

GPU UTILIZATION IN GEOPROCESSING BIG GEODATA: A REVIEW

Sazlieya Saupi Teri, Ivin Amri Musliman, Alias Abdul Rahman

3D GIS Research Lab, Faculty of Built Environment and Surveying,
Universiti Teknologi Malaysia, Skudai, MALAYSIA
sazlieya.teri@gmail.com, ivinamri@utm.my, alias@utm.my

Commission 4, WG 7

KEY WORDS: Graphics Processing Unit (GPU), General Purpose GPU (GPGPU), Parallel Processing, Big Geodata, Geoprocessing.

ABSTRACT

The expansion of data collection from remote sensing and other geographic data sources, as well as from other technology such as cloud, sensors, mobile, and social media, have made mapping and analysis more complex. Some geospatial applications continue to rely on conventional geospatial processing, where limitation on computation capabilities often lacking to attain significant data interpretation. In recent years, GPU processing has improved far more GIS applications than using CPU alone. As a result, numerous researchers have begun utilising GPUs for scientific, geometric, and database computations in addition to graphics hardware use. This paper summarizes parallel processing concept and architecture, the development of GPU geoprocessing for big geodata ranging from remote sensing and 3D modelling to smart cities studies. This paper also addresses the GPU future trends advancement opportunities with other technologies, machine learning, deep learning, and cloud-based computing.

1. INTRODUCTION

Graphic Processing Unit (GPU) hardware has obtained significant advancement in geospatial applications in recent years. Such progress has been primarily fuelled by the expansion of data collection increments (remote sensing, point clouds, laser scanning, sensors). The development of big geodata geoprocessing using GPU parallel computation enables the computer capacity to accelerate the processing efficiently and dynamically.

The widely availability of location-based data now becoming a growing service and have revolutionized in every industry. The datasets collection comes in variety of technology such as satellite, Unmanned Aerial Vehicle (UAV), Light Detection and Ranging (LiDAR), geotagging or even from web-scraping. The application of these geospatial intelligence in e-commerce services for example, helped expanding e-commerce sites for better location visualization, analytical and operational decision-support systems (Chowhan, 2017). This continuously data collection process has made the growth of data storage and gave birth to a relatively new term called big geospatial data.

Big geodata analytics receives a lot of attention these days because it allows users to analyse massive amounts of geospatial data. Big geodata refers to spatial datasets that exceed the capacity to be processed by traditional computing systems (J. Lee & Kang, 2015). Handling these scalable datasets demands complex computational developments and robust computing technologies in order to accomplish the tasks efficiently (Adam et al., 2014). Various ways have been prepared by researchers to process these big geodata, and some have invested a lot of money and time to solve the challenges of processing this data such as High-Performance Computing (HPC), cloud computing and even GPU.

Li (2020), reviewed HPC can be a new framework paradigm to accommodate this big geodata challenges. In fact, this HPC has been used for two decades for the management of this geospatial data (Clarke, 2003; S. Wang & Armstrong, 2003). The expansion

computing technologies from serial computation using Central Processing Units (CPU) to HPC earned the benefits in geospatial field, particularly for big geodata. However, the biggest challenge of maintaining this HPC technology is very costly (Netto et al., 2018) and often this technology is played by large giant companies such as Google, and Amazon. To handle this cost issue, the deployment of Graphic Processing Units (GPU) is much more treasured as this hardware proposes the similar parallelism architecture as HPC, and as we aware very consumer cost friendly (Hasan & Chakraborty, 2021; Jagtap & Rao, 2020) and easily accessible (Chen et al., 2014).

The remaining section is organized as follows: Section 2 provides an overview of GPU developments for big geodata varies from the remote sensing, 3D modelling and smart cities. Section 3 presents the general concepts of parallel geoprocessing architecture, GPU for big geodata, existing GPU programming model and the best practice for speedup performance evaluation technique, Amdahl's Law. Section 4 summaries the overall paper and future trends GPU utilization.

2. RELATED WORKS ON GPU DEVELOPMENTS FOR BIG GEODATA

The increasing potential usage of GPU hardware has demonstrated the GPU's capacity to give faster applications and excellent task-based parallelism that provides a significant increase in processing performance. This potential has been used in a variety of fields, including cryptocurrencies such as Bitcoin to mine digital money (Alkaeed et al., 2020; Iyer & Pawar, 2018), DNA analysis and modelling (Ahmed et al., 2020), large-scale simulations (Saprykin et al., 2019; Vu & Tan, 2019), large-scale tasks with combination of machine learning (Abadi et al., 2016; Nie et al., 2018) and of course, big geodata (Breunig et al., 2020; Zhou et al., 2019).

The current challenges of increasing geospatial datasets magnitude has encouraged researchers all over the world to contribute their simulation approaches and techniques for processing these scalable datasets derived from a variety of high

resolution information sources, such as satellite imageries, remote sensing products, aerial photographs, and web scrapping location extraction (Hoang & Mothe, 2018). The research development can be seen in a few applications such as satellite imageries processing, building conversation, and modelling.

2.1 Remote Sensing

The use of GPU in remote sensing has presented significant computing problems as remote sensing datasets have grown in term of capacity. Traditional geoprocessing methods are no longer capable of supporting these large datasets. (P. Liu et al., 2014). Previously, the utilization GPU functions included satellite imageries classification (Sharma et al., 2020), real-time radiometric correction (Fang et al., 2014), soil parameter inversion (Yin et al., 2020), noise removal (Granata et al., 2020) and hyperspectral image classification (Yusuf & Alawneh, 2018). Some of these applications are being optimised using NVIDIA's application programming interface (API), Compute Unified Device Architecture (CUDA) (Fang et al., 2014; Sharma et al., 2020; Yin et al., 2020) and OpenCL (Granata et al., 2020), an open-source API used for NVIDIA or AMD manufactured GPU. These studies displayed satellite image processing able to demonstrate a good flexibility to GPU computational elements.

Using GPU to accelerate satellite image processing has a huge influence on the remote sensing industry. Sharma *et al.*, (2020) investigated the feasibility of employing GPU to accelerate batch processing of spatial raster data. They concluded that the GPU is capable of drawing conclusions about its applicability in solving various problems related to geoinformation and its efficiency processes by using neural network training for segmenting images of 10 classes that included ground, non-ground, and man-made features of 1601 images.

Li *et al.*, (2019) have created a parallel computing toolkit for terrain analysis that makes use of GPUs to accelerate the processing of large-scale LiDAR datasets when compared to the CPU alone, the experiment showed good speedup ratios performance with a maximum memory size of 8 GB. The toolkit can analyse points and doing statistical calculations. The toolkit demonstrated considerable time savings and addressed how computing costs may be lowered by making it available through GPU, particularly for LiDAR geoprocessing and other derived products.

