A Parallel Multigrid Poisson PDE Solver for Gigapixel Image Editing

Zhenlong Du, Xiaoli Li, Xiaojian Yang, and Kangkang Shen

College of Electronics & Information Engineering, Nanjing University of Technology, Nanjing 210009, P.R.China
duzhlcad@gmail.com

Abstract. With the development of image acquisition technology, gigapixel images are easily produced and widely used in modern society. How to efficiently compile these gigapixel images within gradient domain is the research focus in the community of image processing and computer graphics. To solve Poisson equations involving large-scale unknowns is crucial for gigapixel image editing in gradient domain. Traditional multigrid approach separately performs iteration, restriction and interpolation, bears heavy communication costs between RAM and external memory. In the paper, a parallel multigrid Poisson solver for gigapixel image editing is proposed, which exploits the locality and relevance of memory accessing and updating among the iteration, restriction and interpolation for parallel performing the iteration, restriction and interpolation in the sweeping window. Image stitching experiments show that the presented method exhibits the higher efficiency than the Poisson solver of successive overrelaxation, gauss-seider iteration and traditional multigrid.

Keywords: Poisson PDE solver, parallel multigrid, gigapixel image editing.

1 Introduction

Poisson partial differential equation (PDE) is one kind of elliptic PDE which is widely used in community of science and engineering, such as machinery, physics, information, etc. Since the introduction of Poisson PDE to the image editing by Prez [1], image editing within gradient domain based on Poisson PDE becomes the research focus of image editing, such as, image cloning and composition [2,8], photo montage [3], matting [4], all achieved the photorealistic editing effect. However, with the development of digital acquisition technology, the resolution of acquired images have been increasing, so the space and time complexity in image editing based on Poisson PDE becomes more and more higher. Therefore, the investigation of fast Poisson solver is very significant.

Iterative Poisson PDE solver converges fast in the high frequency region but slow in the low frequency region which easily leads to the solution to fall into the local smooth region, therefore, it is unsuitable for solving Poisson PDE with

Y. Zhang, K. Li, and Z. Xiao (Eds.): HPC 2012, CCIS 207, pp. 89–98, 2013.

large-scale unknowns. Multigrid Poisson solver respectively smoothes the error residual in different frequency levels, and has better efficiency than iteration approach, hence is widely used in gigapixel images editing within gradient domain. But the traditional multigrid approach separately performs iteration, restriction and interpolation, not fully exploits the relevance among data in different stages, bears heavy communication load between RAM and external memory.

A parallel [7] multigrid Poisson solver [6] is proposed which exploits the locality and relevance of memory accessing and updating among the stage of iteration, restriction and interpolation. The presented method in [6] parallel performs the iteration, restriction and interpolation in the sweeping window which effectively reduce the communication load between RAM and external memory and increase the processing efficiency. In the paper a parallel multigrid Poisson PDE solver is presented, which has better performance in solving Poisson equations with large-scale unknowns. The proposed method is testified in gigapixel image editing within gradient domain and demonstrates the high efficiency than iterative Poisson solver.

2 Poisson PDE Solver

The general form of binary Laplace PDE $U(x,y): R^2 \rightarrow R^2$($R$ is the real set) is as the follows.

$$\triangle\, U(x,y) = \nabla \cdot \nabla U(x,y) = f(x,y) \tag{1}$$

Where $\triangle$ is Laplace operator, ∇ is partial derivative operator, and $f(x,y)$ is the boundary condition. Eq.(1) is the Poisson PDE with Neuman boundary when $f(x,y) = 0$ holds, and it is the Poisson PDE with Dirichlet boundary when $f(x,y) = c$ (c is a constant) is satisfied.

In the domain $\Omega = R \times R$, the purpose of image editing within gradient domain is to make $U(x,y)$ best close to the gradient field $\overrightarrow{G}(x,y)$, it is equivalent to the functional min $\parallel U - \overrightarrow{G} \parallel$, and the Poisson solver is able to be derived and expressed as Eq (2).

$$\triangle = \nabla \cdot \nabla \overrightarrow{G} = div\overrightarrow{G} \tag{2}$$

where div is the divergence operator.

The numerical solver of Eq. (2) is firstly converted to the discrete representation. There exists many discretization schema, the five-point discretization is represented as the follows.

$$U(x+1,y) + U(x,y+1) + U(x-1,y) + U(x,y-1) - 4U(x,y) = div\overrightarrow{G}(x,y)$$

For each pixel (x,y) within the image domain Ω, there is a linear equation. The equations of all pixels form an linear equations, which is denoted by $\mathbf{LU} = \mathbf{B}$, in which $\mathbf{L} = (a_{ij})_{n\times n}$, $\mathbf{U} = (u_i)_{n\times 1}$ and $\mathbf{B} = (b_i)_{n\times 1}$ are separately the Laplace matrix, unknowns matrix and boundary matrix.

