High Performance Computing for Autonomous Planetary Exploration

Khaled Sharif
Intelligent Robotics Group
KBR, NASA Ames Research Center
Mountain View, CA
khaled.sharif@nasa.gov

Jordan Ford
Electrical & Computer Eng.
Carnegie Mellon University
Pittsburgh, PA
jsford@andrew.cmu.edu

Red Whittaker
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA
red@cmu.edu

Uland Wong
Intelligent Robotics Group
NASA Ames Research Center
Mountain View, CA
uland.wong@nasa.gov

***Abstract*—Modern parallel computers could power the perception and compression algorithms small planetary rovers require to navigate long distances, construct detailed terrain maps, and communicate discoveries to Earth. This work identifies and comprehensively characterizes four algorithms important to planetary roving that are well-suited for parallel computing. Multiple implementations of dense stereo matching, multi-view stereo, image compression, and triangle mesh compression are evaluated using the NVIDIA Jetson family of high-performance embedded computers. Image and mesh inputs are derived from simulation and used to evaluate the performance, power consumption, and hardware utilization of each device as a function of time. Our results demonstrate the promising capacity for modern embedded computers to expand the range and pace of planetary rover exploration.**

***Index Terms*—autonomy, benchmark, compression, parallel computing, perception, rover**

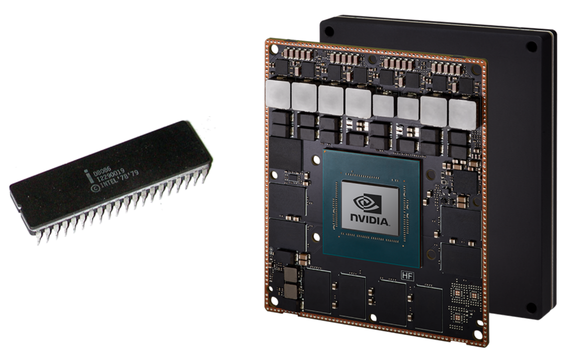

Fig. 1. (Left) An Intel 80C85 processor powered the Sojourner rover. (Right) The NVIDIA Xavier AGX is a modern embedded computer with a powerful GPU designed for autonomous driving and edge AI.

I. Introduction

Small planetary rovers require powerful, efficient computing to navigate long distances, construct detailed terrain maps, and communicate discoveries to Earth. In many cases, the technology hurdle for micro-roving mission architectures is the need to perform perception tasks on-board in support of high-cadence operations. These tasks are well-suited for an emerging class of power-efficient computers, with hardware accelerated support for graphics and compression, and a promising path to space flight. There is a need to comprehensively evaluate the performance of planetary roving tasks on modern embedded, parallel computers.

Past planetary rovers have driven at ponderously slow speeds and utilize human decision making periods between communications gaps. These architectures rely on steady operations over lengthy missions to conduct science. They use slow, heritage processors with large semiconductor feature sizes to survive decades of exposure to cosmic radiation. In contrast, the modern class of micro-rover missions is commercially incentivized and limited by power source. These missions are expected to conduct focused science investigations in a time-boxed manner, where the tempo of operations precludes low-level human decisions and requires performing autonomous perception on board the rover. Because micro-rovers are not intended to survive long durations in deep space, their total radiation exposure is limited, and they can leverage parallel computing devices with smaller feature size [2, 15].

In this work, we select state-of-the-art implementations of four enabling perception algorithms from the world of terrestrial autonomous driving to analyze suitability and characterize performance for future planetary roving. The focus is on the imaging pipeline for autonomy - dense stereo matching for local mapping and hazard avoidance; multi-view stereo for large-scale 3D mapping; compression for transferring sequential imaging data over a limited communications link; and mesh compression for efficiently transferring 3D science and navigation results.

A common basis of comparison was developed for evaluating perception as automotive datasets, such as KITTI [4], are not relevant to planetary environments. Reference datasets were instead generated using a combination of simulated Lunar imagery, lunar terrain models, and images terrestrial analogs. Performance is evaluated using the NVIDIA Jetson series of embedded high-performance computers that combine multiple power-efficient CPUs with a performant GPU. We evaluate the perception benchmarks using Jetson devices and measure their performance and power consumption. In each case, multiple algorithmic implementations are compared with each other and against ground truth for accuracy.