2.2 3D Modelling

Condorelli *et al.*, (2020) proposed a method for automatically recognising lost architectural heritage in video footage to build a metrically validated 3D model. The experiment focuses on reducing human effort in the detection of the searched object and increasing the operator's efficiency in the archive. Machine Learning techniques were recognised as viable solutions to reduce the time required to search for monuments in historical video footage, and the usage of GPU has increased the computational efficiency of such algorithms.

2.3 Smart Cities

3D point cloud technology enables the automated (Paffenholz et al., 2021) and extremely detailed digital capture of real-world (Y. Li et al., 2021) settings such as assets (S. Y. Lee et al., 2013), sites area (Shih & Wang, 2014), cities (Buyuksalih et al., 2019; Park & Guldmann, 2019), and vegetation (Mesas-Carrascosa et al., 2020; Yang et al., 2015); the obtained 3D point clouds represent an important type of geospatial data that categorized

and utilised in wide range of geoinformation applications and systems (Ariff et al., 2020; Discher et al., 2018; Richter et al., 2015).

Ariff *et al.*, (2020) experimented 3D model meshing of 2.77 GB Putrajaya city using two different GPU, NVIDIA GEFORCE GTX 1070 and NVIDIA GEFORCE GTX 850M and they concluded that a graphic card with a higher graphic and computing performance is essential for smoother and faster visualization of 3D model and shorter time processing and lesser technical issue.

Many smart city applications, such as flood risk monitoring (Jing et al., 2019; W. Li et al., 2020), emergencies response (Alazawi et al., 2014), and solar energy potential (Amado & Poggi, 2012; Pilipczuk, 2020) of parcels and rooftops, require 3D maps to contain the height dimension. Making a city smart begins with the construction of a detailed 3D map of the entire city. This data magnitude can be enhanced by applying and utilized GPU for geospatial data processing for better decision-making.

3. PARALLEL GEOPROCESSING FOR BIG GEODATA

Concerning problems occur in big geodata because to the complexity of dataset architecture (Zhao et al., 2016), segmentation (Ruan & Liu, 2020), classification (Uy et al., 2019), and data interaction (Poux et al., 2016). For example, the nature of large 3-Dimensional (3D) dataset irregularity continues to be a barrier in terms of effectively collecting, processing, and storing these large semi-structured/unstructured information (Breunig et al., 2020). Thus, prior studies have used Hadoop-based frameworks (C. Wang et al., 2013), data compression (Javed et al., 2020), cloud computing (K. Liu & Boehm, 2015), and HPC (Zhenlong Li, 2020) to analyse and manage big geodata. Using GPU technology is another significant approach for dealing big geodata processing (Granata et al., 2020; Petrovič et al., 2020; Venugopal & Kannan, 2013).

Some GIS developers (Franklin et al., 2018; Huang, 2012; Mineter et al., 2000; Puri & Prasad, 2015; Singh et al., 2019) developed programs to help solving geospatial problems. The geospatial processing challenges are normally related to storing (Sui, 2018), managing (Yao & Li, 2018), processing, analysing, visualizing (Sipe & Dale, 2003) and verifying the quality of data (S. Li et al., 2016).

Satellites imageries processing (Yusuf & Alawneh, 2018), coordinate transformations (Chavez, 2016; J. Li et al., 2017; Zhu & Wu, 2021), building modelling (Ma et al., 2008; Schweizer, 2015), and other GIS and remote sensing applications can benefit from the parallelism provided by GPGPU programming. For a history of parallel solutions for remote sensing applications spanning from multi-core CPU implementations to early GPU systems Christophe *et al.*, (2011) attained up to 10 to 400 performance gains for processing several gigabytes of remote sensing images by considering the data scalability and data streaming using two programming model, CUDA and OpenCL investing a USD200 GPU hardware.

Aside from raster processing, vector processing has gained popularity in recent years. Since LiDAR technology has become more popular in recent years, the necessity for vector data processing for both raw and processed vector data cannot be overlooked. Mobile Laser Scanning (MLS) and Terrestrial Laser Scanning (TLS) sensors able to emits millions of pulses per seconds and this made data collecting denser and more accurate. H. Wang et al., (2017) used a heterogeneous CPU and GPU to

accelerate a massive point cloud to run a hybrid parallel Spatial Interpolation framework on their systems. They recognise that adopting this CPU-GPU hybrid accommodation will be more practicable for processing big point cloud collections in future.

3.1 Concept of Parallel Architecture

The hype and expansion of parallel processing came in response to process growing frequencies of the data and the alternative to improve the number of processors is no longer practical after a certain point. Therefore, the researchers try to tackle and address the issues by designing the parallel systems with the aim to produce more processing efficiency using multiple cores (Surakhi et al., 2018), manage the power consumption (Garcia et al., 2018; Jin et al., 2017) and CPU overheating (Jin et al., 2017; Rajachandrasekar et al., 2012).

Parallel processing is the use of several processing cores to handle a job or tasks at the same time, whereas sequential processing is the use of a central processing unit (CPU) to carry out just one action at a time (Memon et al., 2017). Traditionally, software has been developed sequentially, which is a simpler technique but as the data magnitude keep increasing, the software become severely constrained by the processor's speed and capacity to execute each sequence of instructions (Rahmawati, 2009).

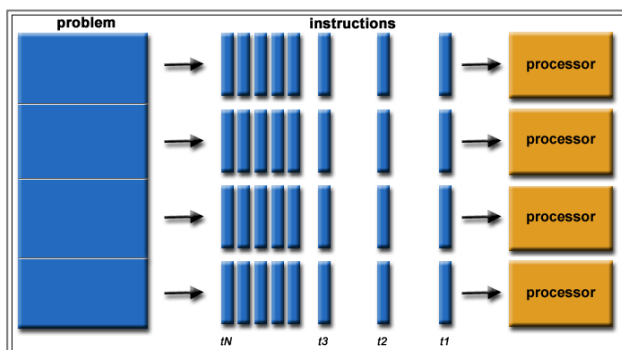


Figure 1. General parallel processing architecture in a single machine (Source: OmniSci).

Figure 1 shows the fundamental parallel architecture in general. The larger tasks are broken down into smaller tasks or calculations, which are individualistic and normally identical tasks that are executed at the same time by multiple processors communicating via machine shared memory, and the results of all small tasks are merged as the final product upon completion (Pervan & Knezovic, 2020). Giant corporations utilise many approaches for big scale structure and high-performance computation including distributed computing, multi-core computing, cluster computing, and grid computing, where this structure may handle the continuous usage (Kayum & Rogowski, 2020) and high-capacity machines (Pavilion & Performance, 2019).