There are many numerical solving methods for Poisson equation such as iteration [5] (including gauss-seider iteration, Jacobi iteration and conjugate gradient

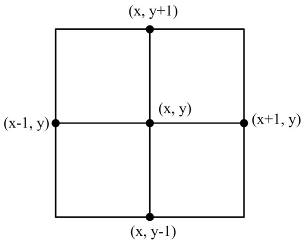

Fig. 1. Five-point discretization

iteration), multigrid [5] and discrete cosine transform, etc. When the number of unknowns in domain reaches 10^4, the size of L is $10^4 \times 10^4$. For the large-scale image(the number of unknowns are greater than 10^6), $\mathbf{L}$ is $10^6 \times 10^6$. Though $\mathbf{L}$ is the banded matrix, and generally it is sparse, the large-scale unknowns still deteriorates the iteration speed and convergence performance.

3 Multigrid Poisson PDE Solver

Multigrid Poisson PDE solver acquires the solution in hierarchical manner, which iterates and smoothes the high-frequency error residual in current level and continually smoothes the low-frequency residual in the next low-resolution level. The error residual is restricted from the higher resolution level to the next lower resolution one. The essence of the multigrid Poisson PDE solver is that smoothing and iterating the high-frequency error residual in the low-resolution level which are again restricted to the lower resolution level, repeats the process until the error residual is sufficiently smooth. Each iteration at a low-resolution level provides a more accurate calibration result for the next high-resolution level. Multigrid Poisson Solver could quickly smooth the high-frequency residual existing in different frequency spectrums, hence it efficiently accelerates the convergence procedure.

Let h be the discretization length. Linear equations discretized from Poisson PDE at h level is expressed as the follow.

$$\mathbf{L}_h \mathbf{U}_h = \mathbf{B}_h \tag{3}$$

Let $\mathbf{U}_h$ and $\overline{\mathbf{U}}_h$ separately be the accurate solution and approximate solution of Eq. (3), $\mathbf{V}_h$ and $\mathbf{d}_h$ separately be be the error quantity between $\mathbf{U}_h$ and $\overline{\mathbf{U}}_h$ and error residual, which are defined as $\mathbf{V}_h = \mathbf{U}_h - \overline{\mathbf{U}}_h$ and $\mathbf{d}_h = \mathbf{L}_h \overline{\mathbf{U}}_h$, respectively.

Multigrid Poisson solver performs in V-cycle manner, includes approximation, restriction and interpolation three procedures. It covers the following steps, solving the approximate solution $\overline{\mathbf{U}}_h$, restricting the error residual $\mathbf{d}_h$ to the lower resolution level H by Eq. (4), acquiring $\mathbf{V}_H$ by $\mathbf{L}_H \mathbf{V}_H = -\mathbf{d}_H$, returning the calibration to the higher resolution level h by interpolation, finally, solving the approximate solution at the highest resolution level, that is $\overline{\mathbf{U}}_h^{new}$ ($\overline{\mathbf{V}}_h$ is the approximation of $\mathbf{V}_h$). The restriction operator R and interpolation operator P used in multigrid solver are defined as Eq. (4) and Eq. (5) respectively.

$$\mathbf{d}_H = R\mathbf{d}_h \tag{4}$$

$$\overline{\mathbf{V}}_h = P\overline{\mathbf{V}}_H \tag{5}$$

The V-cycle in mulitigrid Poisson solver includes two procedures, one is a coarsening process from a high resolution level to the low resolution one, and the other is a refining process from the low resolution to the high resolution. Coarsening begins with the highest resolution level, restricts the error residual $\mathbf{d}$ to the next low-resolution level. Coarsening polishes the error level by level. When the lowest resolution level is reached, the linear equations with the minimal number of unknowns is solved. Refining begins with the lowest resolution level and returns the calibration result from lower resolution level to higher resolution level via interpolation until the solution within the highest resolution level $\mathbf{U}^{new}$ is achieved. Coarsening and refining separately correspond to the process of restriction and interpolation in the figure 2.

