Dense DSM Generation Using the GPU

KIRILL ROTENBERG, LOUIS SIMARD, PHILIPPE SIMARD, Montreal

ABSTRACT

Generating dense digital surface models (DSM) is a vital step in all photogrammetric work. Its precise production influences the quality of subsequent products including the digital terrain model (DTM) and orthomosaic. SimActive's proven DSM capability is employed throughout the industry for its precision and high processing speed. With today's demands for larger and denser DSMs, a considerable strain is placed on the performance. This is driving the need for improved processing speeds for dense DSM generation. The problem has been tackled with new algorithms and the use of multicore CPUs, but have had little success on the large commercial production scale. This paper discusses the suitability of the GPU to resolve the performance dilemma with dense DSM generation. GPU computing will be discussed, followed by a careful look at the challenges involved and the benefits provided. Finally, sample results will be presented and analyzed.

1. INTRODUCTION

In the last years, SimActive has continuously innovated in the field of photogrammetry software. The company has developed a full photogrammetry suite, Correlator3D™, which allows automated production from any aerial or satellite sensor, including UAV platforms. The workflow comprises of automatic modules for aerial triangulation (AT), DSM generation, DTM extraction, orthorectification and mosaic creation as well as manual editing tools as shown in Figure 1.

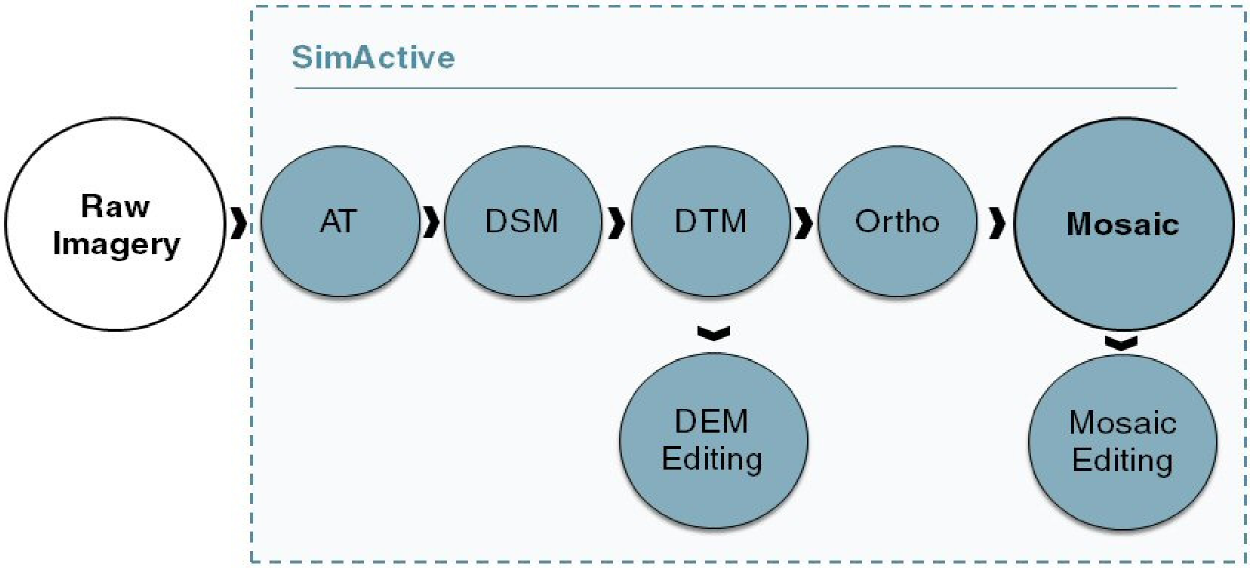

Figure 1: SimActive workflow.

SimActive's innovation is present in its various technological advancements and industry firsts, starting with the development of novel computer vision algorithms and the introduction of the GPU powered DSM in 2008 – the first of its kind. In 2011, SimActive released unique automated seamline algorithms, then in 2012 mosaic fragmentation and new dense elevation algorithms for DSM and finally, a new and unique aerial triangulation in 2013. The AT represented another first in the industry, implementing a GPU solution for truly rapid processing.

