Parallel Generation of Very High Resolution Digital Elevation Models: High-Performance Computing for Big Spatial Data Analysis

Minrui Zheng, Wenwu Tang, Yu Lan, Xiang Zhao, Meijuan Jia, Craig Allan and Carl Trettin

Abstract Very high resolution digital elevation models (DEM) provide the opportunity to represent the micro-level detail of topographic surfaces, thus increasing the accuracy of the applications that are depending on the topographic data. The analyses of micro-level topographic surfaces are particularly important for a series of geospatially related engineering applications. However, the generation of very high resolution DEM using, for example, LiDAR data is often extremely computationally demanding because of the large volume of data involved. Thus, we use a high-performance and parallel computing approach to resolve this big data-related computational challenge facing the generation of very high resolution DEMs from LiDAR data. This parallel computing approach allows us to generate a fine-resolution DEM from LiDAR data efficiently. We applied this parallel computing approach to derive the DEM in our study area, a bottomland hardwood wetland located in the USDA Forest Service Santee Experimental Forest. Our study demonstrated the feasibility and acceleration performance of the parallel interpolation approach for tackling the big data challenge associated with the generation of very high resolution DEM.

Keywords High-performance and parallel computing · Spatial domain decomposition · Very high resolution DEM · LiDAR data

M. Zheng · W. Tang (✉) · Y. Lan · M. Jia · C. Allan
Department of Geography and Earth Sciences, University of North Carolina at Charlotte, 28223 Charlotte, USA
e-mail: wtang4@uncc.edu

M. Zheng · W. Tang · Y. Lan · M. Jia
Center for Applied GIScience, University of North Carolina at Charlotte, 28223 Charlotte, USA

C. Trettin
U.S. Forest Service, Center for Forested Wetlands Research, 29434 Cordesville, USA

X. Zhao
School of Resources and Environmental Science, Wuhan University, Wuhan, China

S. S. Roy et al. (eds.), *Big Data in Engineering Applications*,
Studies in Big Data 44, https://doi.org/10.1007/978-981-10-8476-8_2

1 Introduction

Digital elevation models (DEM) allow for representing the topographic surface of the Earth by providing spatial location and the elevation information over a geospatial area [26]. As a common data source for topographic analysis, DEM data can be produced from a series of technologies, exemplified by Light Detection and Ranging (LiDAR) technologies. Over the past few years, LiDAR data that provides details of geographic features have been increasingly collected to generate DEMs for the delineation and analysis of topographic surfaces, which are essential in a suite of science and engineering domains, such as hydrologic engineering [8, 40], geographic information science and surveying engineering [27, 38], and environmental engineering [13, 18, 41]. Since 1990s, a series of studies have been reported in terms of using LiDAR-derived DEM to support, for example, microtopography analysis [4, 10, 16, 19, 20], plant species distribution [21], and landslide detection [23].

The generation of very high resolution DEMs from LiDAR is well established [6]. However, the computational demand of generating a DEM at very high resolutions (e.g., 0.5, 0.05 m, or even 0.01 m) from LiDAR data is often problematic. The generation of high or very high resolution DEM requires longer computing times together with large storage space requirements as compared to low resolution applications. In other words, the generation of very high resolution DEM is usually accompanied with a big data issue [31, 42] because the volume of the data increases exponentially as spatial resolution becomes finer. To resolve this big data issue facing the generation of very high resolution DEM, high-performance and parallel computing (HPC) represents one possible solution [11, 17, 22].

HPC employs multiple processors (e.g., CPUs) instead of a single one for accelerating a computational problem of interest [39]. Typically, multiple processors used by HPC form into a computing cluster, each of which includes a head node (or master node) and multiple computing nodes connected through network switch [39]. The basic algorithm of parallel computing is to split the entire task of the computational problem into sub-tasks, and then deploy these sub-tasks to multiple processors on the computing cluster for concurrent computation. Once all sub-tasks are completed, the computing nodes return the results to the head node for aggregation. HPC have witnessed an increasing number of applications in scientific fields, such as bioinformatics, molecular dynamics and environmental applications [24, 28, 32, 33, 37].

Thus, to tackle the big data issue, in this study we will generate very high resolution LiDAR-derived DEM using a HPC approach. Our results demonstrate that the HPC is an efficient and effective approach for developing very high resolution DEM that provide representations of topographic features. The parallel computing approach to accelerate the generation of the very high resolution DEM

via spatial interpolation is applicable to the widely available LiDAR data thereby expanding the potential application of this high resolution data. For this study we utilized a landscape that is representative of the lower coastal plain in the Southeastern U.S. where small difference in topographic features may have significant ramifications to a wide array of considerations from water management to ecological processes.

2 Study Area and Data

Our study area is the USDA (United States Department of Agriculture) Forest Service Santee Experimental Forest (https://www.srs.fs.usda.gov/charleston/santee). The USDA Forest Service Santee Experimental Forest (SEF) was established in 1937 with the Francis Marion National Forest, South Carolina, with a total area of 2,468 ha (latitudes are from 33.12165° to 33.192979°, and longitude from −79.752968° to −79.839113°; see Fig. 1). The purpose of the SEF is to provide a basis for experiments, demonstration trials and long-term monitoring of a variety of field-scale ecological, hydrological, and climatic properties. The forest is representative of the lower coastal plain landscape which is being rapidly developed. It is characterized by very low relief, and gauged watersheds on the SEF are used for monitoring hydrologic responses within 1st, 2nd and 3rd order watersheds that are connected to the East Branch of the Cooper River that flows into Charleston estuary and subsequently the Atlantic Ocean. The study area is characterized by mixed pine-hardwood forests in the uplands and bottomland hardwood forested wetlands. The wetlands are influenced by freshwater tidal cycles and non-tidal systems. An important characteristic of the upland and wetland forests is the spatially distributed microtopographic features (hummocks and hollows), which are impacted differentially by fluctuating water levels in this low relief landscape. The microtopography influences they hydrologic storage properties of a watershed [1] as well as biogeochemical processes, especially related to the carbon cycle in forested wetlands [2, 35].

This LiDAR contains 20 tiles covering our study area (in total 31,561,291 points). The averaged point density is 1 point/m^2 of the LiDAR data managed in point cloud form (see Table 1). Other geospatial data including road networks, streams, and the boundary of the SEF were also available.

