Advances in Computer Science Research, volume 44
3rd International Conference on Wireless Communication and Sensor Network (WCSN 2016)

Fast Parallel Compositon of Dunhuang Murals based on Matrix Tearing

Ming Chen, Yong-Chao Wang, Lei Zhao, Duan-Qing Xu
Department of computer science and technology, Zhejiang University, Hangzhou, China
E-mail: chenming@163.com, ychaowang@zju.edu.cn, cszhl@zju.edu.cn, xdq@zju.edu.cn

***Abstract*-In this paper we have developed a robust scalable hybrid scheme for Fast Parallel compositon of Dunhuang murals based on Matrix Tearing. The large cost associated with solution of partitioned linear systems, which is required to perform the matrix–vector multiplication, was avoided by presenting a scheme that requires only the solution of tiny linear systems of the same size as that of the overlap between two partitions. The difficulties associated with the multiple partition approach are resolved by using the reordering.**

Keywords-dunhuang mural; comosition; parallel

I. INTRODUCTION

Gradient-domain techniques are widely used in many applications, such as intrinsic image recovery[1], seamless cloning[2], and gradient domain painting[3]. Most of these methodologies share a common issue: solving the Poisson equation. Taking seamless cloning for example, instead of copying absolute values from the source images into composites, gradient-domain methods typically solve the Poisson equation where the gradients inside the cloned regions come from the source images and the Dirichlet boundary conditions are prescribed by the target images. However, solving the Poisson equation is a computational and memory intensive task, which makes common Poisson methods not suitable for real-time image processing. Besides, when dealing with large scale images, such as mega-pixel images and even gigapixels, this problem will be especially obvious. Various methods are proposed to settle this problem. These methods can be divided into two main categories: the solvers based on reduced space and the solvers based on numerical analysis techniques. The approach described in this paper belongs to the second category.

In the first kind, some algorithm descriptions[4,5] transform full resolution images into reduced spaces, which substantially reduce both memory and computational requirements. In the other kind, we could also classify the methods into two groups: direct solvers and iterative solvers. Pardiso[6], MUMPS[7], and SuperLU[8] belong to the direct methods. Conjugate gradient[9], multigrid method[10], and streaming multigrid solver[11] are the commonly used iterative solvers. However, due to direct solvers' poor scalability, they are not appropriate for large linear systems, while iterative solvers are not so robust and they usually take a great number of iterations to converge. In this paper, we propose a robust scalable hybrid scheme for Fast Parallel compositon of Dunhuang murals based on Matrix Tearing. The large cost associated with solution of partitioned linear systems, which is required to perform the matrix–vector multiplication, was avoided by presenting a scheme that requires only the solution of tiny linear systems of the same size as that of the overlap between two partitions. The difficulties associated with the multiple partition approach are resolved by using the reordering. We have also capitalized on the multilevel parallelism inherent in our scheme combining parallelism across nodes and shared memory within each node.

II. RELATED WORK

A. Gradient-Domain Image Processing

Psychologists have pointed that the human visual system is much more sensitive to local contrast than to slow changes in luminance and chrominance[13]. Retinex theory[14]suggests that humans achieve lightness constancy by perceiving scene lightness only through local luminance ratios at edges. In particular, slow luminance changes may be often superimposed over an image without a noticeable effect. Gradientdomain techniques can take advantages of these properties and are widely used in image processing. Image compositing aims at combiningmanysource images seamlessly into an entire image. It was first demonstrated that we can extract gradients from the sources to form a desired gradient field, and then solve for a new image whose pixel differences best match the desired gradients in least squares sense[2].

