ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-4/W2, 2017
2nd International Symposium on Spatiotemporal Computing 2017, 7–9 August, Cambridge, USA

ELASTIC CLOUD COMPUTING ARCHITECTURE AND SYSTEM FOR HETEROGENEOUS SPATIOTEMPORAL COMPUTING

Xuan Shi*

Department of Geosciences, University of Arkansas, Fayetteville, AR 72701. U.S.A. Email: xuanshi@uark.edu

KEY WORDS: Elastic Architecture, Cloud Computing, Geocomputation, Spatiotemporal

ABSTRACT:

Spatiotemporal computation implements a variety of different algorithms. When big data are involved, desktop computer or standalone application may not be able to complete the computation task due to limited memory and computing power. Now that a variety of hardware accelerators and computing platforms are available to improve the performance of geocomputation, different algorithms may have different behavior on different computing infrastructure and platforms. Some are perfect for implementation on a cluster of graphics processing units (GPUs), while GPUs may not be useful on certain kind of spatiotemporal computation. This is the same situation in utilizing a cluster of Intel's many-integrated-core (MIC) or Xeon Phi, as well as Hadoop or Spark platforms, to handle big spatiotemporal data. Furthermore, considering the energy efficiency requirement in general computation, Field Programmable Gate Array (FPGA) may be a better solution for better energy efficiency when the performance of computation could be similar or better than GPUs and MICs. It is expected that an elastic cloud computing architecture and system that integrates all of GPUs, MICs, and FPGAs could be developed and deployed to support spatiotemporal computing over heterogeneous data types and computational problems.

1. SPATIOTEMPORAL COMPUTING IN THE ERA OF BIG DATA SCIENCE

Data originated from sensors aboard satellites and platforms such as airplane, UAV and mobile systems have generated high spectral, spatial, vertical and temporal resolution data. There is greater potential and challenge to extract more accurate and significant geospatial information with these high resolution data at the level that was not possible before. For example, high spectral resolution (hyperspectral) data characterized by hundreds of relatively narrow (<10 nm) and contiguous bands are useful not only for the general purpose land cover classification, but also for extracting the mineral composition of rocks and identifying crop or tree species (Martin et al. 1998, Xiao et al. 2004, Thenkabail et al. 2004, Clark et al. 2005, Buddenbaum et al. 2006, Boschetti et al. 2007). High spatial resolution (HSR) sensors (Greenberg et al. 2009, Sridharan and Qiu 2013), such as IKONOS, QuickBird, and WorldView-2,3, can provide sub-meter pixel image products, with sufficient detail to allow the delineation of individual geographic objects such as buildings, trees, roads, and grassland (often referred to as feature extraction). Light Detection and Ranging (LiDAR) sendor can offer high vertical resolution of geometry and allows the direct collection of x, y, and z coordinates of ground objects, which makes possible automatic detection of elevated features and construction of 3 dimensional (3D) models of ground surface (Rottensteiner et al. 2005). The temporal resolution are increased not only because of the shortening of revisit frequency, but the simultaneous available of mutiple sensors such as IKNOS, WolderView 1,2,and 3. Traditional statistics based per pixel image processing methods and algorithms have been designed primarily for 2D coarse and moderate spatial resolution multispectral imagery, not appropriate or optimal for the high resolution sensor data.

Inevitably, the volume, velocity, and variety of hyperspectral, HSR and 3D data, along with other socioeconomic, demographic, environmental, and social media data, pose great challenge to existing geospatial software when analyzing such big heterogeneous datasets, because the scale of data and computation are well beyond the capacity of PC-based software due to PC's limited storage, memory, and computing power. For example, DigitalGlobal alone acquires 1 billion km^2 of HSR imagery annually with its 6-satellite constellation, resulting in 63 petabypes data achive in 2013, second only to Facebook among all private companies. *New computing infrastructure and system are required in response to such big data challenge*.