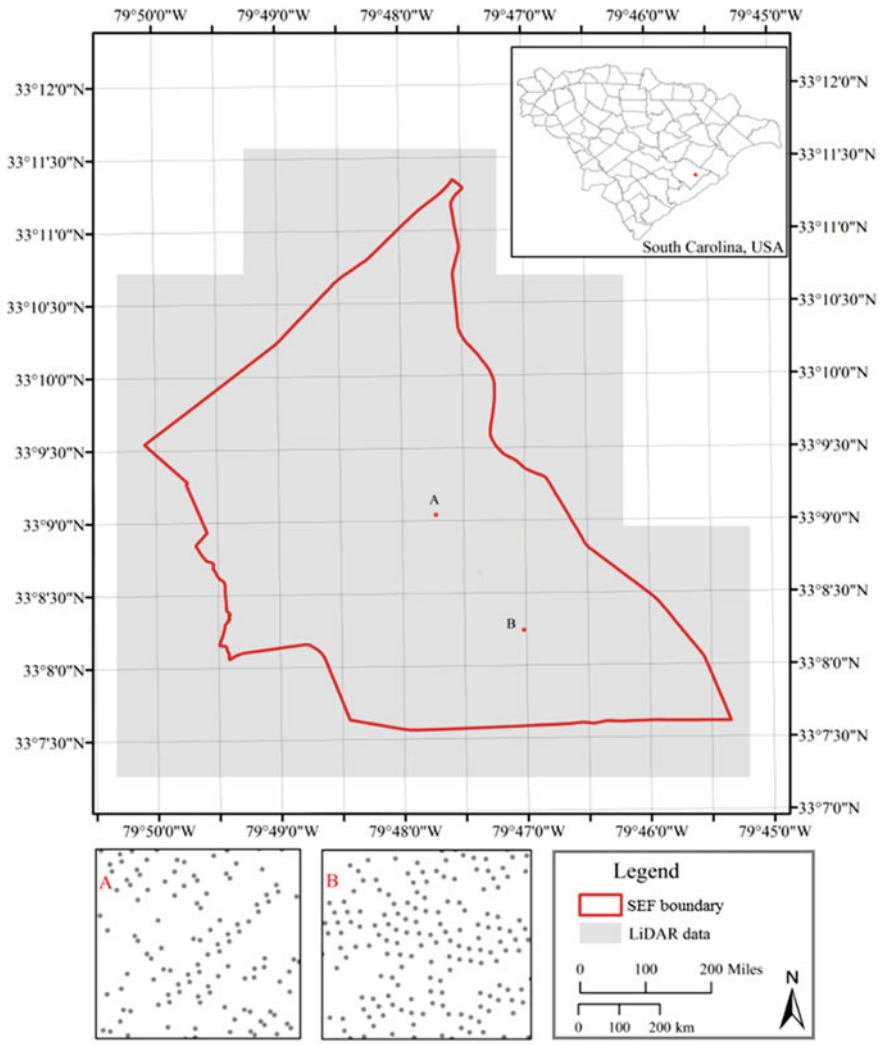

Fig. 1 Study area: santee experiment forest (USDA forest service) within the lower coastal plain, South Carolina

Table 1 Summary of LiDAR data of the study area

Published year	2007
Total tiles of dataset	20
Total size of dataset	3.5G
Geometry type	Point cloud
Unit	Meter
Point density	1 point/m^2

3 Methodology

In this study, we designed a parallel spatial interpolation approach to generate a very high resolution DEM from the LiDAR data in our study region. We performed a spatial domain decomposition on the LiDAR data. Based on the decomposition results, we developed a parallel computing approach for accelerating the generation of high-resolution DEM by applying spatial interpolation of the LiDAR points.

3.1 Spatial Interpolation of LiDAR Data

Spatial interpolation is an approach that predicts the value of an unknown region (point here) based on a number of its surrounding points which values are known. Spatial interpolation can be applied for generating a continuous surface of any geographic variables (e.g., elevation, rainfall, temperature) from sampled locations (as control points). Alternative spatial interpolation algorithms exist, including inverse distance weighted (IDW), Kriging and Spline [25]. All interpolation algorithm can be classified into two basic types: global and local interpolation. The major difference between global and local interpolation is the scope of data used for estimating values of points of interest. Global interpolation uses the entire dataset to estimate points with an unknown value. Local interpolation only considers the points located in a neighborhood distance from the point of interest [36]. IDW is a form of local interpolation algorithms. For massive spatial data, local interpolation has significant advantages because local interpolation can be partitioned into sub-domains based on the location of the neighborhood.

In this study, we focus on using the IDW interpolation method for the generation of very high resolution DEM from LiDAR data. As Zimmerman et al. [43] illustrated, IDW predicts the values of unknown locations using a weighted average of points with known values within a certain distance or a given number of nearest points (e.g., 10–30). The weight is inversely proportional to distance between points. The formula for IDW is as follows:

$$r = \frac{\sum_{i=1}^{m} d_i^{-p} v_i}{\sum_{i=1}^{m} d_i^{-p}} \tag{1}$$

where r is the value of a point to be estimated. v_i is the value of a sampling point. m is the number of nearest neighbors. p is the coefficient of the power function and d_i is the distance from m known nearest sampling points to the estimated point r. The power p controls the influence of neighboring points on determining the estimated value of the unknown location of interest.

Cross validation is often needed to select optimal parameters of spatial interpolation from a number of candidates. In this study, we use a Jackknife method (also known as leave-one-out approach) for cross validation. Jackknife is based on

removing one sample point of the dataset at a time, and repeatedly estimating value using the remaining points in the dataset [34]. The cross validation performance can be evaluated by the root-mean-square error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(V_{it} - V_i)^2}{n}} \tag{2}$$

where V_{it} is the interpolated value of sample i using remaining $n - 1$ records, V_i is the observed value of sample i, and n is the number of samples in the dataset of interest.

3.2 Parallel Interpolation for the Generation of DEM

The generation of DEM data from massive LiDAR data faces computational challenges. In this study, we developed a parallel spatial interpolation approach for the generation of DEM based on LiDAR data. The past two decades have witnessed a variety of studies on the use of parallel spatial interpolation to solve the computational challenge facing massive spatial data. Armstrong and Mariciano [3] used an IDW interpolation method with a MIMD (multiple instruction, multiple data) parallel processing environment. A few years later, Cramer and Armstrong [5] evaluated static and dynamic domain decomposition strategies for parallel interpolation on using IDW algorithm. Guan and Wu [11] investigated the power of multicore-based parallel computing platforms for generating DEM from massive LiDAR data, in which the IDW interpolation algorithm was used. Huang and Yang [17] proposed a grid computing solution based on the Condor platform (aka, Condor, see https://research.cs.wisc.edu/htcondor/) for the spatial interpolation of DEM using IDW from a large spatial dataset. Li et al. [22] developed a general framework for parallel processing of large-scale LiDAR data, and used DEM generation for Colleton County in South Carolina as a case study to demonstrate the utility of the framework implemented based on a map-reduce mechanism.