The requirement for generating dense DSMs from high resolution images has rapidly grown within the industry. The trend was first observed in 2010 and has picked up steam ever since. It must be noted however that the advantage of a much denser model must be traded off with the encumbrance such heavy data brings to the workflow. Oftentimes, the most optimal solution is a balance between data size and density. Hence, Correlator3D™ was designed to handle just such a balance processing of thousands of images efficiently, while producing accurate elevation values with a very high point density. Being the first commercial product to profit from the graphics processing units (GPUs) to accelerate DSM production back in 2008, SimActive has had the time advantage to capitalize on the GPU effort. Since 2008, SimActive has been actively building upon its GPU expertise, optimizing code for its subsequent DSM releases. Now in its third generation, SimActive's DSM module utilizes the most advanced GPU processing techniques for the DSM, staying well ahead of the design curve and providing superior performance as shown in Figure 2.

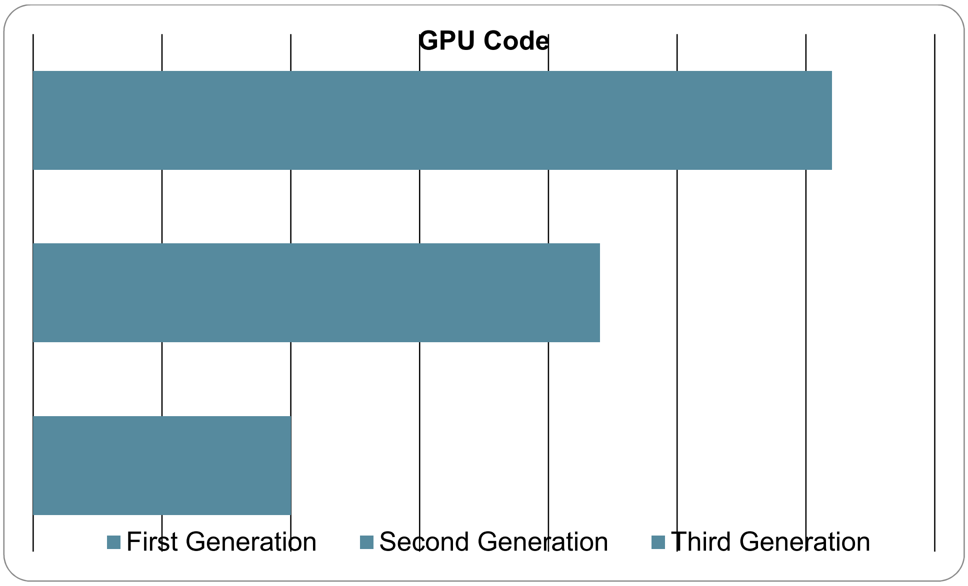

Figure 2: SimActive's relative GPU code performance based on code generation.

2. GPU COMPUTING

GPU computing is the use of a GPU (graphics processing unit) together with a CPU to accelerate general-purpose scientific and engineering applications. GPU computing has become popular and is used in many industries. Photogrammetry is a good candidate for general-purpose graphics processing unit (GPGPU) implementation since it uses image processing, which the GPU is perfectly suited for processing. GPGPU computing offers unprecedented application performance by offloading compute-intensive portions of the application to the GPU, while the remainder of the code still runs on the CPU. The massively parallel architecture of the GPU accelerates processing times. From a user's perspective, applications simply run significantly faster.

CPU + GPU is a powerful combination because CPUs consist of a few cores optimized for serial processing, while GPUs consist of thousands of smaller, more efficient cores designed for parallel performance as shown in Figure 3. Serial portions of the code run on the CPU while parallel portions run on the GPU.

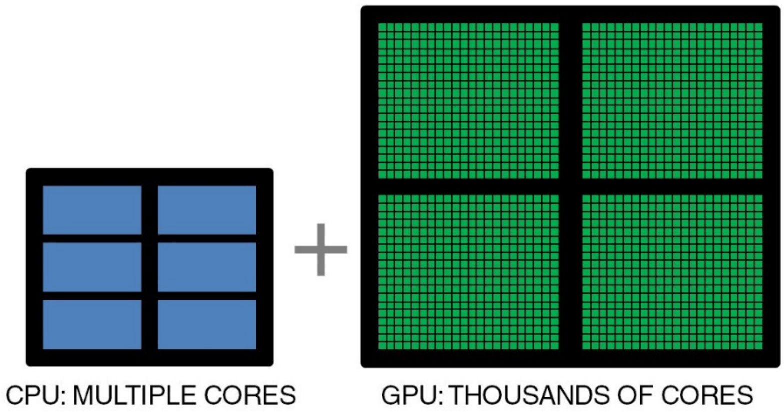

Figure 3: A comparison of processing cores.