This operation has attracted considerable research attention in recent years. This basic method was later extended by Ref[4]. An interactive framework was constructed to generate an entire image by compositing regions of many sources into a photomontage. Another improved edition was later proposed[15], which optimized the boundaries of the copied regions when selecting the source patches. The gradient-domain compositing methodwas first used to deal with videos in 2004[16]. Acoordinate-based approach[17] was later introduced which performs seamless cloning as well as a number of other related operations ina directmanner instead of solving the Poissonequation. An interactive framework was constructed to generate an entire image by compositing regions of many sources into a photomontage. Another improved edition was later proposed[15], which optimized the boundaries of the copied regions when selecting the source patches. The gradient-domain compositing methodwas first used to deal with videos in 2004.16Acoordinate-based approach[17] was later introduced which performs seamless cloning as well as a number of other related operations ina directmanner instead of solving the Poissonequation. An error-tolerant gradient-domain compositing method was proposed[18], which is robust to inaccuracies and prevents color bleeding without changing the boundary location. Gradient-domain techniques are not only used for compositing image regions, however. Various applications can be performed based on gradient domain constraints, ranging from shadow removal[19], intrinsic image recovery,1 high dynamic range (HDR) compression[20], to flash artifact correction[21], alpha matting[22], gradient domain painting[23], reproducing photographic look[23], and image relighting[24].

B. *Sparse Matrix Solver*

There are two main classes of solvers for sparse linear systems: direct and iterative methods. Pardiso[6,18], MUMPS[7,19], and SuperLU8 belong to direct methods. Though the direct methods are quite efficient for rmany applications, they might still be quite costly when matrixes are very large. However, direct methods tend to be more robust than iterative ones, and this property makes them more suitable for "black-box" implementations. Commercial software developers appear to avoid implementing iterative solvers whenever possible. Iterative methods mainly consist of classical preconditioned Krylov subspace methods and preconditioned Richardson iterations. Unlike direct solvers, iterative methods (with classical black box preconditioners) are not as robust. This is true even with the most recent advances in creating lower upper (LU)-based preconditioners[20,21]. Approximate inverse preconditioners [22] are known to be more favorable for parallelism. The Spike algorithm[12] is a parallel solver for banded systems that combines direct and iterative methods, and it is one of the first examples of a hybrid linear system solver. In this paper, inspired by the Spike algorithm, we introduce a new parallel hybrid sparse linear system solver that contains both direct and iterative components. We show that our solver can overcome the drawbacks of direct and iterative solvers: it achieves better scalability than with direct solvers and is more robust than with classical preconditioned iterative solvers.

III. ALGORITHM OVERVIEW

In this section, we detail how we accomplish Fast Parallel Compositon of Dunhuang Murals based on Matrix Tearing. We illustrate the main idea of the algorithm using only two overlapped blocks. The generalization to multiple overlapped blocks is straightforward, e.g. see [15]. We are interested in solving linear systems of the form

$$Mx = b(1) \tag{1}$$

where $M \in R^{n*n}$.

Let us assume that A has been reordered, for example using reverse CutHill–Mckee [17] or spectral [19–21] reordering schemes, into the overlapped diagonal blocks below

$$M = \begin{pmatrix} M_{11} & M_{12} & \\ M_{21} & M_{22} & M_{23} \\ & M_{32} & M_{33} \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \text{and } b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{2}$$

where, Mij,Xi and bi for i , j = 1 ,..., 3 are blocks of appropriate size. Let us define the partitions of M, delineated by lines in the illustration in (2), as

$$M_k = \begin{pmatrix} M_{11}^{(k)} & M_{12}^{(k)} \\ M_{21}^{(k)} & M_{22}^{(k)} \end{pmatrix} \text{ for } k = 1,2 \tag{3}$$

where, except for the overlap, $M_{\partial\omega}^{(k)} = M_{k+\partial,k+\omega}$ for ∂,ω= 0,1. The overlap M_{22} consists of the sum of the top left ∂=ω=0 block of the second partition and the bottom right∂=ω=1 block of the first partition. In other words, for this block the following equality holds $M_{22}^{(1)} + M_{11}^{(2)} = M_{22}$. For clarity of presentation, we assume that the partitions are of equal size m and that each overlap is of size τ . Thus, we can rewrite (1) as two linear systems