The essential of utilizing parallel processing for geospatial analysis functions is undoubtedly relevant gradually over time due to circumstances of datasets size and the efficiency for high performance processing. The challenges to process big geodata still underdeveloped by per function basis and not fully enhanced by geospatial researchers' and community due to hardware complexity and underutilized (Kang et al., 2018; J. Lee & Kang, 2015; Stojanovic & Stojanovic, 2013).

Previous study on processing big geodata, has proposed to utilise parallel processing for high efficiency large-scale geospatial processing (Zhou et al., 2019) and data memory management (Doraiswamy & Freire, 2020). Armstrong (2020) suggested by employing parallel processing to handle geographical information big datasets, the potential to stimulate the computing standard in spatial problem solving, analysis, and modelling will be clear developmental path in geospatial research.

There are two types of parallel programming commonly used in GIS applications: data parallelism and tasks parallelism. Both types can be used simultaneously to increase parallelism, which is referred to as hybrid parallelism (H. Wang et al., 2017). The general tasks and data parallelism is shown in Figure 3 below.

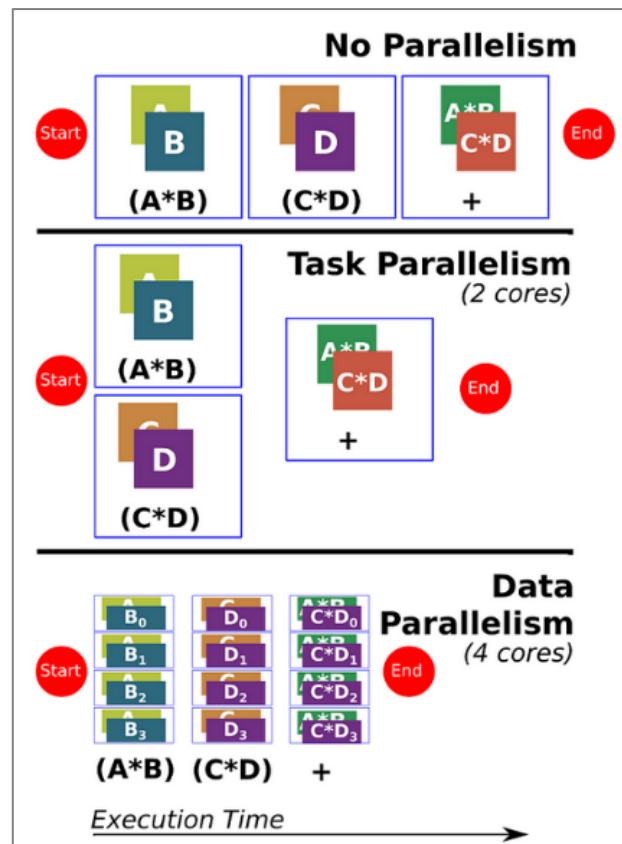


Figure 3. Parallelism in geospatial application: Task parallelism and data parallelism (Source: Google).

In Table 1 shows the comparison summary between task parallelism and data parallelism.

Parallelism Type	Task	Data
Definition	Processing on separate jobs to execute on the same data at the same time, a sequence of activities and gives tasks that can be done concurrently when there are several jobs (Qin, 2017)	Execution of the same task but different data partition (Krizhevsky, 2014). Also known as Spatial domain decomposition is the name for this procedure (Liang et al., 2020).

Datasets	picture tiles in remote sensing, running dozens of spatial models, or conducting hundreds of simulations spatial statistics (Tzeng et al., 2012)	vector features or raster pixels (J. Zhang & You, 2012)
Advantages	<ul style="list-style-type: none"> the data remains unaltered, task parallelism may be easier to construct 	Variety of techniques: <ul style="list-style-type: none"> row column decomposition (J. Li et al., 2017) matrix decomposition (Galoppo et al., 2005), grid decomposition (Caggianese & Erra, 2012) Quadtree partitioning (J. Zhang & York, 2019) recursive bisection (Chisholm, 2020)
Disadvantages	<ul style="list-style-type: none"> need enough jobs to divide and spread across several cores. 	<ul style="list-style-type: none"> Partitioning spatial irregular data such as points, lines, and polygons into smaller partitions Balancing the amount of data in each sub partition.

Table 1. Task Parallelism and Data Parallelism summary

3.2 GPU for Big Geodata

The potential of using GPU is seen able to assist and expand the big geodata processing. Eckart *et al.*, (2016) presented the potential of GPU parallelization during GTC Silicon Valley-2019 for the use in large-scale 3D point clouds processing and beneficial for real-time computation over tens of thousands of points. The use of GPGPU to show significant impact on improving data processing efficiency in terms of time consuming (Gao et al., 2019) and data processing management (Yao & Li, 2018; J. Zhang et al., 2019).

The continuous demand for faster and more efficient development via graphics drives appears to bode well for the future. Due to a few constraints, GPU implementation is still underdeveloped for existing GIS processing products and tools. Stojanovic & Stojanovic (2013) identify the increment geographic data collection have gathering the impacts of analyses and geoprocessing complexity and computational intensity in many GIS application domains due to a lack of parallel GIS algorithms, application libraries, and toolkits in parallel architectures. With the evolution of GPU production, manufacturers launched a range of models with varying total CUDA cores and specialties; as a result, periodic driver updates are required to ensure the drivers operate at their full capacity. Additionally, as bitcoin mining has grown in popularity, miners have discovered that using GPUs to mine the digital currency

enables them to expedite their mining, which has impacted the market's GPU pricing as a result of the increasing demands.

The initial application of GPU is mostly been utilised to render the gaming pipeline (Mišić et al., 2012) so that the images visualizing more realistic, and the image texture is more smoother (Crow, 2004). The largest manufacturers of GPU hardware that are in the market right now are NVIDIA and AMD. Each GPU model released has different cores, depending on the price. For the low-end NVIDIA GPU has about 700 CUDA cores while the high end has more than 4000 CUDA cores (NVIDIA, 2021). Performance for each model is affected by the number of cores (K. H. Choi et al., 2014) and GPU frequency clocks speed (H. J. Choi et al., 2014).

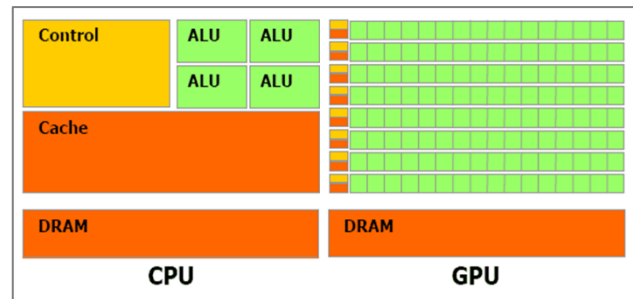


Figure 2. Comparisons between CPU core and GPU core (Source: Google).