It may be tempting to assume that using the GPU will necessarily lead to significant gain in processing times compared to CPU alone. GPU programming is notoriously difficult and requires significant investment to produce a desired outcome.

Programming GPUs is a cumbersome task for two primary reasons: tedious performance optimizations and lack of portability. First, optimizing an algorithm for a specific GPU is a time-consuming task that requires a thorough understanding of both the algorithm and the underlying hardware. Unoptimized code typically only achieves a small fraction of the peak GPU performance. Second, GPU code lacks efficient portability as code written for one GPU can be inefficient when executed on another. Moving code from one GPU to another while maintaining the desired performance is a non-trivial task, often requiring significant modifications to account for the hardware differences. Some algorithms are so complex that it may be near impossible to code using the GPU.

Once it is determined that all necessary algorithms can indeed be implemented on the GPU, the common approach is to write GPU specific code with low level GPU APIs. Although this approach can achieve very good performance, it raises serious portability issues: programmers are required to write a specific version of code for each potential target architecture. It results in high development and maintenance costs.

Furthermore, it is not guaranteed that the application will achieve a higher performance after GPU implementation. It is highly important to manage the CPU, GPU, memory systems and I/O information properly to minimize overhead and optimize operation. Moving large amounts of data (imagery) across a system inefficiently as it communicates between CPU, GPU and memory will eliminate any speed improvements gained using the faster GPU processor. Therefore, it is very important to not only write portable GPU code, but to also intelligently manage the operation of the entire system to truly benefit from the GPU's massively parallel architecture. Figure 4 demonstrates the superior computational power of a GPU compared to a CPU for parallel operations.

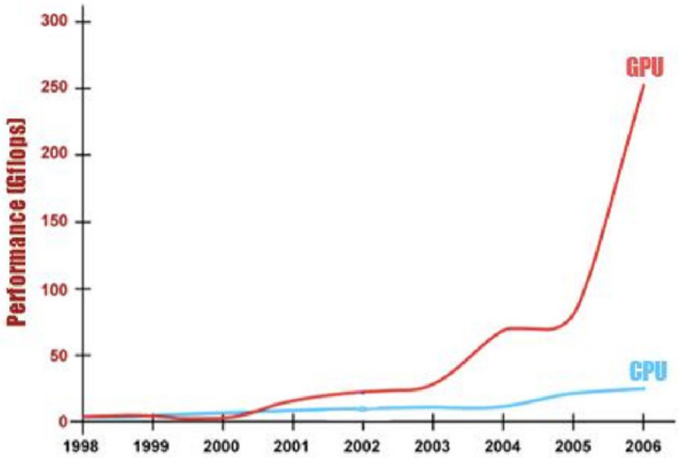

Figure 4: A comparison of the GPU and CPU computing potential.

3. CHALLENGES

The traditional approach for DSM generation is to determine corresponding feature points in image pairs. Then, elevation information is derived by triangulation. These elevation values are finally interpolated to generate a DSM arranged along a regular grid. This technique can be described as a bottom-up approach as image measurements are used to derive a solution, in this case a DSM. The process is depicted in Figure 5.