$$\begin{pmatrix} M_{11}^{(1)} & M_{12}^{(1)} \\ M_{21}^{(1)} & M_{22}^{(1)} \end{pmatrix} \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} = \begin{pmatrix} b_1 \\ \frac{1}{2} b_2 + y \end{pmatrix} \tag{4}$$

$$\begin{pmatrix} A_{11}^{(2)} & A_{12}^{(2)} \\ A_{21}^{(2)} & A_{22}^{(2)} \end{pmatrix} \begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} b_2 - y \\ b_3 \end{pmatrix} \tag{5}$$

where we choose the adjustment vector y such that the solution of the lower part of (4) coincides with the upper part of (5), in other words,

$$x_2^{(1)} = x_1^{(2)} \tag{6}$$

Let us assume, for now, that each overlapped partition is nonsingular, and that

$$M_k^{-1} = \begin{pmatrix} C_{11}^{(k)} & C_{12}^{(k)} \\ C_{21}^{(k)} & C_{22}^{(k)} \end{pmatrix} \tag{7}$$

Thus, using (4), (5) and (6) we obtain the balance system

$$Ny = g \tag{8}$$

$$N = C_{33}^{(1)} + C_{11}^{(2)} \tag{9}$$

$$g = \left(-C_{21}^{(1)}, \frac{1}{2}(C_{11}^{(2)} - C_{22}^{(1)}), C_{12}^{(2)} \right) \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{10}$$

Notice that once the balance system (8) is solved for y, linear systems (4) and (5) can be solved independently in parallel. Since the coefficient matrix M is not available explicitly, we use a modified iterative method to solve the balance system (8), where we compute the residual and matrix–vector products as described below.

$$x_2^{(1)} = B_{21}^{(1)} f_1 + B_{22}^{(1)} (\tfrac{1}{2} f_2 + y) \tag{11}$$

$$x_1^{(2)} = B_{12}^{(2)} f_3 + B_{11}^{(2)} (\tfrac{1}{2} f_2 - y) \quad (12)$$

we obtain

$$r = g - My = x_1^{(2)} - x_2^{(1)} \quad (13)$$

Consequently, considering an initial guess y init to be zero, the initial residual of the iterative solver can be computed as

$$r_{\text{init}} = g = h_1^{(2)} - h_2^{(1)} \quad (14)$$

where

$$\begin{pmatrix} h_1^{(1)} \\ h_2^{(1)} \end{pmatrix} = A_1^{-1} \begin{pmatrix} b_1 \\ \frac{1}{2} b_2 \end{pmatrix} \text{and} \begin{pmatrix} h_1^{(2)} \\ h_2^{(2)} \end{pmatrix} = A_2^{-1} \begin{pmatrix} \frac{1}{2} b_2 \\ b_3 \end{pmatrix} \quad (15)$$

Since

$$Ny = g - r = r_{\text{init}} - r \quad (16)$$

the matrix–vector product can be computed as

$$Np = \bar{y}_2^{(1)} - \bar{y}_1^{(2)} \quad (17)$$

where

$$\begin{pmatrix} \bar{y}_1^{(1)} \\ \bar{y}_2^{(1)} \end{pmatrix} = M_1^{-1} \begin{pmatrix} 0 \\ p \end{pmatrix} \text{and} \begin{pmatrix} \bar{y}_1^{(2)} \\ \bar{y}_2^{(2)} \end{pmatrix} = M_2^{-1} \begin{pmatrix} -p \\ 0 \end{pmatrix} \quad (18)$$

Hence, in our modified iterative method the initial residual and matrix–vector products are computed using (14) and (17),respectively. Generalization to multiple partitions is straightforward and is outlined in [15].

IV. CONCLUSION

In this paper we have developed a robust scalable hybrid scheme for Fast Parallel compositon of Dunhuang murals based on Matrix Tearing. The large cost associated with solution of partitioned linear systems, was avoided by presenting a scheme that requires only the solution of tiny linear systems of the same size as that of the overlap between two partitions. We have also capitalized on the multilevel parallelism inherent in our scheme combining parallelism across nodes and shared memory within each node.