There are typically four steps to design a parallel computing algorithm, including partitioning, communication, agglomeration and mapping [9, 29]. With these four steps, Fig. 2 shows the framework of our parallel computing approach for the generation of DEM based on spatial interpolation.

Partitioning is the first step of the parallel spatial interpolation for the generation of DEM. In this study, we used spatial domain decomposition for the partitioning of a large spatial interpolation problem into sub-problems for acceleration. Spatial domain decomposition is one of the spatial strategies that is popular in parallel spatial modelling particularly with big spatial data. The decomposition strategy divides a dataset into several subtasks based on different task requirements, and then schedule these tasks on multiple processors (i.e., computing node). All computing nodes' results are returned and aggregated on the head node. Over the past

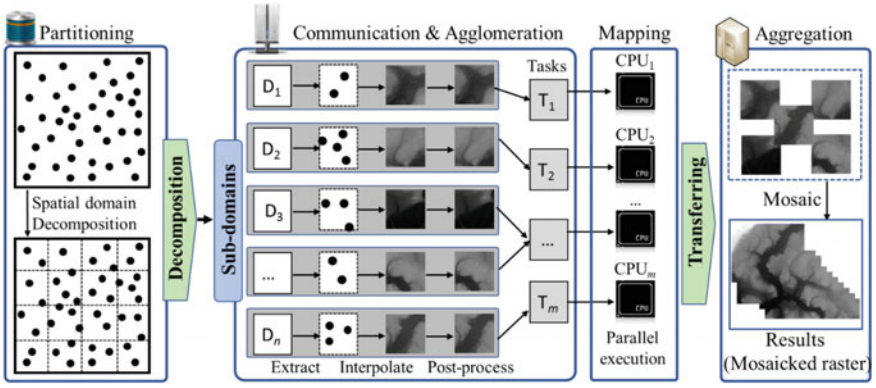

Fig. 2 Framework of the parallel computing approach of spatial interpolation for the generation of very high resolution DEM

several years, domain decomposition is already proved its benefits in accelerating spatial modelling tasks when using parallel computing, and a number of publications on parallel processing based on spatial domain decomposition have been reported. For example, Ding and Densham [7] stressed that one- or two-dimensional decomposition can be used in many types of data: single data type (i.e., binary or categorical) with regular (e.g., square, rectangular, triangle) or irregular shape (e.g., LiDAR data in this study), mixed data types (i.e., binary and categorical) with regular or irregular shape. Because two-dimensional decomposition needs less communication, they also pointed out two-dimensional decomposition is more efficient than one-dimensional decomposition in regular shape data. Besides the one- or two-dimensional decomposition method, there exists other decomposition methods, such as quadtree domain decomposition [36]. We chose to use two-dimensional regular spatial domain decomposition strategies in our study for the parallel spatial interpolation for the generation of DEM data based on LiDAR data. The spatial domain covering our study area is split into a matrix of subdomains in rectangular shapes (see Figs. 2 and 3). Originally, these rectangular sub-domains are non-overlapping.

Handing of communication among tasks associated with subdomains is the second step of parallel computing algorithms. Overlapping spatial domain decomposition is often associated with spatial analysis algorithms in need of information from neighborhood scope [5, 7]. Usually, a non-overlapping domain decomposition is the most efficient way for the parallelization of spatial analysis algorithms. However, non-overlapping decomposition may lead to incorrect results when spatial analysis algorithms depend on neighborhood information, such as IDW algorithm in this study. Therefore, overlapping spatial domain decomposition is often preferred for parallelizing spatial analysis algorithms with neighborhood scope rather than the non-overlapping solution [7]. The overlapping regions of subdomain depends on the neighborhood scope of the focal geospatial features for

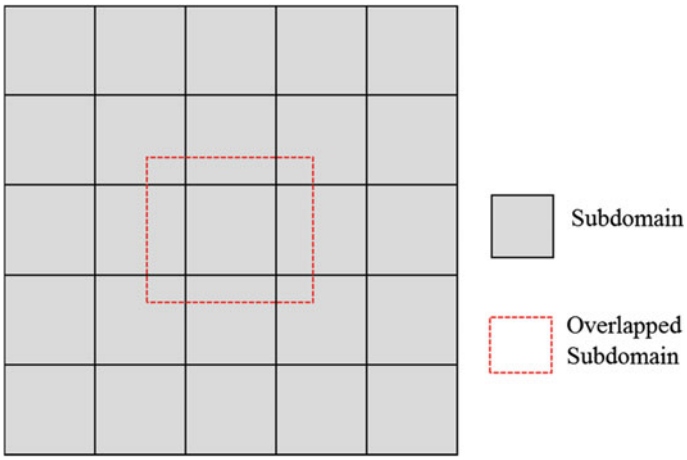

Fig. 3 Illustration on the spatial domain decomposition of parallel spatial interpolation

the spatial analysis algorithms. For example, when using IDW algorithm, the radius of overlapping subdomains is defined by the longest distance from neighboring points to the focal point of interest. A series of meaningful extensions and applications of overlapping domain decomposition have been conducted through the years. For example, Shepard [30] involved overlapping subdomains for parallelizing nearest neighbor search operations. Hohl et al. [14, 15] implemented overlapped subdomains to include neighborhood points within a threshold distance (bandwidth here) to correct edge effects for parallel kernel density estimation.

IDW algorithm used in spatial interpolation in this study relies on the scope of neighborhood because a number of neighboring points are used to estimate the value of a focal point of interest. If the original subdomains that are non-overlapping are directly used, the communication among computing nodes for those neighboring points that may be located in different computing nodes needs to be addressed (e.g., using a message-passing mechanism; see [39]). In this study, instead of message-passing parallelism, we chose to use an alternative approach to address this situation: each subdomain is extended based on a buffer analysis operation (see Fig. 3). As a result, a series of overlapping subdomains are generated that can be used to extract the sub-datasets for spatial interpolation. The radius of the buffer analysis should be larger than the longest distance from neighboring points to the focal point of interest. Thus, it is not necessary to directly handle communication among computing nodes for the parallel spatial interpolation on multiple processors. In other words, there is no overhead for inter-processor communication.