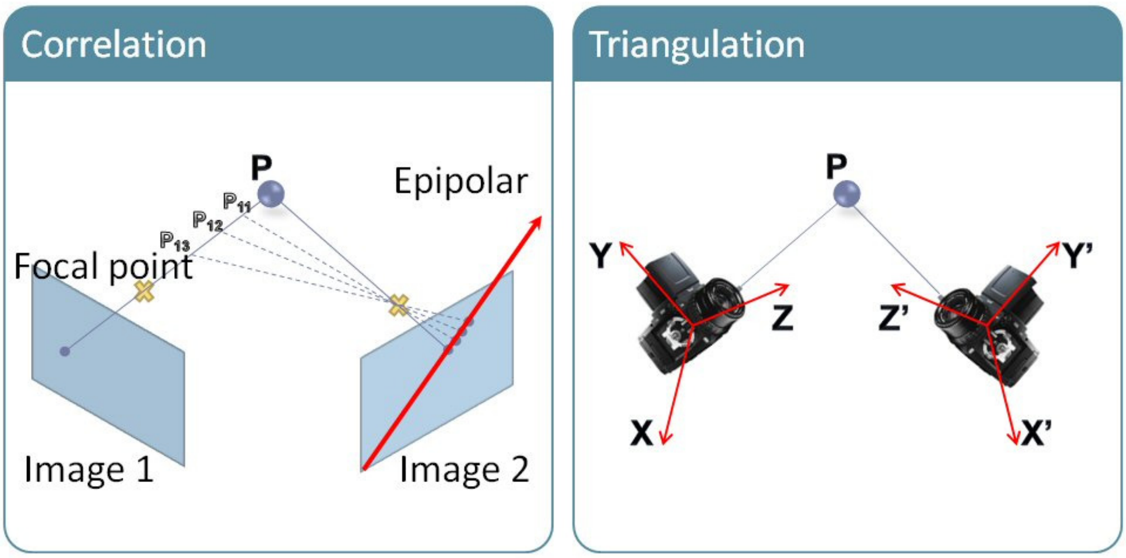

Figure 5: An illustration of correlation and triangulation.

If a dense DSM needs to be generated, then its grid post spacing will approach the ground sampling distance (GSD) of the input imagery. The challenges associated with such a task are multiple compared to generating a DSM at a lower horizontal resolution. First, the correlation process is much more difficult since there is less information available at high spatial resolutions in the imagery. Second, the process becomes much more sensitive to the quality of input data. For example, small radiometry differences, uncertainties in camera calibration as well as quality of exterior orientation

data will render correlation harder. Finally, increasing the grid post spacing of the required DSM will increase the computational time required exponentially.
Introducing the GPU further complicates the problem and adds unique challenges to efficient dense DSM generation. Due to the design of the GPU, all points must be processed at the same time. This adds unique challenges to the fundamental steps of DSM generation: correlation and triangulation. The algorithm can no longer simply take points of interest and determine the elevation on a point by point basis. Every pixel must be assigned a unique elevation value, yet be processed at the same time. Hence, a methodology must be implemented to handle the assignment of individual elevation values to pixels on the GPU, which can only assign a single value per iteration since all pixels must be processed at once. As the DSM approaches the GSD of the imagery, there are more pixels with a higher chance of varying elevations increasing the problem complexity.

Furthermore, as the DSM approaches the GSD of the imagery, each pixel has increasingly more importance in the final result. Therefore, lower image quality will have a much greater impact on the DSM. Moreover, as the amount of pixel information decreases to obtain a more accurate result, there is less filtering, which contributes to a poorer signal-to-noise ratio. If the camera and EO are not perfectly calibrated, there is a large chance that the window in which correlation occurs will not be the same between images. Hence, a comparison between the same image section will never represent the same feature, making an exact match impossible.

Another import aspect is memory. A high density DSM means that more memory is required for an equivalent area of a less dense DSM. However, the GPU imposes limits to the size of individual textures and on the total amount of memory that can be used. Increasingly dense DSMs pose a significant memory challenge. GPU memory management must be handled intelligently.

The GPU is designed to work with matrices, which function well to model frame-based sensors. However, are less suited to model satellite sensors. This is due, partly to the physics of the sensor, but also to the adverse effects of the atmosphere in bending light rays. Therefore, mathematical equations for satellite sensors are non-linear. The GPU handles linear interpolation well, but non-linear equations create difficulty.

4. SIMACTIVE'S APPROACH

SimActive's DSM algorithms are based on a top-down approach that significantly differs from the traditional bottom-up approach: a solution is derived to explain the measurements observed in the images. As opposed to searching for matching points in the imagery, elevation values on the DEM are refined until they correspond with what is observed in the imagery, solving the correlation problem. Specifically, an input DEM is supplied, super-sampled to match the desired output DSM resolution and refined in the aforementioned fashion until the correlation problem is solved for every grid post. The process is depicted in Figure 6.