Figure 2 shows the CPU architecture structure is built to handle various types of tasks and functions in series while the GPU is more data intensive and processed simultaneously. Measuring performance in sequential programming is considerably less complicated and essential than benchmarking in parallel computing since it generally involves finding bottlenecks in the system (Chandrashekar et al., 2021). To bypass bottleneck issue in sequential processing, transferring data across the memory hierarchy is remarkably helping to perform parallel processing onto data science, machine learning, and artificial intelligence. While parallel computing is more difficult and has a higher initial cost, the benefit of being able to solve a problem quicker, frequently surpasses the expenses of purchasing parallel computing hardware. Benchmarks in parallel computing may be accomplished using execution time (Hoefler & Belli, 2015; Lam et al., 2014), performance regression (Jia et al., 2015; Nagasaka et al., 2010) and measuring techniques, such as statistical (Y. Zhang et al., 2011) and repetitions transfer cycle (Bridges et al., 2016).

3.3 GPU Programming Model

Many GPU programming approaches, and data structures are shared by programming model such as CUDA, created and maintained as a product by NVIDIA (NVIDIA, 2021), and Open Computing Language (OpenCL), an open-source language supervised by the Khronos Group (OpenCL, 2021). Because CUDA are limited to NVIDIA hardware, OpenCL is more often used for cross-platform development. However, to attain the same degree of speed as a functionally similar CUDA application, OpenCL programming language may require more manual tuning (Filipović et al., 2017; Petrović et al., 2020; Thoman et al., 2011; van Werkhoven, 2019). Many mainstream programming languages, including C, C++, and Python, have CUDA and OpenCL library bindings supported (Jacob et al., 2010; Sarkar, 2007).

3.4 Speedup Performance Evaluation

Measuring the level of performance in parallel computing architecture, the best practice is to utilize Amdahl's Law formula is always be used., Amdahl's Law calculates how much a computation can be speeded up by running the computation in parallel. Amdahl's Law is named after Gene Amdahl who proposed the formula in 1967 (Amdahl, 1967).

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}} \quad (1)$$

Where,

S_{latency} = the theoretical speedup of the execution of the whole task

s = the speedup of the part of the task that benefits from improved system resources

p = the proportion of execution time that the part benefiting from improved resources originally occupied

However, the number of simultaneous processes practically grows in proportion to the complexity of the task because the load is fixed according to Amdahl's Law, the execution time decreases as the number of processors rises. Amdahl's law is useful in situations when response speed is important (Rocha et al., 2016).

In relation to time that takes to execute a given routine, the CPU outperforms the GPU for smaller workloads. As the magnitude of the workload and the number of datasets arise, the execution time of the GPU begins to decrease in comparison to that of the CPU. Such as for 100% occupancy, the GPU did the addition work faster than the CPU alone. In terms of throughput, or the number of activities executed per second, GPU has always outperformed CPU in many applications. As GPU architecture will remain to progress, GPUs will be fully utilized for throughput rather than latency.

4. CONCLUSION

This paper has discussed the development of GPUs for big geodata, ranging from remote sensing to 3D modelling and smart cities. There are some researchers have seized the graphics hardware platform to accelerate their geoprocessing functions and tools and the results promising a positive performance efficiency. In this paper also introduces the fundamental ideas of parallel geoprocessing architecture, GPU for big geodata, existing GPU programming models, and the best practise for speedup performance evaluation, Amdahl's Law. At the end of the paper, a remarkable view on future trends of the GPU utilisation in other platform such as machine learning, deep learning, TPU, AI, cloud computing and the beneficial use in geospatial and remote sensing aspect.

Processing big scale datasets is commonly used as an evaluation criterion since it shows how quickly a parallel application can solve a problem. However, for enormous issues like processing satellite imageries or other big geodata, without proper data and tasking management assigned during processing may weaken the efficiency results and allowing you to assess how well a parallel programme can scale a massive geospatial processing.

GPUs processing are fast because their high-bandwidth features and technology performed arithmetic processing substantially faster than conventional CPUs. The primary function of GPUs is

to do the calculations required to render three-dimensional (3D) computer graphics. The GPU's ability to process massive datasets is consistent with the nature of machine learning. Processing large scale of datasets is essentially what Machine Learning (ML) and Deep Learning (DL) does, which is why GPUs are advantageous for machine learning jobs. TensorFlow and Pytorch are two libraries that make advantage of GPUs that users can now manipulate the data frames and perform machine learning algorithms on GPUs as well.

The emergence of artificial Intelligence (AI) and autonomous systems have brought breakthroughs in processing capacity, whether GPUs, TPUs, or CPUs, and algorithmic advancements. Google introduced their first Tensors Processing Units (TPU) processing, ASIC, as a solution for their datacentre's computation demands. The heterogenous between CPUs, GPUs, TPUs could enhance more on cost-performance processing.

With the rise of cloud computing, several businesses have offered a cloud-based service as a service. Where users are not required to run a processing on their machine using hardware or software. As a result, significant resources are spent on idle GPUs, while insufficient GPUs are used to run jobs. Manifold is one of the successful companies providing on-the-cloud GPU processing services where the users can automatically run their code on Manifold SQL server and experience the embarrassingly parallelism and performance acceleration.

As GPU advancements become more prevalent in the near future, it is reasonable to anticipate an increase in the number of solutions and geoprocessing tools that leverage GPU platform implementations in geospatial and remote sensing applications. This is likely to broaden the range and complexity of available geoprocessing tools, as well as facilitate geospatial and remote sensing projects that involve large collections of spatial datasets. The integration of artificial intelligence and geospatial technologies has created tremendous unimaginable potential. The advancement of AI, machine learning, deep learning, and other technologies enables researchers and authorities to make more precise town development decisions by deploying predictive town planning in all aspects and future insights and proactive planning for Covid-19 pandemic by harvesting and archiving geo-located crowdsourcing, human movement mapping/tracking, and projection of spatial and temporal trends.