The computing performance of our parallel approach is evaluated using speedup and efficiency. Speedup and efficiency are the two indexes to evaluate the computing performance of a HPC [39]. Speedup (s) is a measure of relative

performance between execution time using parallel computing (t_p) with n CPUs and execution time using sequential computing (t_s; one CPU here), defined as:

$$s = \frac{t_s}{t_p} \tag{3}$$

The theoretically maximum speedup is n with n CPUs (say, linear speedup). Efficiency (e) is a standardized metric that is the ratio of speed up over the number of CPUs used for the computation:

$$e = \frac{s}{n} \tag{4}$$

Studies in the literature demonstrated parallel implementation of DEM interpolation can substantially accelerate the overall computation. For example, Guan and Wu [11] used a parallel solution with pipelining algorithm in their spatial interpolation for DEM generation. The computing time of their algorithm was reduced from 50 to 12 min, with a speedup of 4.2 based on 8 processors. In Huang and Yang [17]'s study, the best speedup for interpolating DEM interpolation using Condor is 16, on the basis of 20 CPUs. Li et al. [22] applied their general-purpose computing framework for parallel processing of DEM interpolation on a Hadoop cluster, and the speedup with 10 computing nodes reached 5.38.

3.3 Implementation

The generation of very high resolution LiDAR-derived DEM in this study uses a set of software packages. We used ESRI ArcMap (version 10.3; http://desktop.arcgis.com/en/arcmap/) for IDW spatial interpolation and the mosaic function for the aggregation of all results from different subdomains into a single dataset. Python script is used to implement spatial domain decomposition. We developed a Python script that combines a series of ArcGIS functionalities to automate our parallel domain decomposition and spatial interpolation process.

4 Experiment and Results

4.1 Setting up of High Performance Computing Environments

The high-performance computing resources used for this study are a Windows-based cluster (Sapphire at the Center for Applied Geographic Information Science, University of North Carolina at Charlotte). The Windows-based cluster consists of

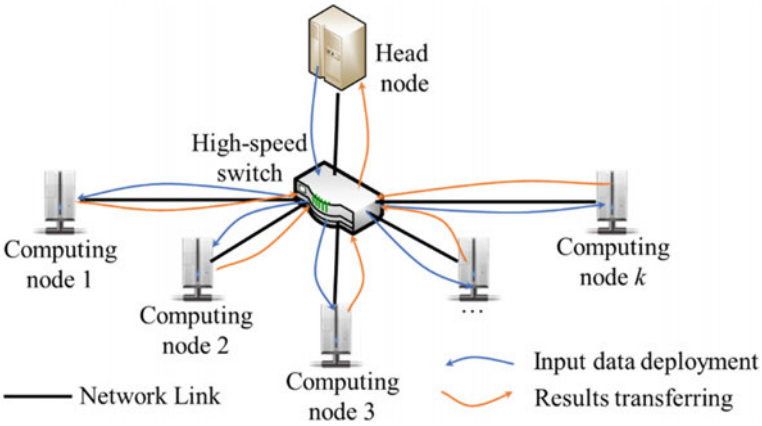

Fig. 4 Illustration on the architecture of a windows-based cluster

20 computing nodes, each of which has 2 CPUs (Intel Core 2 Duo CPU with 3.00 GHz) and 4 GB of memory. The computing nodes are connected through a gigabit band network switch. ArcGIS 10.3 and Python are installed on the head and computing nodes of the cluster (see Fig. 4). We use Microsoft HPC Cluster Manager [12] for job scheduling.

4.2 Result of Very High Resolution LiDAR-Derived DEM

We designed an experiment including five treatments to investigate the impact of granularity level on parallel computing performance. The spatial domain decomposition strategies for the five treatments are 10 × 10, 15 × 15, 20 × 20, 25 × 25, and 30 × 30. As a result, the number of decomposed tiles with LiDAR data is 100, 225, 400, 625, and 900 subdomains for these five treatments (see Fig. 5). Each decomposed tile corresponds to a sub-task that can be further aggregated into a task deployed on a processor.

The distance of the buffer used for creating overlapping regions for each subdomain was set to 1 m. The parameter of power coefficient, p (Eq. 1) for IDW-based spatial interpolation is 3.34 with 0.06 RMSE (Eq. 2) for leave-one-out cross validation. Based on the 0.05 m spatial resolution, the number of rows and columns of spatial interpolation results (as in raster data) is 148,217 × 140,105. All the treatments are computed on the Window cluster discussed in Sect. 4.1.

Figure 6 depicts the microtopographic detail of this low relief bottomland wetland environment derived from the high resolution DEM developed from our computing procedure. Such data are required to apply spatially distributed hydrologic and biogeochemical models in this and similar wetland environments.

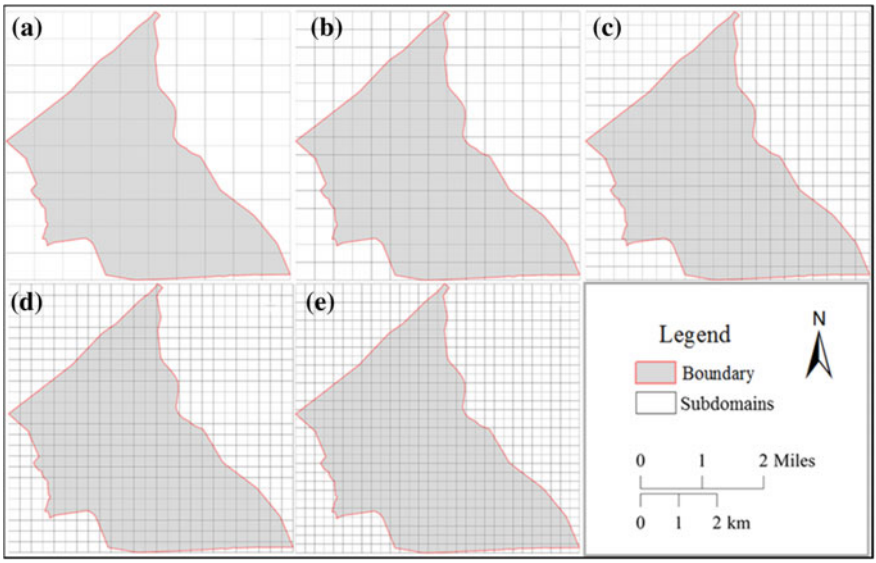

Fig. 5 Map of spatial domain decompositions with different granularities (**a** 10 × 10 subdomains, **b** 15 × 15 subdomains, **c** 20 × 20 subdomains, **d** 25 × 25 subdomains, **e** 30 × 30 subdomains)

4.3 Computing Performance

Table 2 summarized the computing time of these five treatments in response to the number of CPUs used for parallel acceleration. The number of CPUs increases from 2 to 28 with an increment of 2. Usually, the sequential time is computed using the entire dataset on a single CPU. However, the spatial interpolation cannot be directly applied to the entire dataset because the size of the matrix (148,217 × 140,105) used to host the entire spatial interpolation results are too large (which consumes huge amounts of computer memory). Thus, in this study, we used the summation of computing time for each sub-task as an alternative of sequential computing time. The sequential time of the five treatments are 74,866.51 s (about 20.8 h), 70,853.72 s (about 19.7 h), 64,543.72 s (about 17.9 h), 64,343.94 s (about 17.9 h), and 66,869.5 s (about 18.6 h), respectively. As we could observe from Table 2, the execution time for each of the five treatment decreases as more CPUs are added. From Fig. 7, we could see that the execution time rapidly declines when 4 CPUs are used. When the range of used CPUs from 4 to 18, computing time shows a slowly decreasing trend. After 18 CPUs are involved, the parallel computing time is relatively constant (most runs are within 1 h). In particular, when the number of CPUs used for parallel spatial interpolation is 28, the lowest range of computing time is from 3,734 to 2,512 s with increasing the number of decompositions.