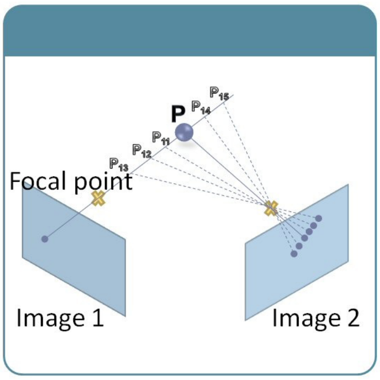

Figure 6: A top-down approach representation.

One drawback of the traditional feature based approach is to calculate only feature elevation points (i.e. at feature points in the image), which may not capture true ground as a result of interpolation. Instead of extracting feature points and interpolating, SimActive tackles the problem by creating a grid and calculating a correlation score for every post, thus not requiring interpolation. Consequently, this drastically reduces the risk of not capturing true ground and provides a more accurate representation of the terrain.

The process begins by solving for many possible solutions as represented by the matrix in Figure 7 below. The idea is to feed the GPU cores with many possible solutions to expose the massively parallel architecture. These solutions are processed by the GPU simultaneously. When a match is found, the problem is solved.

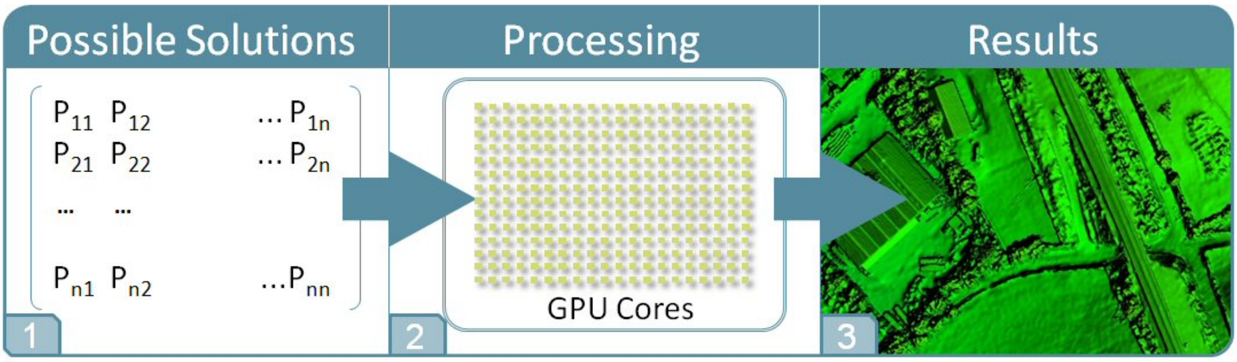

Figure 7: SimActive's DSM using the GPU.

The graphics processing unit (GPU) is used for DSM generation resulting in supremely fast processing speeds. Images are loaded into the GPU memory on a pair-by-pair basis, significantly reducing memory constraints on the system. The process begins by loading a pair of images into the GPU memory (or image tiles depending on the graphics card memory). A DSM patch corresponding to this pair of images is created and stored on disk. This process repeats until all the images have been processed. The resulting overlapping DSM patches are then optimized and merged in the following manner. A weight is associated with every point within each DSM patch based on a confidence measure. This measure is based on different metrics, including one that weights elevation values according to their distance from the centre of the DSM patches (to reduce potential occlusion problems).

Traditional multi-ray matching increases DSM accuracy, but takes substantially longer to compute because of the increased image load due to a higher overlap percentage. The idea behind the technique is to facilitate the correlation process by utilizing the higher overlap images. Using different correlation techniques, Correlator3D™ managed to leverage the improved accuracy afforded by multi-ray matching, without compromising processing speed. At the heart of this is the ability to merge different elevation measurements for the same grid post from various image pairs – effectively multi-ray matching – without the need for a higher overlap. Also, the software's ability to perform correlation in a highly robust manner allows removing the requirement for high overlap imagery.

