Computers & Geosciences 78 (2015) 44–52

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

An improved coarse-grained parallel algorithm for computational acceleration of ordinary Kriging interpolation

Hongda Hu, Hong Shu*

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, Hubei, China

ARTICLE INFO

Article history:
Received 5 May 2014
Received in revised form
9 February 2015
Accepted 13 February 2015
Available online 14 February 2015

Keywords:
Interpolation
Ordinary Kriging
Geostatistics
Parallel computing
MPI

ABSTRACT

Heavy computation limits the use of Kriging interpolation methods in many real-time applications, especially with the ever-increasing problem size. Many researchers have realized that parallel processing techniques are critical to fully exploit computational resources and feasibly solve computation-intensive problems like Kriging. Much research has addressed the parallelization of traditional approach to Kriging, but this computation-intensive procedure may not be suitable for high-resolution interpolation of spatial data. On the basis of a more effective serial approach, we propose an improved coarse-grained parallel algorithm to accelerate ordinary Kriging interpolation. In particular, the interpolation task of each unobserved point is considered as a basic parallel unit. To reduce time complexity and memory consumption, the large right hand side matrix in the Kriging linear system is transformed and fixed at only two columns and therefore no longer directly relevant to the number of unobserved points. The MPI (Message Passing Interface) model is employed to implement our parallel programs in a homogeneous distributed memory system. Experimentally, the improved parallel algorithm performs better than the traditional one in spatial interpolation of annual average precipitation in Victoria, Australia. For example, when the number of processors is 24, the improved algorithm keeps speed-up at 20.8 while the speed-up of the traditional algorithm only reaches 9.3. Likewise, the weak scaling efficiency of the improved algorithm is nearly 90% while that of the traditional algorithm almost drops to 40% with 16 processors. Experimental results also demonstrate that the performance of the improved algorithm is enhanced by increasing the problem size.

1. Introduction

In many situations, data collection is time-consuming and expensive so that only sparse and uneven sample points are available. Thus, these sample data cannot meet the rigorous demands of practical applications. Hence, spatial interpolation is one of the most frequently-used solutions for the generation of a continuous surface from a number of discrete observed points. Given its theoretical advantage and high prediction accuracy, Kriging interpolation originating from geostatistics has been widely applied to spatial prediction and decision making in environmental science (Goovaerts, 2000; Hengl et al., 2004; Holdaway, 1996; Sampson et al., 2013; Venkatram, 1988). However, due to its high computational complexity, the attractive aspects of Kriging are often overshadowed by the slow speed of the computation (Morrison, 2000).

Several fast calculation methods have been integrated into the classical computational procedure to improve the efficiency of Kriging interpolation. The existing methods, such as Gaussian Markov random field, fast matrix–vector products, and Fast Fourier Transformation(FFT), accelerate computation and relieve memory pressure (Fritz et al., 2009; Hartman and Hössjer, 2008; Memarsadeghi et al., 2008; Nowak and Litvinenko, 2013). Moreover, Kriging concerned with full rank covariance matrix inversion makes heavy computation demand when applied to large spatial datasets. Therefore, low-rank covariance approximations have been investigated, which dramatically reduce the computational complexity of Kriging (Cressie and Johannesson, 2008; Furrer et al., 2006; Katzfuss and Cressie, 2011; Kaufman et al., 2008; Zhang and Du, 2008). However, these methods focus on the improvement of serial Kriging algorithms not parallel Kriging algorithms, resulting in the underuse of computational resources.

With the rapid development of multi-core CPU and GPU hardware architecture, parallel computing technology has made remarkable progress. Starting in the 1990s many researchers have

* Corresponding author. Fax: +86 27 68778525.
E-mail address: shu_hong@whu.edu.cn (H. Shu).

http://dx.doi.org/10.1016/j.cageo.2015.02.011
0098-3004/© 2015 Elsevier Ltd. All rights reserved.

devoted themselves to the parallelization of various interpolation algorithms (Armstrong and Marciano, 1996, 1997; Guan and Wu, 2010; Huang et al., 2011; Wang and Armstrong, 2003). Specially, many Kriging parallel programs were implemented on high performance and distributed architectures (Cheng et al., 2010; Gajraj et al., 1997; Gebhardt, 2003; Guan et al., 2011; Kerry and Hawick, 1998; Morrison, 2000). Pesquer et al. (2011) leveraged MPI libraries to exploit a parallel ordinary Kriging interpolation algorithm incorporating automatic variogram fitting. They employed a dynamic load-balancing technique to allocate interpolation tasks so that the parallel Kriging solution is not specific to a particular environment or architecture. Nevertheless, dynamic distribution may incur additional overhead as relocation of tasks takes place, especially in the case of one task packet of a few unobserved points. The memory resources in the parallel environment directly restrict task packet size, and further impair the performance of the parallel algorithm. Similarly, GPU memory may be insufficient to store the right hand side (RHS) matrix of the Kriging linear equations in the implementation by Cheng (2013), since many small data transfers were batched into a single large data transfer from host to device for reducing the frequent memory transfers overhead at the cost of additional storage space. The high memory consumption of these previous parallel algorithms based on the traditional approach mainly results from the store of a large linear system while processing large datasets (Shi and Ye, 2013). An improved approach proposed by Davis and Grivet (1984) made an algebraic reformulation to relate the scale of the Kriging linear system to the size of observed dataset only, thus reducing memory consumption. Moreover, the time-consuming matrix multiplications in the traditional approach are avoided because the Kriging estimator is computed in the form of scalar products. However, there is no previously existing parallel version of this approach, we address with an MPI-based parallelization in a distributed memory system to fill this gap in the research.

