3.1 Parallel Implementation of the Conjugate Gradient Method

We use the original form of the conjugate gradient algorithm as presented as in [7]. With the partitioning described above, the conjugate gradient algorithm is easy to parallelize: each processor is responsible for storing and updating at most

$$\hat{M}_1 \equiv \left\lceil \frac{M_1}{p} \right\rceil 3M_2$$

unknowns. Communication is necessary only for inner products and matrix-vector products. The cost per iteration is $O(\hat{M}_1)$ floating point operations plus the cost of matrix-vector products and preconditioning.

3.2 Parallel Implementation of Matrix-vector Products

Computation of products of the matrix in (7) with a vector requires that each processor send its highest-numbered x-y plane to the processor numbered $L/2$ greater, and its lowest-numbered x-y plane to the processor numbered $L/2$ less (if these processors exist). We remark that we consider a sensor array with $L \times L$ sensors. The stencils corresponding to the matrices $\mathbf{H}^x_{l_1 l_2}(\epsilon) \otimes \mathbf{H}^y_{l_1 l_2}(\epsilon)$ in (5) can then be applied to the local data. The cost per matrix multiply is $O(\hat{M}_1)$ floating point operations plus 2 sends and receives of $3LM_2/2$ numbers per processor.

3.3 Parallel Implementation of the Preconditioning

The preconditioning computation is a somewhat more complex operation. Let

$$\hat{M}_2 \equiv \left\lceil \frac{M_2}{p} \right\rceil 3M_1.$$

We arrange the work in the table below, estimating the cost assuming that M_1 and M_2 are powers of 2. We remark that for $M_1 = M_2 = M$, the total arithmetic cost for the cosine transform based preconditioner is proportional to $3M^2 \log M/p$, and the communication cost (assuming no contention for messages sent simultaneously) is proportional to $3M^2/p$.

Operation (Cost):
Step 1: Each processor computes cosine transforms in the x direction on its local data. ($O(\hat{M}_1 \log M_2)$ operations)
Step 2: The data is rearranged so that each processor has an approximately equal number of y-z planes. This requires a nontrivial amount of communication: the data movement from the perspective of Processor 2 is shown in Figure 2. (Each processor sends at most $3\lceil M_1/p \rceil \lceil M_2/p \rceil$ numbers to numbers to every other processor)
Step 3: Each processor then computes cosine transforms in z direction followed by solving $\lceil M_1 M_2/p \rceil$ linear systems of order 3, followed by inverse cosine transforms. (The cost is the cosine transforms with $O(\hat{M}_2 \log M_1)$ operations and solving 3-by-3 linear systems with $0(\lceil M_1 M_2 \rceil)$ operations)
Step 4: The data is rearranged to its original configuration. (Each processor sends at most $3\lceil M_1/p \rceil \lceil M_2/p \rceil$ numbers to every other processor)
Step 5: Each processor computes inverse cosine transforms in the x direction. ($O(\hat{M}_1 \log M_2)$ operations)

Table 1. Parallel implementation of the cosine transform preconditioning.

4 Numerical Results and Parallel Performance

4.1 Image Reconstruction Results

In this subsection, we first illustrate the effectiveness of using cosine transform preconditioners for solving high resolution color image reconstruction problems. The 128 × 128 original image is shown in Figure 1. The conjugate gradient method is employed to solving the preconditioned system (9). The cross-channel weights for $\mathbf{R}$ are computed from the observed high-resolution image, i.e.,

$$\|\tilde{\mathbf{f}}^{(i)}\|_2 = \|\mathbf{g}^{(i)}\|_2, \quad i \in \{r, g, b\}.$$

We tried the following degradation matrix to degrade the original color image

$$\begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{pmatrix} \otimes \mathbf{H}_L(\epsilon). \tag{10}$$

Gaussian white noises with signal-to-noise ratio of 30 dB were added to each degraded image plane.