SimActive's DSM engine was designed to run on standard desktop PCs and to profit from standard 3D graphics cards such as the NVIDIA GeForce & Quadro series. Contrary to common belief, processing time will not necessarily decrease with more GPUs. This is due to I/O constraints, where the bottleneck is more often in moving large amounts of data around the system rather than in GPU time. Furthermore, SimActive specifically optimized the software for standard PCs to be more accessible.

5. SAMPLE RESULTS

Three case studies will be examined on the accuracy and speed of the DSM generated using SimActive's Correlator3D™ software. A large format camera, satellite sensor and consumer grade UAV camera will be examined. All results were performed on the test system in Table 1.

CPU	Intel i7
GPU	GTX 680
RAM	6 GB
OS	Win 7

Table 1: Test system.

5.1. Large-Format Camera

This case study evaluates the accuracy of the DSM generated using large format aerial imagery and compares elevation values for the DSM with ground truth. The project was flown using a large format camera. The project is composed of 7 flight lines and 357 images with 60% forward overlap and 50% side overlap. The resulting pixel size was 10 cm. The project specifications are presented in Table 2.

Images	357
GSD	0.1 m
Resolution	17310 x 11310

Table 2: Project specifications.

A DSM was generated at a horizontal resolution of 30 cm. Prior to processing, the software predicted a vertical accuracy of 0.1 m for the final DSM. A total time of 47 hours (approximately 7.9 minutes per frame) was necessary for processing. Note, an entire dense DSM project can be processed in under 2 days demonstrating the exceptional speed of Correlator3D™ through its use of the GPU. Figure 8 shows the resulting DSM while Table 3 presents the statistics of the DSM generation process.

Horizontal resolution	0.3 m

Vertical accuracy	0.1 m
Processing time	7.9 min/frame

Table 3: SimActive DSM generation statistics.

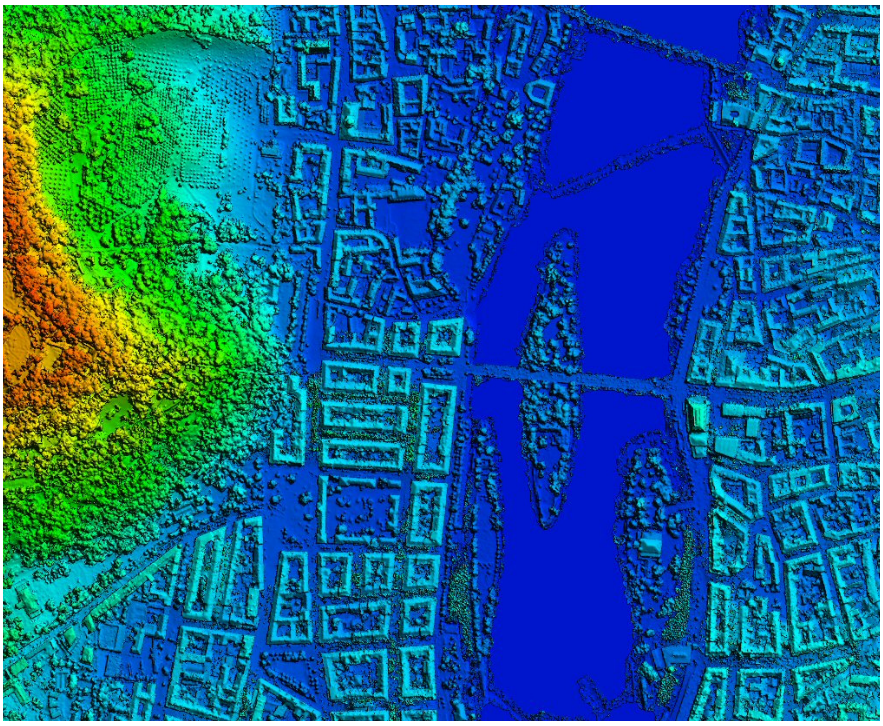

Figure 8: DSM corresponding to the data in Table 2.

To measure the final accuracy of the DSM, ground truth was compared against the elevation values generated by the software. Table 4 presents the results. Observe that the calculated RMSE on the DSM elevation values is 11.1 cm, which represents about one time the input imagery GSD. This value is consistent with the accuracy initially predicted by the software (10 cm), considering that the latter should hold for 95% of the points. Also, note that the observed bias was very small at 4 cm, which is within the GSD of the imagery.