The results of this work indicate that the Jetson line specifically, and embedded automotive-grade computers in general, are promising for use in planetary applications. In particular, we show that the Jetson computer is both suited for fully-autonomous perception functionality and germane to the size, weight, and power (SWAP) constraints for micro-rovers of the 10-30kg range in the latest NASA commercial Lunar surface technology specifications [7].

II. Background

The Sojourner rover landed on the surface of Mars in 1997 in the Ares Vallis region [14]. Sojourner was the first wheeled vehicle to explore another planet, but it never explored more than 12 meters from the Pathfinder lander. Its onboard computer, a 2MHz Intel 80C85 processor (Fig. 1), could not support the autonomous perception and navigation required to explore beyond view of the lander's cameras. In the 45 years since Intel released the 80C85, the computational performance and power efficiency of embedded computers has advanced tremendously, enabling modern planetary rovers to operate far more autonomously than Sojourner ever could.

Recent trends in the space industry have pushed the onboard processing, power, and storage toward terrestrial industrial practices. The goal is to support more complex data organization, autonomous decision making, intensive signal processing and multitasking, and the coordination of large distributed development teams [6]. One important industrial practice is the use of commercial-off-the-shelf (COTS) devices, which compete with conventional radiation hardened components by offering the advantages of higher performance, faster development, and lower cost appropriate for the risk posture of short missions.

Compute Unified Device Architecture (CUDA) [9] is a parallel computing platform and application programming interface model that enables the use of graphics processing units (GPUs) for general purpose processing—an approach known as general-purpose computing on graphics processing units (GPGPU).

ARM big.LITTLE [8] is a heterogeneous computing architecture which couples slow, power-efficient processors (LITTLE) with faster, more power-hungry processors (big). All cores share the same memory, so workloads can be swapped dynamically between big and LITTLE cores to reduce overall power consumption. This model would enable planetary rovers to perform monitoring and safeguarding tasks on a small, efficient processor and to engage a more powerful processor only as-needed for compute-heavy tasks like perception and planning.

The NVIDIA Jetson computers have recently emerged as a series of powerful, energy-efficient embedded computers that combine a GPGPU with a big.LITTLE CPU array. The series flagship, the Jetson AGX Xavier, is the world's first computer designed specifically for autonomous machines. Preliminary testing has demonstrated that these computers are viable for space environments in missions with low total ionizing dose and hence are promising for micro-rover use [2], [15].

III. Benchmark Algorithms

We have identified dense stereo image matching, multi-view stereo modeling, image compression, and triangle mesh compression as critical algorithms for autonomous planetary exploration with great potential for speedup by modern computing. Rovers use dense stereo matching to map local terrain for obstacle avoidance and navigation. Multi-view stereo enables in-situ mapping of large-scale planetary features such as impact melt pits, crater walls, and subterranean caves. Triangle mesh and image compression enable small rovers to return their exploration data to Earth over a shared, low-bandwidth radio link.

A. Dense Stereo Matching

Stereo perception gives planetary rovers the ability to navigate autonomously in an unknown environment by using image sensors to map their surroundings in three dimensions. Stereo cameras at the front of a planetary rover feed synchronized image pairs to a stereo matching algorithm that determines the 3D shape of the landscape ahead of the rover. Efficient stereo matching is critical for planetary rovers. It enables the rover to detect and avoid hazards like rocks or craters.

Semi-global matching (SGM) [5] is a dense stereo matching algorithm that estimates a disparity map from a rectified stereo image pair. We evaluated three implementations of this algorithm on each Jetson computer. To form a mission relevant dataset, we collected images from a previously built simulation of a Lunar rover. Stereo cameras at the front of the simulated rover feed images into each algorithm implementation and produce a dense disparity map. This map is then compared to a ground truth disparity map obtained directly from the simulation to assess the accuracy of each implementation. We also assess the speed of each implementation by measuring the number of input image pairs processed per second.

Because dense stereo matching is meant to run on the rover continuously throughout the mission, we further parameterize this experiment to account for the power modes offered by the Jetson computers. We run each stereo matching implementation on each Jetson computer in both maximum-performance and maximum-efficiency modes.

B. Image Compression

Small planetary rovers have been proposed to explore and image vast terrain features on the Moon. They will explore long distances while capturing thousands of high-resolution, overlapping images. Because small rovers cannot carry, power, or aim a direct-to-Earth radio, they will relay these images to the lander, which will deliver them to Earth. In order to return thousands of high-resolution images over the lander's low-bandwidth connection, significant data compression is required. Because rover exploration imagery exhibits significant overlap between frames, we propose to use video compression techniques onboard the rover to reduce the bit rate required to return vast exploration imagery to Earth.