REFERENCES

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. <http://arxiv.org/abs/1603.04467>
- Adam, K., Adam, M., Fakharaldien, I., Zain, J. M., & Majid, M. A. (2014). Big Data Management and Analysis. March 2017.
- Ahmed, N., Bertels, K., & Al-Ars, Z. (2020). Efficient GPU Acceleration for Computing Maximal Exact Matches in Long DNA Reads. Proceedings of the 2020 10th International Conference on Bioscience, Biochemistry and Bioinformatics.
- Alazawi, Z., Alani, O., Abdjbar, M. B., Altowajri, S., & Mehmood, R. (2014). A smart disaster management system for future cities. WiMobCity 2014 - Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies

- for Smart Cities, Co-Located with MobiHoc 2014, 1–10. <https://doi.org/10.1145/2633661.2633670>
- Alkaeed, M., Alamro, Z., Al-Ali, M., Al-Mohammed, H. A., & Khan, K. (2020). Highlight on Cryptocurrencies Mining with CPUs and GPUs and their Benefits Based on their Characteristics. 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), 67–72.
- Amado, M., & Poggi, F. (2012). Planning For Solar Smart Cities. *DisP - The Planning Review*, 48(2), 4–5. <https://doi.org/10.1080/02513625.2012.731823>
- Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. *Psychoanalytic Review*, 98(5), 597–612. <https://doi.org/10.1521/prev.2011.98.6.597>
- Ariff, S. A. M., Azri, S., Ujang, U., Nasir, A. A. I. M., Fuad, N. A., & Karim, H. (2020). Exploratory study of 3D point cloud triangulation for smart city modelling and visualization. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 44(4/W3), 71–79. <https://doi.org/10.5194/isprs-archives-XLIV-4-W3-2020-71-2020>
- Armstrong, M. P. (2020). High Performance Computing for Geospatial Applications: A Retrospective View. 1998, 9–25. https://doi.org/10.1007/978-3-030-47998-5_2
- Breunig, M., Bradley, P. E., Jahn, M., Kuper, P., Mazroob, N., Rösch, N., Al-Doori, M., Stefanakis, E., & Jadidi, M. (2020). Geospatial data management research: Progress and future directions. *ISPRS International Journal of Geo-Information*, 9(2). <https://doi.org/10.3390/ijgi9020095>
- Bridges, R. A., Imam, N., & Mintz, T. M. (2016). Understanding GPU power: A survey of profiling, modeling, and simulation methods. *ACM Computing Surveys*, 49(3). <https://doi.org/10.1145/2962131>
- Buyuksalih, G., Baskaraca, P., Bayburt, S., Buyuksalih, I., & Abdul Rahman, A. (2019). 3D city modelling of Istanbul based on LiDAR data and panoramic images – Issues and challenges. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(4/W12), 51–60. <https://doi.org/10.5194/isprs-archives-XLII-4-W12-51-2019>
- Caggianese, G., & Erra, U. (2012). GPU accelerated multi-agent path planning based on grid space decomposition. *Procedia Computer Science*, 9(December), 1847–1856. <https://doi.org/10.1016/j.procs.2012.04.203>
- Chandrashekhara, B. N., Sanjay, H. A., & Srinivas, T. (2021). Performance Analysis of Parallel Programming Paradigms on CPU-GPU Clusters. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021, ICAIS*, 646–651. <https://doi.org/10.1109/ICAIS50930.2021.9395977>
- Chavez, D. (2016). Parallelizing Map Projection of Raster Data on Multi-core CPU and GPU Parallel Programming Frameworks.
- Chen, Z., Wang, J., He, H., & Huang, X. (2014). A fast deep learning system using GPU. *Proceedings - IEEE International Symposium on Circuits and Systems*, 1, 1552–1555. <https://doi.org/10.1109/ISCAS.2014.6865444>
- Chisholm, R. (2020). Working With Incremental Spatial Data During Parallel (GPU) Computation By : January.
- Choi, H. J., Son, D. O., Kim, C. H., & Kim, J. M. K. (2014). Impact of clock frequency and number of cores on gpu performance. 2014 International Conference on IT Convergence and Security, ICITCS 2014, 2–5. <https://doi.org/10.1109/ICITCS.2014.7021776>
- Choi, K. H., Jung, D., Kim, S. W., Hwang, T. H., Kwon, J. S., & Kim, D. S. (2014). Study of GPU scalability. *Proceedings of the International Symposium on Consumer Electronics, ISCE*, June, 14–16. <https://doi.org/10.1109/ISCE.2014.6884421>
- Chowhan, R. S. (2017). Remote Sensing , Gis And E-Commerce : Technological Tools To Sustain Agricultural Development. 6(4), 2421–2425.
- Christophe, E., Christophe, E., Michel, J., Michel, J., & Inglada, J. (2011). Remote Sensing Processing: From Multicore to GPU. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(3), 643–652. <https://doi.org/10.1109/JSTARS.2010.2102340>
- Clarke, K. C. (2003). Geocomputation 's future at the extremes : high performance computing and nanoclients. 29, 1281–1295. <https://doi.org/10.1016/j.parco.2003.03.001>
- Condorelli, F., Rinaudo, F., Salvatore, F., & Tagliaventi, S. (2020). A neural networks approach to detecting lost heritage in historical video. *ISPRS International Journal of Geo-Information*, 9(5). <https://doi.org/10.3390/ijgi9050297>
- Crow, T. S. (2004). Evolution of the Graphical Processing Unit. December 2004, 59. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.368&rep=rep1&type=pdf%5Cnhttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.142.368>
- Discher, S., Richter, R., & Döllner, J. (2018). A scalable WebGL-based approach for visualizing massive 3D point clouds using semantics-Dependent rendering techniques. *Proceedings - Web3D 2018: 23rd International ACM Conference on 3D Web Technology*. <https://doi.org/10.1145/3208806.3208816>
- Doraiswamy, H., & Freire, J. (2020). A GPU-friendly Geometric Data Model and Algebra for Spatial Queries. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1875–1885. <https://doi.org/10.1145/3318464.3389774>
- Eckart, B., Kim, K., Troccoli, A., Kelly, A., & Kautz, J. (2016). Accelerated generative models for 3D point cloud data. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, 5497–5505. <https://doi.org/10.1109/CVPR.2016.593>
- Fang, L., Wang, M., Li, D., & Pan, J. (2014). CPU/GPU near real-time preprocessing for ZY-3 satellite images: Relative radiometric correction, MTF compensation, and geocorrection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 229–240. <https://doi.org/10.1016/j.isprsjprs.2013.11.010>
- Filipovič, J., Petrovič, F., & Benkner, S. (2017). Autotuning of OpenCL Kernels with global optimizations. *ACM International Conference Proceeding Series*, Part F1322. <https://doi.org/10.1145/3152821.3152877>
- Franklin, W. R., De Magalhães, S. V. G., & Andrade, M. V. A.