Entry	Ground Truth Elevation (m)	SimActive DSM Elevation (m)	Delta(m)
01	280.31	280.40	0.09
02	277.61	277.73	0.12
03	284.97	284.90	-0.07
04	222.64	222.77	0.13
05	294.39	294.28	-0.11
06	232.31	232.27	-0.04
07	308.69	308.85	0.16
08	309.72	309.82	0.10
09	325.73	325.88	0.15
10	293.05	292.94	-0.11
RMSE			0.11
Bias			0.04

Table 4: SimActive DSM comparison.

5.2. Satellite

This case study examines the quality of SimActive's DSM using satellite imagery from the GeoEye-1 Hobart sample. The resolution of the sample used is 0.5 m, which was down-sampled from 0.41 m as per legal requirements for commercial use. The acquisition specifications are presented in Table 5.

Satellite	GeoEye-1
Orbit height	680 km
Images	2
GSD	0.5 m
Resolution	37943 x 34709

Table 5: Project specifications.

A DSM was generated at a horizontal resolution of 1.5 m. Prior to processing, the software predicted a vertical accuracy of 0.28 m for the final DSM. A total time of 45 minutes was necessary for processing. Figure 9 shows the resulting DSM while Table 6 presents the specifications and statistics of the DSM generation process.

Horizontal resolution	1.5 m
Vertical accuracy	0.28 m
Processing time	15 minutes

Table 6: SimActive DSM generation statistics.

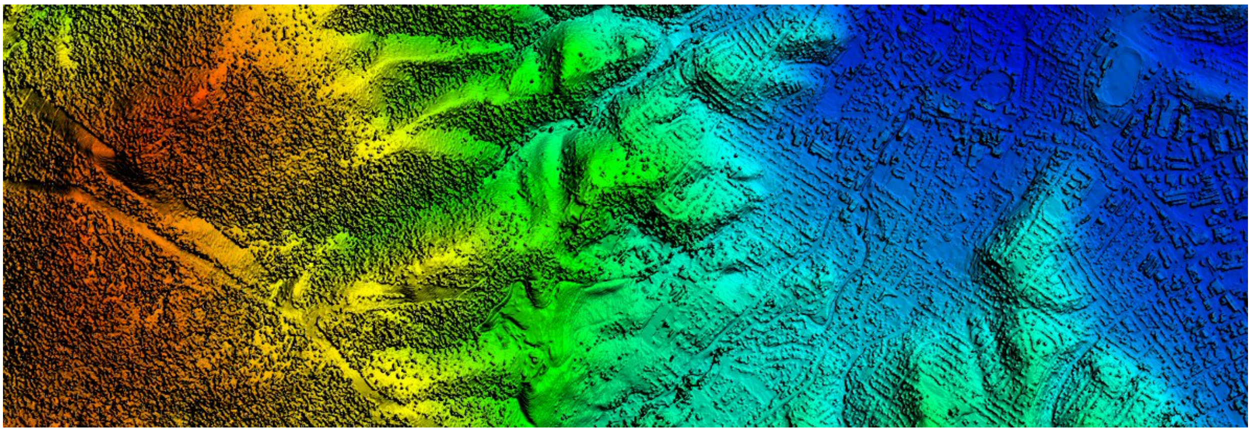

Figure 9: DSM corresponding to the data in Table 6.

To measure the final accuracy of the DSM, seventy-four (74) ground control points were compared against the elevation values generated by the software. Table 7 summarizes the test characteristics.

Input GSD	0.5 m
Estimated error of manual GCP detection	0.5 m
Number of GCP used for analysis	74

Table 7: DSM test characteristics.

Table 8 presents the statistical analysis results comparing the 74 GCPs against the SimActive DSM. Observe that the calculated RMSE for the DSM elevation values is 0.73 m, which represents 1.46 times the input imagery GSD. Also, note that the observed bias was very small at 0.57 m.

RMS	0.727 m
Bias	0.565 m
Standard Deviation	0.461 m

Table 8: DSM analysis results.