In the tests, we used the same regularization parameter for each channel, i.e., $\alpha_r = \alpha_g = \alpha_b = \alpha$. The initial guess was the zero vector and the stopping criteria was $||\mathbf{r}^{(j)}||_2/||\mathbf{r}^{(0)}||_2 < 10^{-6}$, where $\mathbf{r}^{(j)}$ is the normal equations residual after j iterations. Table 1 shows the numbers of iterations required for convergence for $L = 4$, i.e., the number of sensor array used is 4×4. In the tables, "cos", "cir" or "no" signify that the cosine transform preconditioner, the level-2 circulant preconditioner [4] or no preconditioner is used respectively. We see from the tables that for the cosine transform preconditioner converges much faster than the circulant preconditioners for different M, where $M(= M_1 = M_2)$ is the size of the reconstructed image. Also the convergence rate is independent of M for fixed $\epsilon^{x,y}_{l_1 l_2}$. In all cases, the optimal regularization parameter α is chosen such that it minimizes the relative error. Here the relative error of the reconstructed image $\mathbf{f}_c$ to the original image $\mathbf{f}$ is defined as:

$$\sum_{i \in \{r,g,b\}} \frac{||\mathbf{f}^{(i)} - \mathbf{f}_c^{(i)}||_2}{||\mathbf{f}^{(i)}||_2}.$$

4.2 Parallel Performance

In this subsection, we describe the performance of the solvers using the transform based preconditioners on an IBM SP-2 with 48 nodes. Each node consists of a POWER2 RISC processor with a CPU clock rate of 160MHz and 128MB of main memory, and running the AIX operating system. The processors communicate with each other through a high performance switch with an aggregated peak bandwidth of 40MBps. The theoretical peak performance is 640 MFLOPS per processor and the aggregate peak performance is 30.72 GFLOPS. Because of some system management policy, only sixteen processors can be used. The computational component of the program was written in Fortran90 and compiled using the mpxlf90 compiler. All tests used double precision complex floating point computations. Communication was performed using MPI [8] with non-blocking sends (MPI_SEND) and blocking receives (MPI_RECV). Inner products were performed and broadcast using MPI_ALLREDUCE.

	$\epsilon^x_{l_1 l_2} = \epsilon^y_{l_1 l_2} = 0.1$			$\epsilon^x_{l_1 l_2} = \epsilon^y_{l_1 l_2} = 0.2$		
M	cos	cir	no	cos	cir	no
128	7	64	81	10	66	82
256	6	60	81	10	59	82
512	6	60	80	10	59	82
1024	6	59	81	10	59	82

Table 1: Number of iterations for the degradation matrix with $L = 4$.

The results on parallel performance are summarized in Table 3. The entries CPU times for solving the high resolution image reconstruction problem with cosine transform based preconditioners. The timings reflect the averages of over five runs. It is evident from these data that the preconditioned conjugate gradient method with cosine transform based preconditioners display a large amount of parallelism especially when M is large. Table 4 shows the total speedup, i.e., ratio of CPU time on 4 processors to CPU time on p processors, for different M. The typically larger speedups observed when M is large stem from the larger amount of arithmetic for this problem. We attribute this to the workload between the communication among processors and the amount of computation required by the cosine transform based preconditioner.

The preconditioners presented here enable efficient parallel solution of the high resolution color image reconstruction problem. Performance of the conjugate gradient method with the cosine transform based preconditioner is relatively insensitive to the number of pixel values and the subpixel displacement error. We

	$\epsilon^x_{l_1 l_2} = \epsilon^y_{l_1 l_2} = 0.1$			
p	$M = 128$	$M = 256$	$M = 512$	$M = 1024$
4	1.19	3.78	16.56	67.62
8	0.77	2.34	10.74	35.10
16	0.70	1.68	6.48	24.72
	$\epsilon^x_{l_1 l_2} = \epsilon^y_{l_1 l_2} = 0.2$			
p	$M = 128$	$M = 256$	$M = 512$	$M = 1024$
4	1.71	6.33	27.60	112.70
8	1.12	3.95	18.05	58.54
16	1.01	2.81	10.93	41.28

Table 2: CPU times for solving (8) with using cosine transform based preconditioner for different M.