Modern computers have excellent hardware acceleration for video compression and decompression algorithms. We assess the Jetson computer's performance and power consumption when compressing (encoding) a series of terrain images. We use the same images that are used as input to our structure from motion pipeline. Video compression techniques leverage the significant overlap between frames to achieve high compression ratios.

C. Multi-View Stereo

Planetary rovers capture thousands of high resolution images of various geological features in order to construct a detailed terrain map. The problem with this approach is that it does not scale well. The images cannot be transmitted to Earth without significant lossy compression. In many scenarios, the best solution to this problem is to process the images on board the rover and construct high-fidelity three-dimensional models from the imagery. The resulting models retain all the necessary information from the images, and redundant data can be omitted from automatic transfer. As the rover continuously explores the planetary surface, the models increase in fidelity and coverage. The model can also be used by the rover to decide which locations have sufficient coverage and which locations require further exploration.

Structure from Motion is the process of reconstructing three-dimensional structure from a series of images taken from different viewpoints. Incremental structure from motion is a sequential processing pipeline with an iterative reconstruction component. It commonly starts with feature extraction and matching, followed by geometric verification. The resulting sparse point cloud serves as the foundation for the reconstruction stage, which seeds the model with a carefully selected two-view reconstruction, before incrementally registering new images, triangulating scene points, filtering outliers, and refining the reconstruction using bundle adjustment.

We used a dataset of images previously collected at a sinkhole in Utah [3]. The sinkhole was determined to be a suitable terrestrial analog for a lunar pit (Fig. 3). To assess the reconstructed model accuracy, we used a LIDAR scan of the sinkhole as geometric ground truth. We assessed the Jetson computer's performance and power consumption as it constructed a three dimensional model of the sinkhole from the collected images. We also assessed the resulting model accuracy by comparing it to the LIDAR scan by calculating point to plane deviation.

D. Triangle Mesh Compression

The result of the structure from motion pipeline is a three dimensional structure in the form of a triangle mesh. This mesh is textured with relevant subsections of the collected images, and all remaining imagery is redundant and therefore can be safely discarded. The resulting triangle mesh is therefore significantly smaller than the sum of the captured images. However, this triangle mesh may still be too large in size to be transmitted back to Earth over a shared, low bandwidth radio link.

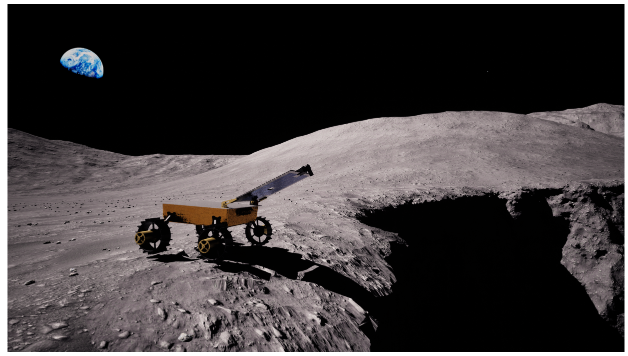

Fig. 2. The images and meshes used in this work were created using the planetary rover simulation environment shown here.

A solution to this problem involves compressing the mesh using techniques that are specifically designed to reduce and simplify triangle mesh data structures. We assess the Jetson computer's performance and power consumption when compressing a very large triangle mesh. The same triangle mesh that is output from the structure from motion pipeline is utilized in this benchmark for planetary mission relevance.

Fig. 3. The West Desert Sinkhole is a terrestrial pit with size and shape comparable to pits on the Moon. The simulated pit used in this work is derived from a LIDAR scan of the West Desert Sinkhole.

E. Simulation Environment and Benchmark Input Data

A lunar rover simulation environment was developed and used to create relevant inputs for benchmarking (Fig. 2). The simulator generates terrain images from a randomized procedural process and can accept input from pre-existing data. A simulated stereo camera pair on the front of the simulated rover was used to capture synchronized, rectified imagery for dense stereo matching and image compression. Disparity maps were ray-traced directly from the simulation geometry to serve as ground truth for evaluating the accuracy of the stereo matching algorithms (Fig. 4).