- (2018). Data structures for parallel spatial algorithms on large datasets. *Proceedings of the 7th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial* 2018, 16–19. <https://doi.org/10.1145/3282834.3282839>
- Galoppo, N., Govindaraju, N. K., Henson, M., & Manocha, D. (2005). LU-GPU: Efficient algorithms for solving dense linear systems on graphics hardware. *Proceedings of the ACM/IEEE 2005 Supercomputing Conference, SC'05*, 2005(c). <https://doi.org/10.1109/SC.2005.42>
- Gao, C., Baig, F., Vo, H., Zhu, Y., & Wang, F. (2019). Accelerating Cross-Matching Operation of Geospatial Datasets using a CPU-GPU Hybrid Platform. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 3402–3411. <https://doi.org/10.1109/BigData.2018.8622600>
- Garcia, A. M., Schepke, C., Girardi, A. G., & Da Silva, S. A. (2018). Power consumption of parallel programming interfaces in multicore architectures: A case study. *Proceedings - 2018 Symposium on High-Performance Computing Systems, WSCAD* 2018, 77–83. <https://doi.org/10.1109/WSCAD.2018.00021>
- Granata, D., Palombo, A., Santini, F., & Amato, U. (2020). Noise removal from remote sensed images by non local means with OpenCL algorithm. *Remote Sensing*, 12(3), 1–19. <https://doi.org/10.3390/rs12030414>
- Hasan, K. M. A., & Chakraborty, S. (2021). GPU Accelerated Tensor Computation of Hadamard Product for Machine Learning Applications. *2021 International Conference on Information and Communication Technology for Sustainable Development, ICICT4SD* 2021 - Proceedings, 1–5. <https://doi.org/10.1109/ICICT4SD50815.2021.9396980>
- Hoang, T. B. N., & Mothe, J. (2018). Location extraction from tweets. *Information Processing and Management*, 54(2), 129–144. <https://doi.org/10.1016/j.ipm.2017.11.001>
- Hoefler, T., & Belli, R. (2015). Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. In *Nature* (Vol. 388, pp. 539–547).
- Huang, X. (2012). Parallel Buffer Generation Algorithm for GIS. *Journal of Geology & Geosciences*, 02(02). <https://doi.org/10.4172/2329-6755.1000115>
- Iyer, S. G., & Pawar, A. D. (2018). GPU and CPU Accelerated Mining of Cryptocurrencies and their Financial Analysis. *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2018 2nd International Conference On, 599–604.
- Jacob, F., Whittaker, D., Thapaliya, S., Bangalore, P., Mernik, M., & Gray, J. (2010). CUDACL: A tool for CUDA and OpenCL programmers. *17th International Conference on High Performance Computing, HiPC* 2010, May 2014. <https://doi.org/10.1109/HIPC.2010.5713193>
- Jagtap, S. V., & Rao, Y. S. (2020). GPU accelerated circuit analysis using machine learning-based parallel computing model. *SN Applied Sciences*, 2(5), 1–13. <https://doi.org/10.1007/s42452-020-2667-6>
- Javed, M., Meraz, M., & Chakraborty, P. (2020). A Quick Review on Recent Trends in 3D Point Cloud Data Compression Techniques and the Challenges of Direct Processing in 3D Compressed Domain. *ArXiv*, 1–6.
- Jia, W., Garza, E., Shaw, K. A., & Martonosi, M. (2015). GPU Performance and Power Tuning Using Regression Trees (pp. 1–26).
- Jin, C., de Supinski, B. R., Abramson, D., Poxon, H., DeRose, L., Dinh, M. N., Endrei, M., & Jessup, E. R. (2017). A survey on software methods to improve the energy efficiency of parallel computing. *International Journal of High Performance Computing Applications*, 31(6), 517–549. <https://doi.org/10.1177/1094342016665471>
- Jing, C., Du, M., Li, S., & Liu, S. (2019). Geospatial dashboards for monitoring smart city performance. *Sustainability (Switzerland)*, 11(20). <https://doi.org/10.3390/su11205648>
- Kang, X., Liu, J., Dong, C., & Xu, S. (2018). Using high-performance computing to address the challenge of land use/land cover change analysis on spatial big data. *ISPRS International Journal of Geo-Information*, 7(7), 20–22. <https://doi.org/10.3390/ijgi7070273>
- Kayum, S., & Rogowski, M. (2020). High-performance computing applications transition to the cloud in upstream. *4th EAGE Workshop on High Performance Computing for Upstream in the Middle East: HPC Through the 4th Industrial Revolution*. <https://doi.org/10.3997/2214-4609.201903299>
- Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. <http://arxiv.org/abs/1404.5997>
- Lam, B. C., Barboza, A., Agrawal, R., & George, A. D. (2014). Benchmarking Parallel Performance on Many-Core Processors. In *First Workshop, OpenSHMEM 2014 Annapolis, MD, USA, March 4–6, 2014 Proceedings: Vol. 8356 LNCS*. https://doi.org/10.1007/978-3-319-05215-1_9
- Lee, J., & Kang, M. (2015). Geospatial Big Data : Challenges and Opportunities ☆. *Big Data Research*, 2(2), 74–81. <https://doi.org/10.1016/j.bdr.2015.01.003>
- Lee, S. Y., Majid, Z., & Setan, H. (2013). 3D data acquisition for indoor assets using terrestrial laser scanning. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(2W1), 221–226. <https://doi.org/10.5194/isprsannals-II-2-W1-221-2013>
- Li, J., Finn, M. P., & Castano, M. B. (2017). A Lightweight CUDA-based parallel map reprojection method for raster datasets of continental to global extent. *ISPRS International Journal of Geo-Information*, 6(4), 1–18. <https://doi.org/10.3390/ijgi6040092>
- Li, S., Dragicevic, S., Castro, F. A., Sester, M., Winter, S., Coltekin, A., Pettit, C., Jiang, B., Haworth, J., Stein, A., & Cheng, T. (2016). Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115, 119–133. <https://doi.org/10.1016/j.isprsjprs.2015.10.012>
- Li, W., Batty, M., & Goodchild, M. F. (2020). Real-time GIS for smart cities. *International Journal of Geographical Information Science*, 34(2), 311–324. <https://doi.org/10.1080/13658816.2019.1673397>