	$\epsilon^x_{l_1 l_2} = \epsilon^y_{l_1 l_2} = 0.1$			
p	$M = 128$	$M = 256$	$M = 512$	$M = 1024$
8	1.55	1.62	1.54	1.93
16	1.70	2.25	2.56	2.74
	$\epsilon^x_{l_1 l_2} = \epsilon^y_{l_1 l_2} = 0.2$			
p	$M = 128$	$M = 256$	$M = 512$	$M = 1024$
8	1.53	1.60	1.53	1.93
16	1.69	2.25	2.53	2.73

Table 3: Speedup.

have tested our method on an IBM SP-2. The method displays high efficiencies in an implementation on a parallel computer.

Acknowledgments

The work of Michael Ng was supported by HKU 7147/99P, and HKU CRCG Grant Nos. 10201939 and 10202720.

References

[1] N. Bose and K. Boo, *High-resolution image reconstruction with multisensors*, International Journal of Imaging Systems and Technology, 9 (1998), pp. 294–304.

[2] R. Chan, T. Chan, M. Ng, W. Tang, and C. Wong, *Preconditioned iterative methods for high-resolution image reconstruction with multisensors*, Proceedings to the SPIE Symposium on Advanced Signal Processing: Algorithms, Architectures, and Implementations, Vol. 3461, San Diego CA, July, 1998, pp. 348–357, Ed: F. Luk.

[3] R. Chan, T. Chan, and C. Wong, *Cosine Transform Based Preconditioners for Total Variation Deblurring*, IEEE Trans. Image Proc., 8 (1999), pp. 1472–1478.

[4] R. Chan and M. Ng, *Conjugate gradient method for Toeplitz system*, SIAM Review, 38 (1996), pp. 427–482.

[5] N. Galatsanos, A. Katsaggelosm, R. Chin, and A. Hillery, *Least Squares Restoration of Mulitchannel Images*, IEEE Trans. Signal Processing, 39 (1991), pp. 2222–2236.

[6] N. Galatsanos and R. Chin, *Restoration of Color Images by Multichannel Kalman Filtering*, IEEE Trans. Signal Processing, 39 (1991), pp. 2237–2252.

[7] G. Golub and C. Van Loan, *Matrix Computations*, 2rd ed., The Johns Hopkins University Press, 1993.

[8] W. Gropp, E. Lusk and A. Skjellum, *Using MPI*, The MIT Press, London, 1994.

[9] F. Luk and D. Vandevoorde, *Reducing boundary distortion in image restoration*, Proc. SPIE 2296, Advanced Signal Processing Algorithms, Architectures and Implementations VI, 1994.

[10] S. Kim, N. Bose, and H. Valenzuela, *Recursive reconstruction of high resolution image from noisy undersampled multiframes*, IEEE Trans. on Acoust., Speech, and Signal Process., 38 (1990), pp. 1013–1027.

[11] M. Ng, R. Chan, T. Chan, and A. Yip, *Cosine Transform Preconditioners for High Resolution Image Reconstruction*, Res. Rept. 99-10, Dept. Math., The Chinese University of Hong Kong, or Linear Algebra Appls., to appear.

[12] R. Schultz and R. Stevenson, *Extraction of high-resolution frames from video sequences*, IEEE T. Image Proces., 5 (1996), pp. 996–1011.

[13] A. Tekalp, M. Ozkan, and M. Sezan, *High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration*, In Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process., III, pp. 169–172, San Francisco, CA, March 1992.

[14] R. Tsai and T. Huang, *Multiframe image restoration and registration*, Advances in Computer Vision and Image Processing, 1 (1984), pp. 317–339.