From Fig. 7 and Table 2, we see that with the use of 25 × 25 or 30 × 30 decompositions results in limited reductions in parallel computing time. Instead,

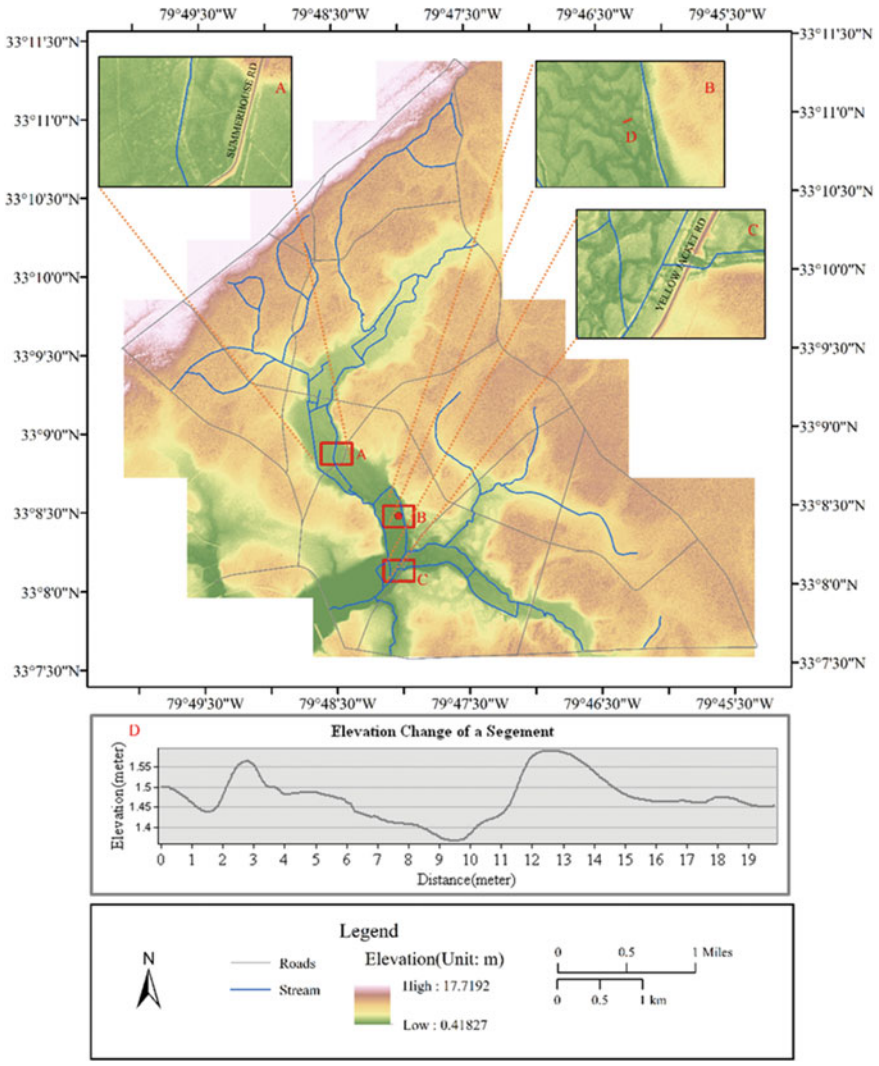

Fig. 6 Map of very high resolution LiDAR-derived DEM (spatial resolution: 0.05 m; landscape size: 148,217 × 140,105)

very fine partitioning will likely lead to increase in computing time because the time spent on spatial domain decomposition time is rapidly increasing. Figure 8 shows computing time for each step of the parallel spatial interpolation (including decomposition, spatial interpolation, and post-processing) over five treatments. Figure 8a illustrates the scenario of sequential computing time and Fig. 8b parallel computing time using 28 CPUs. We observe that spatial interpolation dominates the entire process both for sequential and parallel computing. The computing time spent

Table 2 Computing performance of the experiment of five decomposition granularity levels over number of CPUs (unit: seconds)

CPU	10 × 10	15 × 15	20 × 20	25 × 25	30 × 30
2	37,560.69	35,984.50	32,394.33	32,247.05	33,449.69
4	19,422.50	18,258.56	16,298.61	16,225.25	16,756.99
6	12,981.45	12,129.59	10,892.55	10,894.40	11,240.28
8	10,257.82	9,248.81	8,492.82	8,164.69	8,465.50
10	8,528.61	7,437.02	6,813.05	6,732.07	6,802.18
12	7,250.45	6,351.11	5,605.52	5,512.69	5,705.71
14	6,285.16	5,693.98	4,913.88	4,733.35	4,871.34
16	5,158.56	5,084.68	4,913.88	4,194.72	4,306.56
18	5,329.11	4,494.84	3,893.03	3,863.67	3,784.10
20	4,592.07	3,819.58	3,525.73	3,463.27	3,486.67
22	4,083.29	3,816.80	3,209.88	3,064.32	3,115.70
24	3,950.40	3,360.74	2,961.28	2,869.90	2,937.03
26	3,889.81	3,255.98	2,735.48	2,679.55	2,690.08
28	3,733.85	2,982.76	2,644.97	2,492.91	2,511.66
Sequential	74,866.51	70,853.72	64,543.72	64,343.94	66,869.50