Comparatively, two MPI-based Kriging algorithms are designed based on the traditional and improved approaches, focusing on parallelizing the most computation-intensive parts for solving the linear system and estimating each unknown point. All the points to be estimated are split into a few blocks, and packaged and assigned to each processor at once. The data interpolation of each unknown point can be regarded as an embarrassingly parallel problem that is obviously decomposable into many identical and separate subtasks (Pacheco, 2011).

Through a series of comparisons, our novel parallel algorithm achieves better performance in both execution efficiency and memory consumption. In addition, our experiment statistically demonstrates the relationship between the efficiency of the improved algorithm and the problem size.

The rest of this paper is divided into three sections. Section 2 first gives a review of two kinds of serial approaches to ordinary Kriging, and then introduces the implementation details of parallel algorithms. Section 3 presents the experimental results and makes pairs of comparisons. The contributions of our work are summarized in Section 4.

2. Methodology and implementation

2.1. The traditional serial approach to ordinary Kriging

In past decades, a large number of Kriging variants have been developed. This study focuses on ordinary Kriging (OK) with an intrinsic stationary regionalized random variable. The unobserved points are predicted as the weighted linear average of observed points, as shown by Eq. (1):

$$\hat{Z}_0 = \sum_{i=1}^{n} \lambda_i Z_i \tag{1}$$

where $\hat{Z}_0$ represents the estimation of one unobserved point, Z_i means the attribute value of one observed point i, and λ_i is the weight coefficient of Z_i for calculating $\hat{Z}_0$.

In theory, OK provides the best linear unbiased prediction at unobserved locations. That is to say, the estimator's expected value is equal to the true value, and the prediction error variance is minimized (Cressie and Cassie, 1993). In order to ensure unbiased estimation, the sum of the weight coefficients of all observed points for the certain unobserved point must be equal to 1. Thus a system of linear equations is built as in the following matrix form (Eq. (2)):

$$\begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma_{n1} & \cdots & \gamma_{nn} & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \mu \end{bmatrix} = \begin{bmatrix} \gamma_{01} \\ \vdots \\ \gamma_{0n} \\ 1 \end{bmatrix} \tag{2}$$

where γ_{ij} stands for the variogram value between point i and point j, and μ is a Lagrange multiplier.

To solve Eq. (2), many parallel algorithms first calculate the inverse of the coefficient matrix, and then calculate the weight vector by matrix–vector multiplication (Cheng, 2013; Pesquer et al., 2011). In general, many efforts have been made towards the parallelization of matrix operations (e.g., matrix sum, matrix multiplication and matrix inverse) (Gutiérrez de Ravé et al., 2014). In our computation, before solving the linear system, the coefficient matrix is decomposed into two matrices (a lower triangular matrix L and an upper triangular matrix U) by the LU decomposition method. Next, the decomposed system of linear equations is solved directly by forward and backward substitution without an explicit inverse matrix. This is a more efficient and more accurate solution (Intel, 2013).

A classical algorithm of ordinary Kriging interpolation consists of five steps:

1) Calculating the empirical spatial variogram values at different lags to fit the theoretical variogram models (e.g., the spherical model);
2) Calculating the variogram coefficient matrix of sample data (observed points) and decomposing this matrix by the LU decomposition method;
3) Calculating the variogram vector between one unobserved point and all observed points;
4) Solving the decomposed system of linear equations to obtain the weight vector, and then calculating the estimation of an unobserved point;
5) Repeating steps 3 and 4 to work out the next unobserved point until all unobserved points are estimated.

2.2. The improved serial approach to ordinary Kriging

Through the analysis of the steps described in Section 2.1, given M and N stand for the sizes of unobserved and observed datasets respectively, the time complexity of step 2 for the LU decomposition is $O(N^3)$ while the time complexity of step 4 for solving the decomposed system of linear equations reaches $O(M^*N^2)$. Frequently in practice, only a few sparse sample points are available to estimate many dense unknown points in a gridded space for the generation of a smooth continuous prediction surface, and this will result in that the time complexity of step 4 substantially exceeds $O(N^3)$. It also turns out that the computation of weights