2. HYBRID COMPUTING ARCHITECTURE AND SYSTEMS

Heterogeneous spatial data integration, processing, and analysis have been a challenging task as spatial computation has been playing an essential part in a variety of significant areas of science, engineering and decision-making. Such areas include geospatial-related natural and social sciences, public safety and emergency response, spatial intelligence analytics and military operations, ecological and environmental science and engineering, and public health. Geospatial data represents real-world geographic features or objects in either vector or raster data models. In the vector model, features are captured as discrete geometric objects and represented as points, lines or polygons with non-spatial attributes. In the raster model, features are represented on a grid, or as a multidimensional matrix, including satellite imagery and other remotely sensed data. While many algorithms were developed to process vector, raster or imagery data, handling heterogeneous data is required in a variety of spatiotemporal computing tasks.

High-resolution geospatial data has become increasingly available along with an accelerating increase in data volume. Consequently spatial data may be stored as separate tiles for efficient data exchange, integration, processing, analysis and visualization. Distributed or separated data has raised a variety of challenges and complexities in spatial computation for knowledge discovery. Traditional software may only work on one tile of data for computation or analysis. If all tiles are merged or mosaicked into a single piece, its size may become

*Corresponding author

too huge to be processed on a desktop computer. If the whole data is maintained as separate tiles, however, the analytic result may not be correct or consistent among multiple separated tiles.

In the problem-solving and decision-making processes, the performance of geospatial computation is severely limited when massive datasets are processed. *Heterogeneous geospatial data integration and analytics obviously magnify the complexity and operational time frame.* Many large-scale geospatial problems may be not processable at all if the computer system does not have sufficient memory or computational power.

Designing novel algorithms and deploying the solutions in massively parallel computing environment to achieve the capability for scalable data processing and analytics over peta-scale, complex, and heterogeneous spatial data with consistent quality and high-performance is the central theme of spatiotemporal computing. Emerging computer architecture and system that combine multi-core CPUs and accelerator technologies, like many-core GPUs and Intel MIC coprocessors, would provide substantial computing power for many time-consuming spatial-temporal computation and applications. New multi-core architectures combined with application accelerators hold the promise to achieve scalability and high performance by exploiting task and data levels of parallelism that are not supported by the conventional systems. Such a distributed computing environment is particularly suitable for large-scale spatiotemporal computation over distributed or separated spatial data, while the potential of such advanced Cyberinfrastructure remains unexplored in this domain. For example, ESRI's commercial product ArcGIS has an estimated 615 stand-alone tools in 18 categories (Gao and Goodchild 2013), plus hundreds of other interactive tools under the ArcMap interface. Very few of these more than 1,000 functional modules in ArcGIS are parallelized into applicable solutions on GPUs or MICs.

In the chapters on "*GPGPU in GIS*" (Shi and Huang 2016a) and "*MIC in GIS*" (Shi and Huang 2016b) included in the *Encyclopaedia of GIS*, details about GPU and MIC were introduced. As representative accelerators, GPUs and MICs have been utilized to construct hybrid computer architecture and systems that integrate multicore processors, manycore coprocessors. For example, Keeneland is a hybrid supercomputer that has 528 CPUs and 792 GPUs funded by NSF. The Keeneland Full Scale (KFS) system consists of 264 HP SL250G8 compute nodes, each with 2 eight-core Intel Sandy Bridge (Xeon E5) processors, 3 NVIDIA M2090 GPU accelerators. Beacon is a Cray CS300-AC Cluster Supercomputer that offers access to 48 compute nodes and 6 I/O nodes joined by FDR InfiniBand interconnect. Each compute node is equipped with 2 Intel Xeon 8-core processors, 4 Intel Xeon Phi (MIC) coprocessors, 256 GB of RAM, and 960 GB of storage. Each Xeon Phi coprocessor contains 60 MIC cores and 8 GB on-board memory. Beacon provides 768 conventional cores and 11,520 accelerator cores that offer over 210 TFLOP/s of combined computational performance, 12 TB of system memory, 1.5 TB of coprocessor memory, and over 73 TB of storage, in aggregate. Both Keeneland and Beacon were used to enable many large scale spatiotemporal computation and simulation (Shi and Xue 2016a, 2016b, Lai et al. 2016, Guan et al. 2016, Shi et al. 2014a, Shi et al. 2014b).