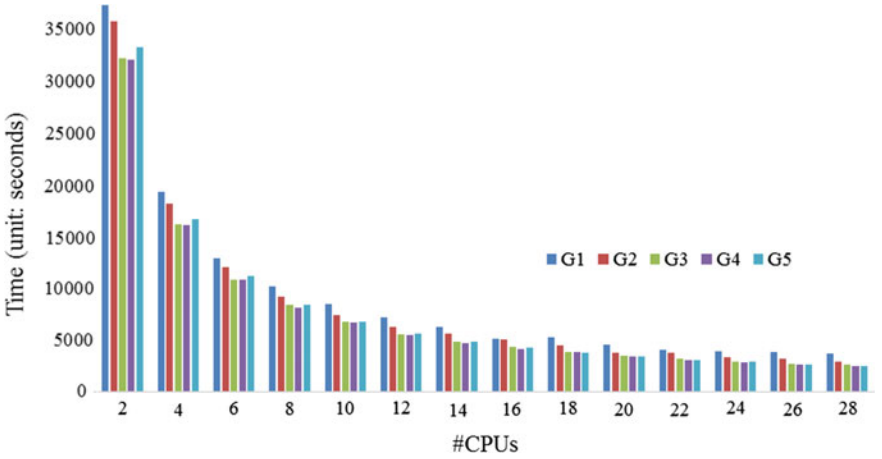

Fig. 7 Parallel computing time of the experiment of spatial domain decomposition granularities over different numbers of CPUs (G1: 10 × 10 subdomains; G2: 15 × 15 subdomains; G3: 20 × 20 subdomains; G4: 25 × 25 subdomains; G5: 30 × 30 subdomains)

on post-processing generally tends to be the shortest among the three steps. For sequential computing, the spatial interpolation and post-processing times decrease slightly when spatial domain decomposition becomes finer, but decomposition time tends to increase. Further, the total execution time tends to decrease since computing time spent on spatial interpolation the step that dominates the entire

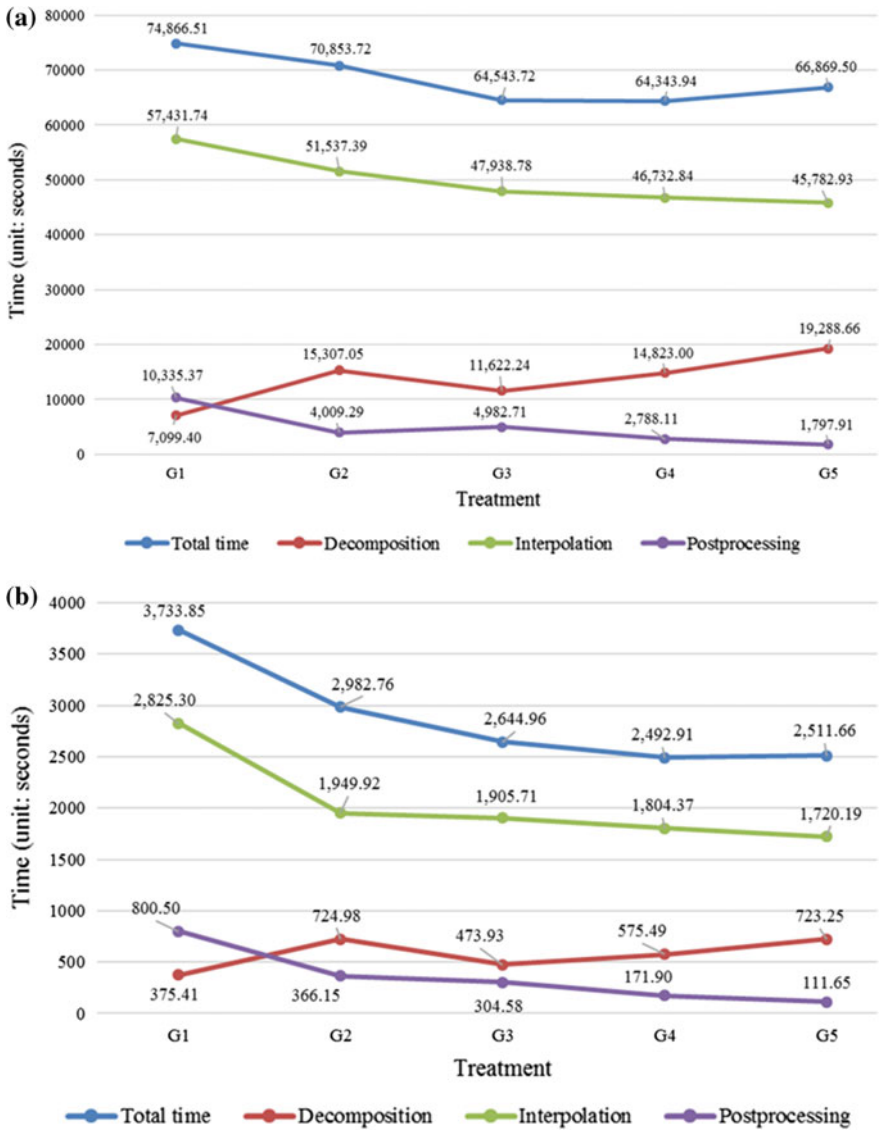

Fig. 8 Computing time for each step of parallel spatial interpolation over five treatments (**a** sequential computing time, **b** parallel computing time; time unit: seconds; #CPUs: 28; G1: 10 × 10 subdomains, G2: 15 × 15 subdomains; G3: 20 × 20 subdomains; G4: 25 × 25 subdomains; G5: 30 × 30 subdomains)

computing time exhibits a decreasing pattern in response to finer decomposition granularity. Likewise, the parallel computing time (Fig. 8b) shows similar patterns in response to increase in the granularity of spatial domain decomposition,