A simulated camera mounted on the top of the rover's solar panel was used to capture overlapping imagery of a simulated lunar pit for evaluating multi-view stereo. The textured triangle mesh created by multi-view stereo is compared to the simulated terrain mesh and is used in benchmark evaluations of triangle mesh compression.

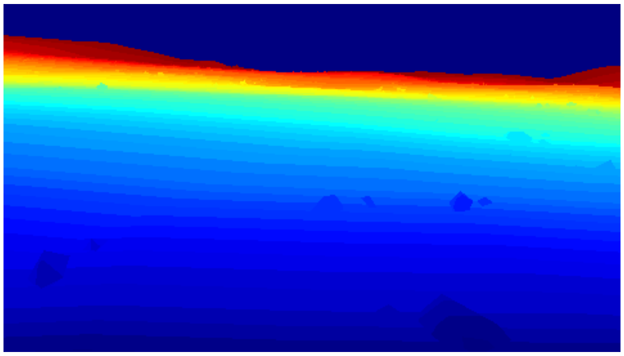

Fig. 4. Ground truth disparity maps produced using the planetary rover simulation environment are used to evaluate dense stereo matching accuracy.

IV. Results & Analysis

We evaluate each benchmark on the NVIDIA Jetson TX2, Xavier NX, and Xavier AGX devices in their maximum performance modes and record the power consumption and time required to complete each task. They were all flashed using NVIDIA Jetpack 4.5 to ensure that they run the experiments in a consistent, reproducible environment. All three machines run identical software, the same operating system (Ubuntu Linux 18), and the same CUDA library (version 10).

A. Dense Stereo Matching

We have selected three implementations of the semi-global matching (SGM) algorithm. The first (OpenCV version 4.1.1) does not use hardware acceleration and is bound by the speed of the Jetson computer's CPU. The second (NVIDIA VisionWorks version 1.6.0) and third (LibSGM) use the Jetson's power efficient GPU to accelerate stereo matching and offload work from the computer's CPU.

	Frames per second			Power consumption (W)		
	TX2	NX	AGX	TX2	NX	AGX
OpenCV	45	77	83	2.5	3.3	6.5
VisionWorks	49	75	155	6.0	6.7	16.3
LibSGM	65	99	175	6.5	7.8	18.4

TABLE I. The number of frames processed per second and the average power consumed by each of the stereo matching implementations running on each Jetson computer in **maximum performance mode**.

	Frames per second			Power consumption (W)		
	TX2	NX	AGX	TX2	NX	AGX
OpenCV	25	18	15	0.9	1.3	1.4
VisionWorks	38	65	40	2.7	3.5	2.5
LibSGM	47	71	36	2.7	3.9	2.5

TABLE II. The number of frames processed per second and the average power consumed by each of the three stereo matching implementations on each of the Jetson computers in **maximum efficiency mode**.

Method	Disparity Error $> 5\%$
OpenCV	27.5%
VisionWorks	23.6%
LibSGM	23.8%

TABLE III. The accuracy of each stereo matching algorithm is evaluated as the percentage of disparity errors that are greater than 5% of the maximum disparity.

B. Image Compression

We have selected Advanced Video Coding (H264) [12] and High Efficiency Video Coding (H265) [13] as suitable algorithms for encoding a series of images taken by a planetary rover. H264 is a video compression standard based on block-oriented, motion-compensated coding. It is by far the most commonly used format for the recording, compression, and distribution of video content, used by 91% of video industry. H264 is succeeded by H265, which offers between 25% to 50% better data compression at the same level of video quality, or substantially improved video quality at the same bit rate. Both algorithms are executed by hardware accelerators built into all of the Jetson devices.

We use ffmpeg [1] (version 3.4.8) to produce a suitable baseline for comparison. We run ffmpeg using default settings, but keep the resolution, frame-rate, and bit-rate the same between all algorithms.

	Time elapsed (s)			Power consumption (W)		
	TX2	NX	AGX	TX2	NX	AGX
ffmpeg	614	286	103	5.8	7.2	18.8
h264	28	25	15	5.0	5.7	12.0
h265	28	27	15	5.0	5.6	11.2

TABLE IV. The elapsed time and average power required by each Jetson computer to encode 100 images into a video

Method	Compression Ratio
ffmpeg	2.9
h264	38.1
h265	37.1

TABLE V. The compression ratio achieved by each video compression algorithm when used to encode 100 images into a video.