Besides conventional hybrid computer clusters, Hadoop and Spark are the new distributed computing architecture and platforms that can be applied for spatiotemporal computing. Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage. Hadoop MapReduce is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner. MapReduce refers to the following two different tasks that Hadoop programs perform: 1) map is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs). 2) reduce takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

In comparison to Hadoop that has heavy I/O and file transactions, Apache Spark is a new distributed computing platform for general-purpose scientific computation. Spark extends the popular MapReduce model to efficiently support more types of computations, including interactive queries and stream processing. Furthermore, Spark provides ability to run computations in memory, and the system is more efficient than MapReduce for complex applications running on disk. Spark is designed to cover a wide range of workloads applicable on distributed systems, including batch applications, iterative algorithms, interactive queries, streaming and graph based network calculation through its rich built-in libraries, including Spark SQL, streaming, MLlib and GraphX. By supporting these workloads in the distributed engine, Spark makes it easy and inexpensive to combine different processing types, which is often necessary in production data analysis pipelines. Spark is also highly accessible, offering simple APIs in Python, Java, Scala, and SQL. Very few of these more than 1,000 functional modules in ArcGIS are transformed into MapReduce solutions applicable on Hadoop or Spark platforms.

At last, an emerging new hybrid architecture integrates FPGAs as coprocessors along with traditional multicore or manycore processors. FPGA has the advantage for its energy-efficient manner, while it can be reconfigured based on different applications. There are growing interest to deploy FPGAs in cloud computing and data center systems. For example, in Microsoft's Catapult project, FPGAs were integrated into a data center to accelerate the Bing search engine. By using FPGAs to implement the document ranking algorithm, the performance was doubled at only a 30% increase in energy (Putnam et al. 2015). Intel acquired Altera (one major provider of FPGAs) for 16.7 billion dollars as Intel also would like to integrate FPGAs with Xeon multicore processors to build data and computing centers (Morgan 2014). FPGAs have the potential to build energy efficient next generation computing systems, if several challenging problems can be resolved to enable reconfigurable computing to become mainstream solutions.

3. HETEROGENEOUS SPATIOTEMPORAL COMPUTING ON HYBRID COMPUTING SYSTEMS

In recent years, my research team has been working on parallel and high performance spatiotemporal computation on supercomputers KraKen (a cluster of CPUs), Keeneland (a cluster of GPUs), and Beacon (a cluster of MICs) over different spatiotemporal problems, including ISODATA for unsupervised image classification, MLC for supervised image classification, Kriging interpolation, Cellular Automata based urban sprawl

simulation, agent-based modelling on information diffusion, affinity propagation (AP), near-repeat calculation, and so on. In 2016, the 2nd place award was given by ACM SIGSPATIAL International Student Research Competition for the research entitled *Accelerating the Calculation of Minimum Set of Viewpoints for Maximum Coverage over Digital Elevation Model Data by Hybrid Computer Architecture and Systems*. This research introduces how to accelerate the calculation of minimum set of viewpoints for maximum coverage over digital elevation model data using Intel's Xeon Phi on supercomputer Beacon equipped with Intel's Many-Integrated-Core (MIC) coprocessors. This data and computation intensive process consists of a series of geospatial computation tasks, including 1) the automatic generation of control viewpoints through map algebra calculation and hydrological modeling approaches; 2) the creation of the joint viewshed derived from the viewshed of all viewpoints to establish the maximum viewshed coverage of the given digital elevation model (DEM) data; and 3) the identification of a minimum set of viewpoints that cover the maximum terrain area of the joint viewshed. The parallel implementation on Beacon achieved more than 100x speedup in comparison to the sequential implementation. The outcome of the computation has broad societal impacts since the solutions can be applied to real-world applications and decision-making practice, including civil engineering, infrastructure optimization and management, and military operations.

Throughout the course of such extensive development on different spatiotemporal themes, it can be concluded that different kind of hybrid computer architecture and systems may be more suitable to different kind of problems. GPU has been utilized in a variety of our prior works (Shi and Xue 2016b, Guan et al. 2016, Shi et al. 2014, Shi et al. 2014). Particularly when many spatiotemporal modules would process matrix data, GPU could be a perfect accelerator to improve the performance. On the other hand, some spatiotemporal computation and simulation would contain randomized procedures, which may devaluate the functionality of GPU. A typical CUDA-capable GPU is organized into an array of highly threaded streaming multiprocessors (SMs). Within the SM, computing threads are grouped into block, which is then managed by a grid structure. Within a block of threads, the threads are executed in groups of 32 called a warp. In the case of the random procedures, if different threads in a warp need to do different things, all threads will compute a logical predicate and several predicated instructions. This is called warp divergence. When all threads execute conditional branches differently, the execution cost could be the sum of both branches. Warp divergence can lead to a big loss of parallel efficiency. In comparison to such a problem in the utilization of GPUs, MIC exemplifies its advantage in dealing with randomized procedures in spatiotemporal computation. Each MIC has a total of 240 threads on 60 processing cores. The vector processing unit on the Intel MIC processor is very efficient and can deal with operations involving many independent operands.