ACKNOWLEDGEMENTS

This work was financially supported by the Zhejiang Province Natural Science Foundation (Y16F020023), Zhejiang Province Project" Key technology research and system development of image restoration based on convolutional neural network", Zhejiang Province Project “Research and development of real time rendering algorithm for super large scale 3D data for mobile terminal”, Zhejiang Province Project “Research on key technologies and system development of modeling and display of multi spectral materials for 3D printing ” and Zhejiang Province Project" Research on the key technology and system development of automatic generation of video virtual viewpoint for security monitoring".

REFERENCES

[1] Y. Weiss, “Deriving intrinsic images from image sequences,” in Computer Vision ICCV 2001, Proc. Eighth IEEE International Conference on IEEE, Vol. 2, pp. 68–75, IEEE Computer Society, Vancouver,Canada (2001).

[2] P. Pérez et al., “Poisson image editing,” ACM Trans. Graphics (TOG) 22(3), 313–318 (2003).

[3] A. Orzan et al., “Diffusion curves: a vector representation for smoothshaded images,” ACM Trans. Graphics (TOG) 27(3) (2008).

[4] A. Agarwala, “Efficient gradient-domain compositing using quadtrees,” ACM Trans. Graphics (TOG) 26(3) (2007).

[5] J. Kopf et al., “Joint bilateral upsampling,” ACM Trans. Graphics (TOG) 26(3) (2007).

[6] O. Schenk and K. Gärtner, “Solving unsymmetric sparse systems of linear equations with PARDISO,” Future Gener. Comput. Syst. 20(3), 475–487 (2004).

[7] P. R. Amestoy et al., “Hybrid scheduling for the parallel solution of linear systems,” Parallel Comput. 32(2), 136–156 (2006).

[8] X. S. Li and J. W. Demmel, “SuperLU_DIST: a scalable distributed-memory sparse direct solver for unsymmetric linear systems,” ACM Trans. Math. Software (TOMS) 29(2), 110–140 (2003).

[9] J. R. Shewchuk, “An introduction to the conjugate gradient method without the agonizing pain,” School of Computer Science Carnegie Mellon University Pittsburgh, http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf (4 August 1994).

[10] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed., PWS publishing company, Boston (2003).

[11] [11]. M. Kazhdan and H. Hoppe, “Streaming multigrid for gradient-domain operations on large images,” ACM Trans. Graphics (TOG) 27(3) (2008).

[12] E. Polizzi and A. H. Sameh, “A parallel hybrid banded system solver: the SPIKE algorithm,” Parallel Comput. 32(2), 177–194 (2006).

[13] S. E. Palmer, Vision Science: Photons to Phenomenology, MIT Press, Cambridge, MA (1999).

[14] E. H. Land, The Retinex Theory of Color Vision, Scientific American, USA (1977).

[15] J. Jia et al., “Drag-and-drop pasting,” ACM Trans. Graph. (TOG) 25(3), 631–637 (2006).

[16] O. Schenk and K. Gärtner, “On fast factorization pivoting methods for sparse symmetric indefinite systems,” Electron. Trans. Num. Anal. 23, 158–179 (2006).

[17] P. R. Amestoy et al., “A fully asynchronous multifrontal solver using distributed dynamic scheduling,” SIAM J. Matrix Anal. Appl. 23(1), 15–41 (2001).

[18] M. Benzi et al., “Orderings for incomplete factorization preconditioning of nonsymmetric problems,” SIAM J. Sci. Comput. 20(5), 1652–1670 (1999).

[19] M. Benzi et al., “Preconditioning highly indefinite and nonsymmetric matrices,” SIAM J. Sci. Comput. 22(4), 1333–1353 (2000).

[20] M. Benzi et al., “A sparse approximate inverse preconditioner for the conjugate gradient method,” SIAM J. Sci. Comput. 17(5), 1135–1149