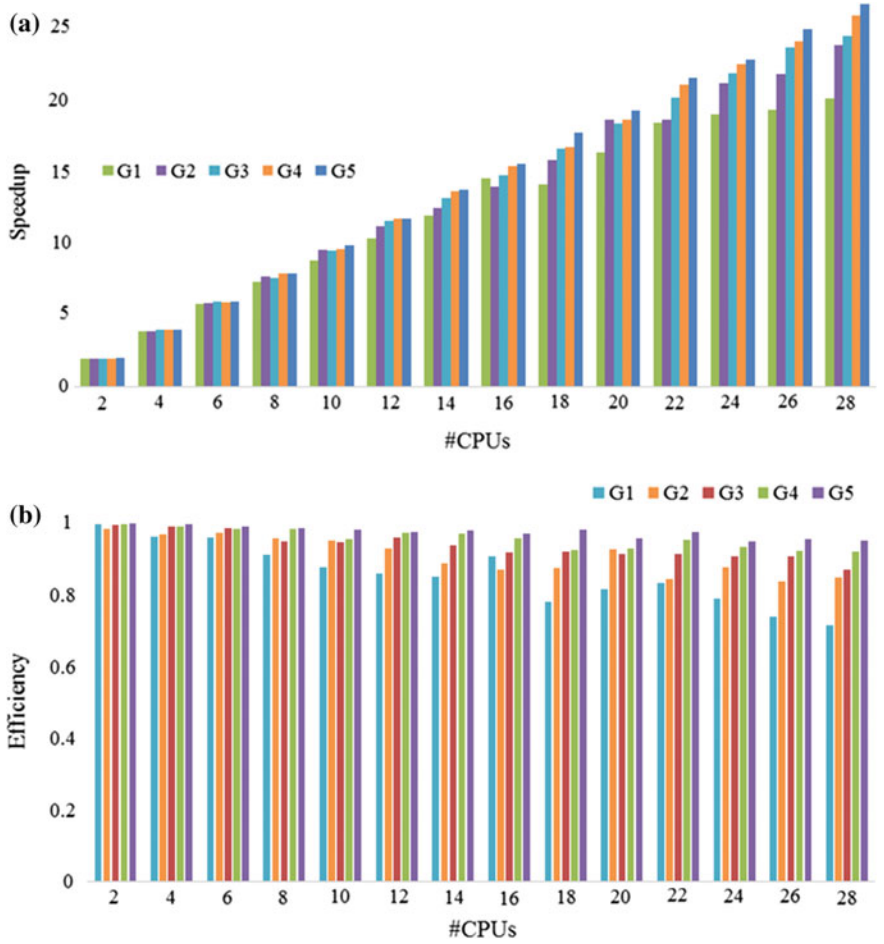

Fig. 9 Speedup and efficiency results in response to the number of CPUs (**a** speedup; **b** efficiency; G1: 10 × 10 subdomains, G2: 15 × 15 subdomains; G3: 20 × 20 subdomains; G4: 25 × 25 subdomains; G5: 30 × 30 subdomains)

but change in the total parallel computing time from 25 × 25 to 30 × 30 decompositions is marginal (about 20 s).

Figure 9 demonstrates speedup and efficiency results over the number of CPUs for the five treatments. Generally, speedup tends to increase as the number of CPUs increases (Fig. 9a). But, increase in speedup when more CPUs are used tend to be slow for coarse spatial domain decomposition. Efficiency results (Fig. 9b) show a decreasing pattern. When the number of CPUs employed is less than 6, the efficiencies of five treatments are close to 1 because the computing performance for

each CPU is similar. But, except for the finest decomposition (30 × 30 tiles), efficiencies of all decomposition treatments tend to decrease as the number of CPUs becomes larger. For 30 × 30 tiles, the efficiencies stay close to the highest efficiency (i.e., 1) under most circumstances. When spatial domain decomposition becomes finer, the computing time associated with each decomposed subdomains (tiles) tend to be smaller. Thus, the computing time for those tasks that are aggregated from multiple subdomains tends to be balanced. As a result, the efficiency of parallel computing for fine spatial domain decomposition tends to be higher than coarse decomposition.

5 Conclusion

In this study, we demonstrated that a high performance and parallel computing approach to interpolate LiDAR data for generating very high resolution DEM. Those DEMs play an essential role in topographic analyses with a focus on micro-level features, which are often of particular importance for a suite of geospatially related science and engineering applications. In this study, the spatial resolution of 0.05 m was used. Our parallel spatial interpolation results show that very high resolution DEMs provide substantial support for delineating micro-level detail of topographic surfaces such as hummocks or hollows in a low relief wetland environment.

The high-performance and parallel computing solution proposed in this study demonstrated its ability to accelerate the generation of very high resolution DEM using spatial interpolation. As more CPUs were introduced, the execution time tends to decrease substantially (e.g., from 20.8 h to 41 min when using 28 CPUs). Our results suggest that when spatial domain decomposition becomes finer, the efficiency of parallel computing tends to be lowered (though the generation of DEM still gains acceleration benefits). In other words, spatial domain decomposition strategies are pivotal in reaping the high-performance computing power for big spatial data analytics.

Future work will concentrate on the following aspects. First, we will extend our approach to a more in depth microtopography analysis. More DEM-derived metrics (e.g., slope and surface roughness) will be introduced to explain and differentiate the microtopography features in our study area. Second, we will further examine other spatial domain decomposition strategies for the acceleration of the generation of very high resolution DEM. Third, but not last, we will apply the proposed parallel computing approach to other study regions.

Acknowledgements We thank support from US NSF XSEDE Supercomputing Resource Allocation (SES170007), and USDA Forest Service grant "Development and Operation of a Web GIS-enabled Data Management System for the Santee Experiment Forest".

References

1. Amoah, J. K. O., Amatya, D., & Nnaji, S. (2013). Quantifying watershed surface depression storage: determination and application in a hydrologic model. *Hydrological Processes, 27*(17), 2401–2413.
2. Anderson, C. J, & Lockaby, B. G. (2011). Forested wetland communities as indicators of tidal influence along the Apalachicola River, Florida, USA. *Wetlands, 31*(5), 895.
3. Armstrong, M. P., & Marciano R. (1993). Parallel spatial interpolation. In *Autocarto-Conference*.
4. Brubaker, K. M., Myers, W. L., Drohan, P. J., Miller, D. A., & Boyer, E. W. (2013). The use of LiDAR terrain data in characterizing surface roughness and microtopography. *Applied and Environmental Soil Science*, 13. https://doi.org/10.1155/2013/891534
5. Cramer, B. E., & Armstrong, M. P. (1999). An evaluation of domain decomposition strategies for parallel spatial interpolation of surfaces. *Geographical Analysis, 31*(2), 148–168.
6. Deilami, K., & Hashim, M. (2011). Very high resolution optical satellites for DEM generation: A review. *European Journal of Scientific Research, 49*(4), 542–554.
7. Ding, Y., & Densham, P. J. (1996). Spatial strategies for parallel spatial modelling. *International Journal of Geographical Information Systems, 10*(6), 669–698.
8. Emerson, C. H., Welty, C., & Traver, R. G. (2005). Watershed-scale evaluation of a system of storm water detention basins. *Journal of Hydrologic Engineering, 10*(3), 237–242.
9. Foster, I. (1995). *Designing and building parallel programs* (Vol. 78). Boston: Addison Wesley Publishing Company.
10. Griffin, L. F., Knight, J. M., & Dale, P. E. R. (2010). Identifying mosquito habitat microtopography in an Australian mangrove forest using LiDAR derived elevation data. *Wetlands, 30*(5), 929–937. https://doi.org/10.1007/s13157-010-0089-8.
11. Guan, X., & Huayi, W. (2010). Leveraging the power of multi-core platforms for large-scale geospatial data processing: Exemplified by generating DEM from massive LiDAR point clouds. *Computers and Geosciences, 36*(10), 1276–1282.
12. HPC. (2016). Windows HPC Cluster Manager. https://technet.microsoft.com/en-us/library/ff919397.aspx.
13. Hickey, R., Smith, A., & Jankowski, P. (1994). Slope length calculations from a DEM within ARC/INFO GRID. *Computers, Environment and Urban Systems, 18*(5), 365–380.
14. Hohl, A., Delmelle, E. M., & Tang, W. (2015). Spatiotemporal domain decomposition for massive parallel computation of space-time kernel density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2*(4), 7.
15. Hohl, A., Delmelle, E., Tang, W., & Casas, I. (2016). Accelerating the discovery of space-time patterns of infectious diseases using parallel computing. *Spatial and Spatio-Temporal Epidemiology, 19,* 10–20.
16. Huang, C.-H., & Bradford, J. M. (1992). Applications of a laser scanner to quantify soil microtopography. *Soil Science Society of America Journal, 56*(1), 14–21.
17. Huang, Q., & Yang, C. (2011). Optimizing grid computing configuration and scheduling for geospatial analysis: An example with interpolating DEM. *Computers and Geosciences, 37*(2), 165–176.
18. Jensen, R. P., Bosscher, P. J., Plesha, M. E., & Edil, T. B. (1999). DEM simulation of granular media—structure interface: Effects of surface roughness and particle shape. *International Journal for Numerical and Analytical Methods in Geomechanics, 23*(6), 531–547.
19. Knight, J. M., Dale, P. E. R., Spencer, J., & Griffin, L. (2009). Exploring LiDAR data for mapping the micro-topography and tidal hydro-dynamics of mangrove systems: An example from southeast Queensland, Australia. *Estuarine, Coastal and Shelf Science, 85*(4), 593–600.
20. Komiyama, A., Santiean, T., Higo, M., Patanaponpaiboon, P., Kongsangchai, J., & Ogino, K. (1996). Microtopography, soil hardness and survival of mangrove (Rhizophora apiculata BL.) seedlings planted in an abandoned tin-mining area. *Forest Ecology and Management, 81*(1), 243–248.