Although hardware accelerators have been widely used in computer clusters, there are limitations as well. For example, the extra communication overhead between the coprocessor and the host can easily offset the performance benefit from the parallel processing on GPUs. Originally, all the kernels running on GPUs have to be created by host processors. This limitation makes it impossible or inconvenient to implement some algorithms with dynamic behavior. The communication cost between the host processors and the coprocessors has been a traditional drawback since the on-board memory on the accelerators is separate from the host main memory. Therefore, the source data and the results have to be transferred between these two pieces of memory. Data transfer between two GPUs crossing two different nodes has to go through host memory, introducing unnecessary overhead. NVIDIA Kepler GPU architecture supports a "dynamic parallelism" feature in which kernels (i.e., tasks) can generate new kernels while running on GPU. This feature makes it possible to implement algorithms with dynamic behavior completely on the GPU. Furthermore, the "GPUDirect" feature could be explored to carry out data communication directly when multiple GPUs are used.

In the case of MICs, both the native model and the offload model only utilize the Xeon Phi coprocessor, while the host (Xeon) CPU is not efficiently used, or not used, in the calculation. Hybrid solutions were explored to optimally utilize both the Xeon CPU and MIC coprocessors (Lai et al. 2016). For example, to extend the native model, we create MPI ranks that reside on the host CPU and the MIC coprocessors. If m MIC (Xeon Phi) coprocessors and n host CPU processors are used, m × 60 + n MPI processes are created in the parallel implementation. In the case of offload model, the workload is first distributed to CPUs through MPI. Then a host CPU will offload part of the job to a MIC card using OpenMP. On the host CPU, we also use OpenMP to spawn multiple threads for parallel processing. Asynchronous offload allows overlap of data transfer and compute. The host initiates an offload to be performed asynchronously and can proceed to next statement after starting this computation. In general, the hybrid-offload solution could be more flexible and efficient (Lai et al. 2016).

Although FPGAs have been increasingly integrated into cloud computing systems and data centers, very few geographers and GIScientists would have worked on FPGAs. In a pilot study, we implemented a generic Cellular Automata (CA) algorithm using a cluster of FPGAs. In (Shi et al. 2014a), in comparison to the use cases of embarrassingly parallelism and geospatial computing with loose communication and data exchange, CA was the most complicated use cases due to its intensive data partition and exchange among distributed computing nodes. In (Shi et al. 2014a), we applied MPI's SEND and RECV commands to enable data communication and exchange to complete Game of Life (GOL). In the latest pilot study, we re-designed the solution by transforming the GOL into a pipeline style that is applicable on a cluster of FPGAs.

#	Size of the grid: 8,192 x 8,192				Size of the grid: 16,384 x 16,384			
	MPI + CPU	MPI + GPU	MPI + MIC	FPGA	MPI + CPU	MPI + GPU	MPI + MIC	FPGA
1				38.9				287.4
2	78.2	24.9	7.3	19.6	312.7	122.2	29.0	143.6
4	39.2	12.8	4.3	9.9	155.6	59.1	14.7	42.3
8	21.8	6.3	3.3	5.5	78.1	29.7	8.5	21.8
16	10.4	4.2	3.8	3.2	39.4	17.2	6.6	11.7

Table 1. Scalability and performance (by second) comparison for GOL computation on clusters of CPUs, GPUs, MICs, and FPGAs

Table 1 displays the result of scalability and performance comparison for GOL computation using clusters of CPUs, GPUs, MICs, and FPGAs. The size of GOL is 8,192 x 8,192 and 16,384 x 16,384. In this pilot study, 100 iterations were implemented over different numbers (#) of processors. In all cases, FPGA displays superiority than CPU and GPU. In the case of MIC, each MIC utilized 60 cores. When the size of GOL is 8,192 x 8,192, FPGA achieved a better performance than MIC. Based on the trajectory of time used on GOL simulation when the size is 16,384 x 16,384, it is expected

when 32 or more FPGAs are used, the performance of FPGA could be better than that for the same number of MICs. When energy efficiency is considered, FPGA is superior in all cases. Considering many spatiotemporal computation are raster based while such convolutional calculation is commonly applied, this pilot study on clusters of FPGAs will have broader impact in the future in the GIScience community.