5.3. UAV

This case study examines the quality of SimActive's DSM using imagery acquired from a UAV. The resolution of the sample used is 5 cm. The acquisition specifications are presented in Table 9.

Images	2569
GSD	0.05 m
Resolution	5472 x 3648

Table 9: Project specifications.

A DSM was generated at a horizontal resolution of 0.25 m. Prior to processing, the software predicted a vertical accuracy of 6 cm for the final DSM. A total time of only 7.1 hours was necessary for processing. Figure 10 shows the resulting DSM while Table 10 presents the statistics of the DSM generation process.

Horizontal resolution	0.25 m
Vertical accuracy	0.06 m
Processing time	10 seconds/frame

Table 10: SimActive DSM generation statistics.

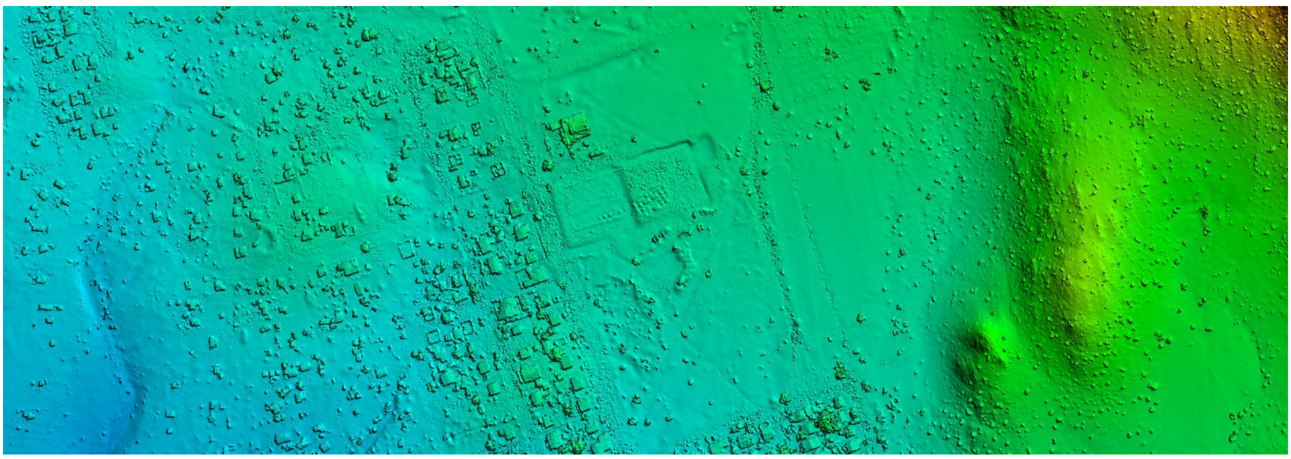

Figure 10: DSM corresponding to the data in Table 10.

To measure the final accuracy of the DSM, 10 ground control points were compared against the elevation values generated by the software. Table 11 summarizes the test characteristics.

Input GSD	0.05 m
Estimated error of manual GCP detection	0.05 m
Number of GCP used for analysis	10

Table 11: DSM test characteristics.

Table 12 presents the statistical analysis results comparing the 10 GCPs against the SimActive DSM. Observe that the calculated RMSE for the DSM elevation values is 7.5 cm, which represents 1.4 times the input imagery GSD. Also, note that the observed bias was very small 2.8 cm.

RMS	0.075 m
Bias	0.028 m
Standard Deviation	0.021 m

Table 12: DSM analysis results.

6. CONCLUSION

Generating dense elevation models using the GPU was determined to provide superior performance over using the CPU alone. The challenges of GPU computing was discussed and its specific effect on extracting maximum value for photogrammetry applications. Algorithm portability and interoperability between hardware are the main obstacles to implementing an effective dense DSM solution. SimActive's solution solved the challenges of GPU computing and produced a highly sophisticated dense DSM module. Building on its initial GPU code in 2008, SimActive continued optimizing the code and is now in its third generation. The challenge remains to continue the optimization effort for future GPUs and to extract maximum value. SimActive's dense DSM produced consistently accurate results with the RMSE approaching the input image GSD, at exceptional processing speeds.