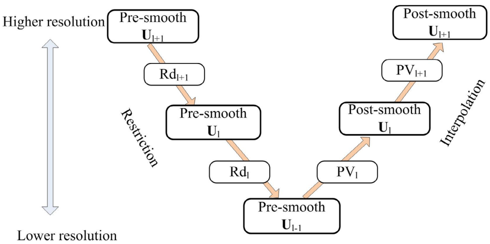

Fig. 2. The procedure of multigrid V-cycle

Both restriction and interpolation in V-cycle process need iteration, which gradually polishes the error. The iteration which performs before the restriction is named pre-smooth, and which does after the interpolation are called post-smooth.

3.1 Multigrid Poisson PDE Solver

Traditional multigrid Poisson solver acquires the solution of linear equations discretized from Poisson PDE in V-cycle manner. In the left half V-cycle of Figure 2, it solves the approximate solution after the fixed number of iteration, meanwhile restricts the error residual $\mathbf{d}$ of each level to the lower resolution level through operator R. In the right half V-cycle of Figure 2, the algorithm returns the calibration result to the high resolution level through operator P by interpolation. The algorithm of traditional multigrid Poisson solver is as algorithm 1 depiction.

Algorithm 1. Multigrid Poisson PDE Solver

Input: k, N
Output: $\mathbf{U}$

1. Approximately solving $\mathbf{LU} = \mathbf{B}$, and acquiring $\overline{\mathbf{U}}_{old}$.
2. Restriction
   Do $h = N-1, \cdots, 2, 1$
   2a. Approximately solving $\mathbf{L}_h\mathbf{U}_h = \mathbf{B}_h$ after k iterations, producing $\overline{\mathbf{U}}_h$.
   2b. $\mathbf{V}_h \leftarrow \mathbf{U}_h - \overline{\mathbf{U}}_h$, $\mathbf{d}_h \leftarrow \mathbf{L}_h\overline{\mathbf{U}}_h - \mathbf{B}_h$.
   2c. $\mathbf{d}_h = \mathbf{R}\mathbf{d}_{h+1}$
   End Do
3. Solving $\mathbf{L}_0\mathbf{V}_0 = \mathbf{d}_0$
4. Interpolation
   Do $h = 2, \cdots, N-1, N$
   4a. Approximately solving $\mathbf{L}_h\mathbf{V}_h = \mathbf{d}_h$ after k iterations.
   $\mathbf{V}_h = \mathbf{P}\mathbf{V}_{h-1}$
   If $h = 1$, $\overline{\mathbf{U}}_h \leftarrow \overline{\mathbf{U}}_{old} + \mathbf{V}_h$
   End Do

Where k is the number of iterations and N is the number of multigrid levels.

4 Parallel Multigrid Poisson PDE Solver

The conventional multigrid poisson solver separately performs the process of approximation, restriction and interpolation. The data $\mathbf{V}$ and $\mathbf{d}$ at each level in the RAM need to be loaded twice, one is used for restriction, the other is for interpolation. Due to the limited capability of memory, only part of data could be loaded into RAM, the rest of data are gradually loaded into RAM according to the computation requirement. Therefore, for solving Poisson PDE with large-scale unknowns, traditional multigrid Poisson solver bears heavy communication between RAM and external memory, increases the operation time and reduces the computation efficiency.

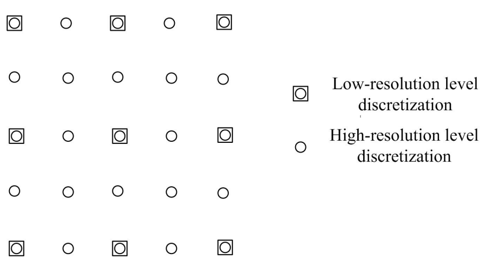

Fig. 3. The discretization point relationship between low-resolution and high-resolution

The figure 3 demonstrates the discretization points relationship between the neighbouring low-resolution and high-resolution level. When solver runs, matrix **L**, **B**, **U** and $\overline{\mathbf{U}}$ load only one time in RAM, the elements in them could be indexed only according to the grid position when they are used in different resolution level. At each level, **V** and **d** are produced and used for transferring the linkage among different resolution level. **d** evaluated in restriction is used for interpolation stage, therefore, **V** and **d** need to be resided in RAM. The total number of elements in **V** and **d** with N level is expressed as Eq. (6), where $\|\cdot\|$ is for counting the number of elements in matrix. For the gigapixel image editing within gradient domain, the number elements in **V** and **d** are 2×10^7, the sum of elements in **L**, **V** and **d** would be 5×10^7 when the multigrid level N is 5.

$$\sum_{h=1}^{N} \|\mathbf{V}_h\| = \sum_{h=1}^{N} \|\mathbf{d}_h\| \approx 10^7 \times (2 - 2^{1-N}) \tag{6}$$

Most of operations in multigrid V-cycle belong to the operation of matrix-matrix multiplication or matrix-vector multiplication which is suitable for parallelized. In this paper, by full exploitation the local accessing coherence of memory data in $\overline{\mathbf{U}}_h$, $\mathbf{V}_h$ and $\mathbf{d}_h$, the current accessed data is constructed a working set W, and then W is shifted along the image column direction for updating need of $\mathbf{V}_h$ and $\mathbf{d}_h$. The shifting of working set is for making use of the data having been loaded in RAM.