Partially sponsored by NSF SMA-1416509, we also developed pilot study to utilize *Spark* cloud computing platform for geocomputation and data mining over big geospatial and social media Data. Geospatial computation is special because both spatial geometry and non-spatial attributes have to be processed and analyzed. Hadoop/Spark is specifically advantageous in handling text data. In the pilot study to complete geocoding procedures based on twitter user profile, both the source data (i.e. hundreds of tweets) and the matching data (e.g. Gazetteer data that have 11 millions of records) will generate huge amount of comparison. Even if conventional parallel computing solution could be developed to resolve this problem, implementing Scala script on Spark could be the most efficient approach to complete the task. Figure 1 displays the workflow and algorithm to complete the geocoding procedure based on twitter user profile and gazetteer data.

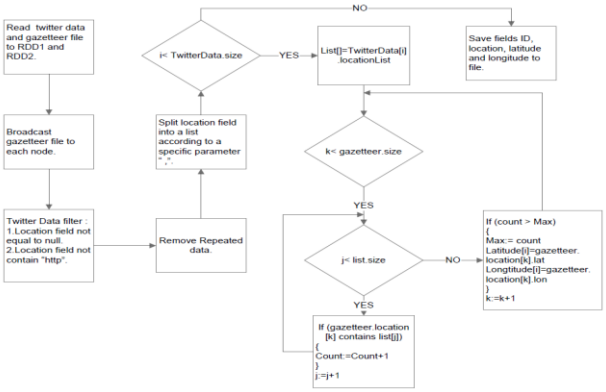

Figure 1. Workflow and algorithm to complete the geocoding procedure based on twitter user profile and gazetteer data

4. VISION AND CONCLUSION

Spatiotemporal computing would have to deal with different types of data and varied algorithms. As exemplified in the prior works, different kind of hardware infrastructure could be more appropriate and efficient to a certain kind of spatiotemporal computing problems. Existing supercomputing infrastructure, however, may have been equipped with a major type of hardware, such as a cluster of MICs or GPUs. This means such kind of computing infrastructure may not be flexible or elastic to cope with the needs of spatiotemporal computing.

Cloud computing, on the other hand, has one key concept of Infrastructure as a service (IaaS) that supports large numbers of virtual machines to imitate dedicated hardware. Within the virtual environment, it is possible for researchers to select the most appropriate and efficient processors and accelerators to complete the spatiotemporal computing tasks. It is expected that an elastic cloud computing architecture and system could integrate CPUs, GPUs, MICs, and FPGAs in response to the needs for heterogeneous spatiotemporal computing. Such an elastic computing infrastructure will be fundamental to the next generation of geographic information system and science in the era of big data science.

ACKNOWLEDGEMENT

This research was partially supported by the National Science Foundation (NSF) through NSF SMA-1416509.

REFERENCES

Boschetti, M., Boschetti, L., Oliveri, S., Casati, L. and Canova, I., 2007. Tree species mapping with Airborne hyper-spectral MIVIS data: the Ticino Park study case. International Journal of Remote Sensing, 28(6): 1251-1261.

Buddenbaum, H., Schlerf, M. and Hill, J., 2005. Classification of coniferous tree species and age classes using hyperspectral data and geostatistical methods. International Journal of Remote Sensing, 26(24): 5453-5465.

Clark, M.L., Roberts, D.A. and Clark, D. B., 2005. Hyperspectral discrimination of tropical rain forest tree species at leaf to crown scales. Remote Sensing of Environment, 96(3): 375-398.

Gao, S. and Goodchild, M. F., 2013. Asking spatial questions to identify GIS functionality. in Proc. Fourth International Conference on Computing for Geospatial Research and Application, Jul. 2013, pp. 106–110.