C. Multi-View Stereo

We have selected COLMAP [10, 11] (version 3.7) as a suitable software library for building our photogrammetry pipeline. The software makes good utilization of the GPU on each Jetson computer because it manages to parallelize various stages of the pipeline. We optimize the library for our specific use case. For example, because we have an approximate location for each photo in our pipeline input, we can selectively choose which images should match and which should be skipped. We are also able to incorporate the location of each image in the final model so that the model is geometrically accurate. The software is run in incremental

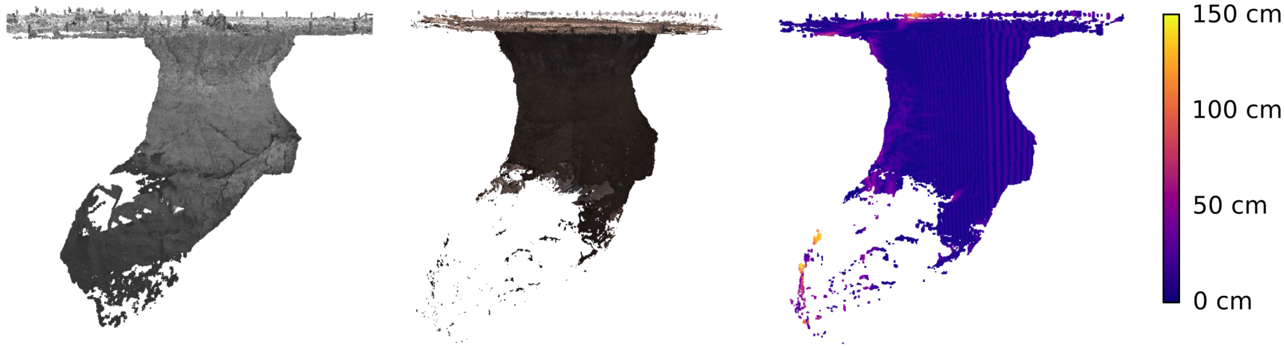

Fig. 5. The simulated lunar pit is based on a LIDAR scan of a terrestrial pit (left). The multi-view stereo pipeline produces a photogrammetric model of the pit (center). The point-to-plane deviation between the photogrammetric model and LIDAR ground truth (right) is less than 10cm across the majority of the pit interior.

	Time elapsed (min)			Power consumption (W)		
	TX2	NX	AGX	TX2	NX	AGX
Feature Extraction	5	4	3	8.3	10.2	18.2
Sparse Matching	36	11	7	9.5	15.1	29.3
Bundle Adjustment	83	101	67	4.5	6.1	9.1
Dense Matching	473	373	245	8.6	10.6	18.8
Dense Reconstruction	7	5	4	4.4	5.4	9.6
Full Pipeline	604	494	325	7.1	9.5	17.0

TABLE VI. The elapsed time and average power consumed by each Jetson computer to run each stage of the photogrammetry pipeline, where N=796 images

batches of the input photos in order to accurately simulate the rover taking a series of images over the course of several days. Each batch of images consists of 796 images.

We measure the time elapsed (minutes) and average power consumption (watts) of each Jetson computer as they run various stages of our photogrammetry pipeline. The pipeline is split into two halves: sparse reconstruction, which extracts and matches features, then runs bundle adjustment to generate a sparse point cloud; and dense reconstruction, which runs dense image matching and reconstruction in order to generate a high-fidelity three-dimensional triangle mesh.

The geometric accuracy of the resulting three-dimensional model is shown in Fig. 5. The accuracy is determined by calculating the point-to-plane deviation of the resulting model with the LIDAR ground truth.

D. Triangle Mesh Compression

Draco is a library for compressing and decompressing 3D geometric meshes and point clouds, and is intended to improve the storage and transmission of 3D graphics. Draco (version 1.4.1) was selected as a suitable software library for compressing the triangle mesh generated as a result from our photogrammetry pipeline. We run the Draco encoder at both its default (level 7) and maximum (level 10) compression levels, and record the time elapsed (seconds), average power consumption (watts), and compression ratio achieved.