21. Lassueur, T., Joost, S., & Randin, C. F. (2006). Very high resolution digital elevation models: Do they improve models of plant species distribution? *Ecological Modelling, 198*(1), 139–153.
22. Li, Z., Hodgson, M. E., & Li, W. (2016). A general-purpose framework for parallel processing of large-scale LiDAR data. *International Journal of Digital Earth*, 1–22.
23. McKean, J., & Roering, J. (2004). Objective landslide detection and surface morphology mapping using high-resolution airborne laser altimetry. *Geomorphology, 57*(3), 331–351.
24. Milne, L., Lindner, D., Bayer, M., Husmeier, D., McGuire, G., Marshall, D. F., et al. (2008). TOPALi v2: A rich graphical interface for evolutionary analyses of multiple alignments on HPC clusters and multi-core desktops. *Bioinformatics, 25*(1), 126–127.
25. Mitas, L., & Mitasova, H. (1999). Spatial interpolation. *Geographical Information Systems: Principles, Techniques, Management and Applications, 1,* 481–492.
26. Moore, I. D., Grayson, R. B., & Ladson, A. R. (1991). Digital terrain modelling: A review of hydrological, geomorphological, and biological applications. *Hydrological Processes, 5*(1), 3–30.
27. Naoum, S., & Tsanis, I. K. (2003). Hydroinformatics in evapotranspiration estimation. *Environmental Modelling and Software, 18*(3), 261–271.
28. Prasannakumar, V., Vijith, H., & Geetha, N. (2013). Terrain evaluation through the assessment of geomorphometric parameters using DEM and GIS: Case study of two major sub-watersheds in Attapady, South India. *Arabian Journal of Geosciences, 6*(4), 1141–1151.
29. Rauber, T., & Rünger, G. (2013). *Parallel programming: For multicore and cluster systems*, Springer Science & Business Media.
30. Shepard, W. E. (2000). *A parallel approach to searching for nearest neighbors with minimal interprocess communication*. uga.
31. Tang, W., & Feng, W. (2017). Parallel map projection of vector-based big spatial data: Coupling cloud computing with graphics processing units. *Computers, Environment and Urban Systems, 61,* 187–197.
32. Tang, W., & Wang, S. (2009). HPABM: A hierarchical parallel simulation framework for spatially-explicit agent-based models. *Transactions in GIS, 13*(3), 315–333.
33. Tang, W., Feng, W., Zheng, M., & Shi, J. (2017). Land cover classification of fine-resolution remote sensing data. In *Reference module in earth systems and environmental sciences*. Elsevier.
34. Tomczak, M. (1998). Spatial interpolation and its uncertainty using automated anisotropic inverse distance weighting (IDW)-cross-validation/jackknife approach. *Journal of Geographic Information and Decision Analysis, 2*(2), 18–30.
35. Trettin, C. C., Czwartacki, B. J., Allan, C. J., & Amatya, D. M. (2016). Linking freshwater tidal hydrology to carbon cycling in bottomland hardwood wetlands. In Stringer, C. E., Krauss, K. W., Latimer, J. S. (Eds.), *Headwaters to estuaries: Advances in watershed science and management-proceedings of the fifth interagency conference on research in the watersheds* (p. 302). March 2–5, 2015, North Charleston, South Carolina. e-General Technical Report SRS-211. Asheville, NC: US Department of Agriculture Forest Service, Southern Research Station.
36. Wang, S., & Armstrong, M. P. (2003). A quadtree approach to domain decomposition for spatial interpolation in grid computing environments. *Parallel Computing, 29*(10), 1481–1504.
37. Wang, S., & Armstrong, M. P. (2009). A theoretical approach to the use of cyberinfrastructure in geographical analysis. *International Journal of Geographical Information Science, 23*(2), 169–193.
38. Werner, M. G. F. (2001). Impact of grid size in GIS based flood extent mapping using a 1D flow model. *Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere, 26*(7–8), 517–522.
39. Wilkinson, B., & Allen, M. (1999). *Parallel programming: Techniques and applications using networked workstations and parallel computers*. Prentice-Hall.

40. Wise, S. (2000). Assessing the quality for hydrological applications of digital elevation models derived from contours. *Hydrological Processes, 14*(11–12), 1909–1929.
41. Wu, S., Li, J., & Huang, G. H. (2008). A study on DEM-derived primary topographic attributes for hydrologic applications: Sensitivity to elevation data resolution. *Applied Geography, 28*(3), 210–223.
42. Zikopoulos, P., & Eaton, C. (2011). *Understanding big data: Analytics for enterprise class hadoop and streaming data*, McGraw-Hill Osborne Media.
43. Zimmerman, D., Pavlik, C., Ruggles, A., & Armstrong, M. P. (1999). An experimental comparison of ordinary and universal kriging and inverse distance weighting. *Mathematical Geology, 31*(4), 375–390.