Greenberg, J., Dobrowski, S., Vanderbilt, V. 2009. Limitations on maximum tree density using HSR remote sensing and environmental gradient analysis. Remote Sensing of Environment 2009, 113, 94-101.

Guan, Q., Shi, X., Huang, M., and Lai, C., 2016. A hybrid parallel cellular automata model for urban growth simulation over GPU/CPU heterogeneous architectures. International Journal of Geographical Information Science. Volume 30, Issue 3. pp. 494-514. DOI: 10.1080/13658816.2015.1039538

Lai, C., Huang, M. and Shi, X., 2016. SRC: Accelerating the Calculation of Minimum Set of Viewpoints for Maximum Coverage over Digital Elevation Model Data by Hybrid Computer Architecture and Systems. Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. San Francisco, CA.

Martin, M., Newman, S., Aber, J. and Congalton, R., 1998. Determining forest species composition using high spectral resolution remote sensing data. Remote Sensing of Environment, 65(3): 249-254.

Morgan, T., 2014. Intel Mates FPGA with Future Xeon Server Chip. https://www.enterprisetech.com/2014/06/18/intel-mates-fpga-future-xeon-server-chip/

Putnam, A., Caulfield, A. M., Chung, E. S., Chiou, D., Constantinides, K., Demme, J., Esmaeilzadeh, H., Fowers, J., Gopal, G. P., Gray, J., Haselman, M., Hauck, S., Heil, S., Hormati, A., Kim, J. Y., Lanka, S., Larus, J., Peterson, E., Pope, S., Smith, A., Thong, J., Xiao, P. Y., and Burger, D., 2015. A reconfigurable fabric for accelerating large-scale data center services. IEEE Micro, 35(3):10–22, May 2015.

Rottensteiner, F.; Trinder, J.; Clode, S.; Kubik, K. 2005. Using the dempster–shafer method for the fusion of lidar data and

multi-spectral images for building detection. Information Fusion 2005, 6, 283-300.

Sridharan, H. and Qiu, F., 2013. Developing an Object-based HSR Image Classifier with a Case Study Using WorldView-2 Data. Photogrammetric Engineering & Remote Sensing, 79(11).

Shi, X. and Huang, M., 2016a. GPGPU in GIS. In Shekhar, S. et al. (eds) Encyclopedia of GIS. Published by Springer

Shi, X. and Huang, M., 2016b. MIC in GIS. In Shekhar, S. et al. (eds) Encyclopedia of GIS. Published by Springer.

Shi, X. and Xue, B., 2016a. Deriving a Minimum Set of Viewpoints for Maximum Coverage over Any Given Digital Elevation Model Data. International Journal of Digital Earth. Volume 9, Issue 12. pp. 1153-1167.

Shi, X. and Xue, B., 2016b. Parallelizing maximum likelihood classification on computer cluster and graphics processing unit for supervised image classification. International Journal of Digital Earth. http://www.tandfonline.com/doi/full/10.1080/17538947.2016.1251502

Shi, X., Lai, C., Huang, M. and You, H., 2014a. Geocomputation over the Emerging Heterogeneous Computing Infrastructure. Transactions in GIS, vol. 18, no. S1, pp. 3-24, Nov. 2014.

Shi, X., Huang, M., You, H., Lai, C. and Chen, Z., 2014b. Unsupervised image classification over supercomputers Kraken, Keeneland and Beacon. GIScience & Remote Sensing, Volume 51, Issue 3. 2014. pp. 321-338

Shi, X. and Ye, F., 2013. Kriging interpolation over heterogeneous computer architectures and systems. GIScience & Remote Sensing. Volume 50, Issue 2, 2013. pp.196-211

Thenkabail, P. S., Enclona, E. A., Ashton, M. S., Legg, C. and De Dieu, M. J., 2004. Hyperion, IKONOS, ALI, and ETM+ sensors in the study of African rainforests. Remote Sensing of Environment, 90(1): 23-43.

Xiao, Q., Ustin, S. and McPherson, E., 2004. Using AVIRIS data and multiple-masking techniques to map urban forest tree species. International Journal of Remote Sensing, 25(24): 5637-5654.