- Li, Y., Liu, P., Li, H., & Huang, F. (2021). A comparison method for 3D laser point clouds in displacement change detection for arch dams. *ISPRS International Journal of Geo-Information*, 10(3). <https://doi.org/10.3390/ijgi10030184>
- Li, Zhenlong. (2020). Geospatial Big Data Handling with High Performance Computing: Current Approaches and Future Directions. In W. Tang & S. Wang (Eds.), *High Performance Computing for Geospatial Applications* (pp. 53–76). Springer International Publishing. https://doi.org/10.1007/978-3-030-47998-5_4
- Li, Zhiqiang, Cheng, C., Kwan, M.-P., Tong, X., & Tian, S. (2019). Identifying Asphalt Pavement Distress Using UAV LiDAR Point Cloud Data and Random Forest Classification. *ISPRS International Journal of Geo-Information*, 8(1), 39. <https://doi.org/10.3390/ijgi8010039>
- Liang, L., Zhang, Q., Song, P., Zhang, Z., Zhao, Q., Wu, H., & Cao, L. (2020). Overlapping communication and computation of GPU_CPU heterogeneous parallel spatial domain decomposition MOC method Enhanced Reader.pdf.
- Liu, K., & Boehm, J. (2015). Classification of big point cloud data using cloud computing. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(3W3), 553–557. <https://doi.org/10.5194/isprsarchives-XL-3-W3-553-2015>
- Liu, P., Yuan, T., Ma, Y., Wang, L., Liu, D., Yue, S., & Kołodziej, J. (2014). Parallel processing of massive remote sensing images in a GPU architecture. *Computing and Informatics*, 33(1), 197–217.
- Ma, C., Qi, Y., Chen, Y., Han, Y., & Ge, C. (2008). VR-GIS: An integrated platform of VR navigation and GIS analysis for city/region simulation. *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry, VRCAl 2008, January 2021*. <https://doi.org/10.1145/1477862.1477883>
- Memon, Z. A., Samad, F., Awan, Z. R., Aziz, A., & Siddiqi, S. S. (2017). CPU-GPU Processing CPU-GPU Processing. February 2018.
- Mesas-Carrascosa, F. J., de Castro, A. I., Torres-Sánchez, J., Triviño-Tarradas, P., Jiménez-Brenes, F. M., García-Ferrer, A., & López-Granados, F. (2020). Classification of 3D point clouds using color vegetation indices for precision viticulture and digitizing applications. *Remote Sensing*, 12(2). <https://doi.org/10.3390/rs12020317>
- Mineter, M. J., Dowers, S., & Gittings, B. M. (2000). Towards a HPC framework for integrated processing of geographical data: Encapsulating the complexity of parallel algorithms. *Transactions in GIS*, 4(3), 245–261. <https://doi.org/10.1111/1467-9671.00052>
- Mišić, M. J., Durđević, D. M., & Tomašević, M. V. (2012). Evolution and trends in GPU computing. *MIPRO 2012 - 35th International Convention on Information and Communication Technology, Electronics and Microelectronics - Proceedings, February 2018*, 289–294.
- Nagasaka, H., Maruyama, N., Nukada, A., Endo, T., & Matsuoka, S. (2010). Statistical power modeling of GPU kernels using performance counters. *2010 International Conference on Green Computing, Green Comp 2010*, 115–122. <https://doi.org/10.1109/GREENCOMP.2010.5598315>
- Netto, M. A. S., Calheiros, R. N., Rodrigues, E. R., Cunha, R. L. F., & Buyya, R. (2018). HPC cloud for scientific and business applications: Taxonomy, vision, and research challenges. *ACM Computing Surveys*, 51(1), 1–29. <https://doi.org/10.1145/3150224>
- Nie, B., Xue, J., Gupta, S., Patel, T., Engelmann, C., Smirni, E., & Tiwari, D. (2018). Machine Learning Models for GPU Error Prediction in a Large Scale HPC System. *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 95–106.
- Paffenholz, J., Steinhoff-knopp, B., Paffenholz, J., & Steinhoff-knopp, B. (2021). 3D Point Cloud Based Spatio-Temporal Monitoring of Natural Objects 3D Point Cloud Based Spatio-Temporal Monitoring of Natural Objects (11119) Jens-André Paffenholz , Bastian Steinhoff-Knopp (Germany) and Corinna Harmening (Austria) FIG e-Working Wee. June 2021, 21–25.
- Park, Y., & Guldman, J. (2019). Creating 3d City Models With Building Footprints And Lidar Point Cloud Classification: January. <https://doi.org/10.1016/j.compenvurbsys.2019.01.004>
- Pavilion, H., & Performance, G. S. (2019). Hyperparallel Seismic Processing.
- Pervan, B., & Knezovic, J. (2020). A survey on parallel architectures and programming models. *2020 43rd International Convention on Information, Communication and Electronic Technology, MIPRO 2020 - Proceedings, October, 999–1005*. <https://doi.org/10.23919/MIPRO48935.2020.9245341>
- Petrovič, F., Štrelák, D., Hozzová, J., Ol’ha, J., Trembecký, R., Benkner, S., & Filipovič, J. (2020). A benchmark set of highly-efficient CUDA and OpenCL kernels and its dynamic autotuning with Kernel Tuning Toolkit. *Future Generation Computer Systems*, 108, 161–177. <https://doi.org/10.1016/j.future.2020.02.069>
- Pilipczuk, O. (2020). Sustainable smart cities and energy management: The labor market perspective. *Energies*, 13(22). <https://doi.org/10.3390/en13226084>
- Poux, F., Hallot, P., Neuville, R., & Billen, R. (2016). Smart Point Cloud: Definition And Remaining Challenges. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(2W1), 119–127. <https://doi.org/10.5194/isprs-annals-IV-2-W1-119-2016>
- Puri, S., & Prasad, S. K. (2015). A parallel algorithm for clipping polygons with improved bounds and a distributed overlay processing system using MPI. *Proceedings - 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015*, 576–585. <https://doi.org/10.1109/CCGrid.2015.43>
- Qin, C.-Z. (2017). Cuda/GPU. In S. Shekhar, H. Xiong, & X. Zhou (Eds.), *Encyclopedia of GIS* (pp. 388–394). Springer International Publishing. https://doi.org/10.1007/978-3-319-17885-1_1606
- Rahmawati, F. L. (2009). Challenges in microprocessor physical and power management design. *009 International Symposium on VLSI Design, Automation and Test, 2(5)*, 255. ???
- Rajachandrasekar, R., Besson, X., & Panda, D. K. (2012).