The parallelization of parallel multigrid Poisson solver proposed in this paper embodies in two aspect, on one hand, the restriction and interpolation across different resolution level transfer with each other, on the other hand, the elements is maximally shared between the restriction and interpolation. The parallelization among different resolution level shows that, in the iteration, the updating is able to be executed when the required elements be ready. The parallelization between restriction and interpolation presents that, after finishing the restriction, when the required data get ready, interpolation could be performed. The solution could be achieved when the interpolation is accomplished at the highest resolution level.

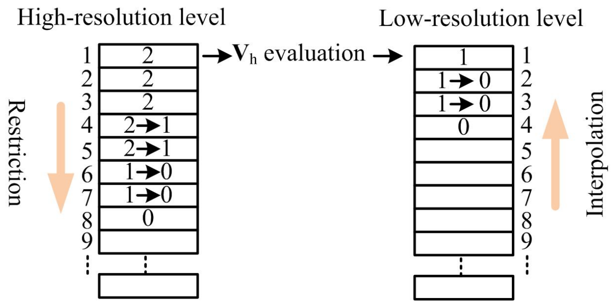

Fig. 4. The parallelization of parallel multigrid Poisson solver

In figure 4, suppose $k = 2$ and $N = 2$, when the 6th and 7th execution cycle perform the restriction of $\mathbf{d}_1 \rightarrow \mathbf{d}_0$, the 4th and 5th execution cycle implement restriction of $\mathbf{d}_2 \rightarrow \mathbf{d}_1$. When the data in the first execution cycle of interpolation is updated, $\mathbf{V}_0$ starts to be calculated and the 2nd and 3rd execution cycle could interpolate $\mathbf{V}_1 \rightarrow \mathbf{V}_0$.

Row of pixels is used as the operation unit for constructing working set W, which consists in the current processing row i, adjacent processed row and to be processed row. The processed row is utilized for updating the current row data, and the introduction of adjacent processed row is to calculate the error residual and calibration after updating the current row data.

Table 1. Data Window

restriction	
$i_{h-1} + 2k + 1 < \lfloor (i_h - 1)/2 \rfloor$	
data window	$[i_h - 3, i_h + 2k + 1]$
restriction from the resolution level of $h-1$	$[i_h + 2k + 1]$
pre-smooth	$[i_h - 1, i_h + 2k + 1]$
error residual	$[i_h - 3, i_h + 1]$
interpolation	
$i_{h+1} + 2k + 1 < 2i_h - 1$	
data window	$[i_h - 1, i_h + 2k + 1]$
interpolation to the resolution level of $h+1$	$[i_h + 2k + 1]$
post-smooth	$[i_h - 1, i_h + 2k + 1]$

Table 1 gives the data windows size in restriction and interpolation. Taking the restriction as an example, set i_h as the present processing line, when the $2k+1$ line of i_h in the $h-1$ level is finished updating, the h level could perform pre-smooth, therefore, the pre-smooth window is $[i_h - 1, i_h + 2k + 1]$. Since the smoothed line could be used for evaluating the error residual, the error residual window is $[i_h - 3, i_h + 1]$. The data window setting in interpolation is similar to the restriction.

The algorithm of parallel multigrid Poisson Solver is as the follow.

Algorithm 2. Parallel multigrid Poisson solver

Input: K, N, L(the number image rows)
Output: $\mathbf{U}$

1. Evaluating the $k-1$ lines of $\overline{\mathbf{U}}_h$ as the pre-processing data..
2. Do $l = K,\ K+1, \cdots,\ L-K$
   - 2a. Evaluating $\overline{\mathbf{U}}_h$ within $[i_l - 3, i_l + 2k + 1]$.
   - 2b. Calculating $\mathbf{d}_h$ within $[i_l - 3, i_l + 1]$.
   - 2c. Acquiring $\mathbf{V}_h$ within $[i_l - 3, i_l + 1]$

   End Do
3. Calculating $\mathbf{V}_N$ on each level of l $(l = K+1,\ \cdots,\ L)$ and then updating $\mathbf{U}$.