- Monitoring and predicting hardware failures in HPC clusters with FTB-IPMI. Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2012, 1136–1143. <https://doi.org/10.1109/IPDPSW.2012.139>
- Richter, R., Discher, S., & Döllner, J. (2015). Out-of-core visualization of classified 3D point clouds. Lecture Notes in Geoinformation and Cartography, PartF3(March 2015), 227–242. https://doi.org/10.1007/978-3-319-12181-9_14
- Rocha, R., Silva, F., Rocha, R., & Dcc-fcup, F. S. (2016). Performance Metrics Parallel Computing.
- Ruan, X., & Liu, B. (2020). Review of 3D Point Cloud Data Segmentation Methods. International Journal of Advanced Network, Monitoring and Controls, 5(1), 66–71. <https://doi.org/10.21307/ijanmc-2020-010>
- Saprykin, A., Chokani, N., & Abhari, R. S. (2019). Large-scale multi-agent mobility simulations on a GPU: towards high performance and scalability. ANT/EDI40.
- Sarkar, S. (2007). The era of GPU Computing using CUDA C and JCUDA. March 2018.
- Schweizer, R. (2015). Spatial BIM Queries: A Comparison between CPU and GPU based Approaches.
- Sharma, D. K., Malikov, V., Parygin, D., Golubev, A., Lozhenitsina, A., & Sadovnikova, N. (2020). GPU-Card Performance Research in Satellite Imagery Classification Problems Using Machine Learning. Procedia Computer Science, 178(2019), 55–64. <https://doi.org/10.1016/j.procs.2020.11.007>
- Shih, N., & Wang, P. (2014). Using Point Cloud with GIS and Virtual Reality to Manage and Inspect Building Plumbing. Anzasca.Net, 174–179. http://anzasca.net/wp-content/uploads/2014/08/ANZAScA2006_Shih_Naai-Jung_and-Wang-Pin-Hun.pdf
- Singh, H., Garg, R. D., & Karnatak, H. C. (2019). Online image classification and analysis using OGC web processing service. Earth Science Informatics, 12(3), 307–317. <https://doi.org/10.1007/s12145-019-00378-z>
- Sipe, N. G., & Dale, P. (2003). understand and control malaria in Indonesia. Malaria Journal, 8, 1–9.
- Stojanovic, N., & Stojanovic, D. (2013). High-performance computing in GIS: Techniques and applications. International Journal of Reasoning-Based Intelligent Systems, 5(1), 42–49. <https://doi.org/10.1504/IJRIS.2013.055126>
- Sui, D. (2018). Geospatial Big Data. Understanding Spatial Media, April, 110–118. <https://doi.org/10.4135/9781526425850.n10>
- Surakhi, O., Khanafseh, M., & Sarhan, S. (2018). A survey on parallel multicore computing: Performance & improvement. Advances in Science, Technology and Engineering Systems, 3(3), 152–160. <https://doi.org/10.25046/aj030321>
- Thoman, P., Kofler, K., Studt, H., Thomson, J., & Fahringer, T. (2011). Automatic OpenCL device characterization: Guiding optimized kernel design. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6853 LNCS(PART 2), 438–452. https://doi.org/10.1007/978-3-642-23397-5_43
- Tzeng, S., Patney, A., & Owens, J. D. (2012). GPU Task-Parallelism : Primitives and Applications. GTC 2012.
- Uy, M. A., Pham, Q. H., Hua, B. S., Nguyen, T., & Yeung, S. K. (2019). Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. Proceedings of the IEEE International Conference on Computer Vision, 2019-October, 1588–1597. <https://doi.org/10.1109/ICCV.2019.00167>
- van Werkhoven, B. (2019). Kernel Tuner: A search-optimizing GPU code auto-tuner. Future Generation Computer Systems, 90, 347–358. <https://doi.org/10.1016/j.future.2018.08.004>
- Venugopal, V., & Kannan, S. (2013). Accelerating real-time LiDAR data processing using GPUs. Midwest Symposium on Circuits and Systems, March, 1168–1171. <https://doi.org/10.1109/MWSCAS.2013.6674861>
- Vu, V. A., & Tan, G. (2019). A Framework for Mesoscopic Traffic Simulation in GPU. IEEE Transactions on Parallel and Distributed Systems, 30, 1691–1703.
- Wang, C., Hu, F., Sha, D., & Han, X. (2013). Efficient lidar point cloud data managing and processing in a Hadoop-based distributed framework. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4(4W2), 121–124. <https://doi.org/10.5194/isprs-annals-IV-4-W2-121-2017>
- Wang, H., Guan, X., & Wu, H. (2017). A hybrid parallel spatial interpolation algorithm for massive LiDAR point clouds on heterogeneous CPU-GPU systems. ISPRS International Journal of Geo-Information, 6(11). <https://doi.org/10.3390/ijgi6110363>
- Wang, S., & Armstrong, M. P. (2003). A quadtree approach to domain decomposition for spatial interpolation in Grid computing environments. 29, 1481–1504. <https://doi.org/10.1016/j.parco.2003.04.003>
- Yang, H., Chen, W., Qian, T., Shen, D., & Wang, J. (2015). The extraction of vegetation points from LiDAR using 3D fractal dimension analyses. Remote Sensing, 7(8), 10815–10831. <https://doi.org/10.3390/rs70810815>
- Yao, X., & Li, G. (2018). Big spatial vector data management: a review. Big Earth Data, 2(1), 108–129. <https://doi.org/10.1080/20964471.2018.1432115>
- Yin, Q., Wu, Y., Zhang, F., & Zhou, Y. (2020). GPU-based soil parameter parallel inversion for PolSAR data. Remote Sensing, 12(3). <https://doi.org/10.3390/rs12030415>
- Yusuf, A., & Alawneh, S. (2018). A Survey of GPU Implementations for Hyperspectral Image Classification in Remote Sensing. Canadian Journal of Remote Sensing, 44(5), 532–550. <https://doi.org/10.1080/07038992.2018.1559725>
- Zhang, J., & York, N. (2019). Efficient Quadtree Construction for Indexing Large-Scale Point Data on GPUs : Bottom-Up vs . Top-Down. Adms.
- Zhang, J., & You, S. (2012). CudaGIS: Report on the design and realization of a massive data parallel GIS on GPUs. Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS 2012, c, 101–108. <https://doi.org/10.1145/2442968.2442981>

Zhang, J., You, S., & Gruenwald, L. (2019). Large-Scale Spatial Data Processing on GPUs and GPU-Accelerated Clusters. 27–34.

Zhang, Y., Hu, Y., Li, B., & Peng, L. (2011). Performance and power analysis of ATI GPU: A statistical approach. Proceedings - 6th IEEE International Conference on Networking, Architecture, and Storage, NAS 2011, 149–158. <https://doi.org/10.1109/NAS.2011.51>

Zhao, L., Chen, L., Ranjan, R., Choo, K. K. R., & He, J. (2016). Geographical information system parallelization for spatial big data processing: a review. Cluster Computing, 19(1), 139–152. <https://doi.org/10.1007/s10586-015-0512-2>

Zhou, Y., Wang, S., & Guan, Y. (2019). An efficient parallel algorithm for polygons overlay analysis. Applied Sciences (Switzerland), 9(22), 1–19. <https://doi.org/10.3390/app9224857>

Zhu, J., & Wu, P. (2021). Towards effective bim/gis data integration for smart city by integrating computer graphics technique. Remote Sensing, 13(10). <https://doi.org/10.3390/rs13101889>