The algorithm 2, parallel performs the iteration, restriction and interpolation which make good use of locality and relevance of memory accessing.

5 Experiment

The paper implements the parallel multigrid Poisson solver on dual-core PC computer of TongFang E2180 with 2G RAM. The presented method is used for image composition with a resolution of 1280×720 or 1024×768.

Fig. 5. Image Composition (Data 1)

Fig. 6. Image Composition (Data 2)

The configurations of P and R exploited in the experiment are as the following, which are set by bilinear interpolation.

$$P = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \qquad R = \frac{1}{4}P^T = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

In parallel multigrid Poisson solver, the resolution at low resolution level is 2 times to the neighbouring the high resolution level, that is to say, $H = 2h$, and the highest resolution is the size of source image.

The figure 5 and figure 6 are the composed image within gradient domain by the presented method in the paper. The registration [8] of all images is performed before composition. Figure 5 is a panorama with 8100×3680 which is composed by 5 pieces of image with size 1280×720. Figure 6 is another panorama with 7963×3580 which is composed by 5 pieces of image with size of 1024×768. The RAM usage, I/O Communication, iterations as well as operation time of the figure 5 and figure 6 are listed in the table 2.

Table 2. Comparison of different Poisson solver

Algorithm		RAM usage(M)	I/O Communication(M)	Iterations	Time (s)
Over-relaxation SOR	Data1	56	7.53	676	9.656
	Data2	51	7.3	615	8.433
Jaccobi	Data1	56	7.52	1365	19.499
	Data2	51	7.34	1127	15.451
Multigrid	Data1	224	5.95	232	3.315
	Data2	204	5.53	209	2.866
Parallel Multigrid	Data1	184	3.53	165	2.357
	Data2	180	3.36	160	2.194

In table 2, the parallel multigrid Poisson solver is superior to the traditional algorithm on RAM usage, and is obviously superior to the overrelaxation iteration, Jacobi iteration and traditional multigrid algorithm.

6 Conclusion

Poisson PDE is a commonly used partial differential equation in the simulation and image processing. How to efficiently compile the gigapixel images within gradient domain is the research focus in recent years. Gigapixel image editing in gradient domain needs solving Poisson equation with large-scale unknowns. Traditional multigrid solver is not highly efficient on PC. Therefore, a parallel multigrid Poisson solver for gigapixel image editing within gradient domain is proposed, which exploits the locality and relevance of memory accessing and updating among the stages of iteration, restriction and interpolation for parallel performing. The approach could efficiently accomplish the gigapixel image editing within gradient domain on the PC machine.

Traditional multigrid Poisson solver separately performs iteration, restriction and interpolation which could not make good use of the locality and relevance of memory data. The presented method of parallel multigrid Poisson solver for gigapixel image editing has the higher efficiency, it better utilizes the locality and relevance of memory accessing and updating for parallel performing the iteration, restriction and interpolation.

The proposed approach of parallel multigrid Poisson solver is suitable for Poisson equation with Neumann boundary condition and structured data. Expanding the proposed approach for gigiapixel image editing with complex gradient domain, namely for solving Poisson equation with Dirichlet boundry condition is our future work.

References

1. Pérez, P., Gangnet, M., Blake, A.: Poisson Image Editing. ACM Transactions on Graphics 22(3), 313–318 (2003)
2. Levin, A., Zomet, A., Peleg, S., Weiss, Y.: Seamless Image Stitching in the Gradient Domain. In: Pajdla, T., Matas, J. (eds.) ECCV 2004, Part IV. LNCS, vol. 3024, pp. 377–389. Springer, Heidelberg (2004)
3. Agarwala, A., Dontacheva, M., Agarwala, M., Drucker, S., et al.: Interactive Digital Photomontage. ACM Transaction on Graphics 23(3), 294–302 (2004)
4. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson Matting. ACM Transaction on Graphics 23(3), 315–321 (2004)
5. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C. Cambridge University Press (2002)
6. Kazhdan, M., Hoppe, H.: Streaming Multigrid for Gradient-Domain Operations on Large Images. ACM Transaction on Graphics 27(3), Article No. 21 (2008)
7. Chow, E., Falgout, R.D., Hu, J.J., Tuminaro, R.S., Yang, U.M.: A Survey of Parallelization Techniques for Multigrid Solvers. In: Frontiers of Parallel Processing for Scientific Computing. The Society for Industrial and Applied Mathematics (2006)
8. Szeliski, R.: Image Alignment and Stitching: A Tutorial. Foundations and Trends in Computer Graphics and Computer Vision 2(1), 1–104 (2006)