	Time elapsed (s)			Power consumption (W)		
	TX2	NX	AGX	TX2	NX	AGX
libzip	698	586	351	4.74	5.89	9.6
draco (7)	192	120	81	4.60	5.80	9.3
draco (10)	469	310	199	4.50	5.70	9.3

TABLE VII. The elapsed time and power consumed to compress the triangle mesh terrain model

Method	Compression Ratio
libzip	1.6
draco (7)	14.4
draco (10)	16.7

TABLE VIII. The compression ratio obtained by each of the triangle mesh compression algorithms

We use libzip (version 3.0) to produce a suitable baseline for comparison. We run libzip to compress the same triangle mesh at its maximum compression setting.

E. Analysis and Discussion

Of the three devices tested in our work, the Jetson Xavier NX delivers the greatest performance per watt, and the Jetson Xavier AGX offers the greatest absolute performance. All three devices tested in this work are capable of delivering the necessary compute capability for autonomous planetary roving. In our estimation, the Jetson NX provides the most desirable balance of capability and SWAP efficiency for rovers in the 10-30kg class, which is a critical threshold for NASA's Commercial Lunar Payload Services (CLPS) programs.

Each Jetson device can be configured appropriately to be as powerful or as efficient as required for a mission. Different power modes on the Jetson can be configured and customized to turn cores on or off dynamically depending on the task at hand and energy constraints.

Our experimental results from each benchmark have revealed several areas of future work to improve algorithm performance and reduce average power consumption. Dense matching is currently the largest bottleneck of our structure from motion pipeline. Runtimes for structure from motion can be decreased by substituting this step for a patch-based or adaptive-resolution method. Large disparity errors in all stereo matching algorithms we tested indicate the need for new matching features better suited to planetary terrain.

V. Conclusions

The results of this work identify and characterize four tasks with principal importance in planetary roving with high potential for parallel speed-up. We evaluate a proposed family of flight-forward compute modules, the NVIDIA Jetson, for these tasks. To parameterize computational performance, a novel mission-relevant dataset is developed. We present the first benchmark evaluation of modern embedded computing modules as they apply to planetary rover perception, mapping, and communication. Through careful empirical analysis, we show that the NVIDIA Jetson is well-suited for planetary roving, in terms of overall performance, power consumption, and hardware utilization. These results pave the way for future study of the Jetson family of compute modules and their applications to planetary rover exploration.

Acknowledgment

The developments presented here are supported by NASA's Innovative Advanced Concepts (NIAC) program under grant #80HQTR19C0034.

References

[1] Yun Cheng et al. "Design and implementation of mediaplayer based on FFmpeg". In: *Software Engineering and Knowledge Engineering: Theory and Practice*. Springer, 2012, pp. 867–874.

[2] Sarah Katz Christopher Heistand and Amanda Voegtlin. "NVIDIA Jetson TX2i Radiation Report". In: *Military and Aerospace Programmable Logic Devices* (2019).

[3] Jordan Ford et al. "Image and Lidar Dataset of the West Desert Sinkhole: An Analog for Steep-Walled Planetary Pits". In: *3rd International Planetary Pits and Caves Conference* (2019).

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[5] Heiko Hirschmuller. "Accurate and efficient stereo processing by semi-global matching and mutual information". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. IEEE. 2005, pp. 807–814.

[6] George Lentaris et al. "High-performance embedded computing in space: Evaluation of platforms for vision-based navigation". In: *Journal of Aerospace Information Systems* 15.4 (2018), pp. 178–192.

[7] Sarah Noble. "Lunar Surface Instrument And Technology Payloads". In: *Lunar Surface Technology Instruments and Payloads Program Call for Proposals* (2019).

[8] Edson Luiz Padoin et al. "Performance/energy trade-off in scientific computing: the case of ARM big. LITTLE and Intel Sandy Bridge". In: *IET Computers & Digital Techniques* 9.1 (2014), pp. 27–35.

[9] Jason Sanders and Edward Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.

[10] Johannes Lutz Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[11] Johannes Lutz Schönberger et al. "Pixelwise View Selection for Unstructured Multi-View Stereo". In: *European Conference on Computer Vision (ECCV)*. 2016.

[12] Gary J Sullivan, Pankaj N Topiwala, and Ajay Luthra. "The H. 264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions". In: *Applications of Digital Image Processing XXVII*. Vol. 5558. International Society for Optics and Photonics. 2004, pp. 454–474.

[13] Vivienne Sze, Madhukar Budagavi, and Gary J Sullivan. "High efficiency video coding (HEVC)". In: *Integrated circuit and systems, algorithms and architectures*. Vol. 39. Springer, 2014, p. 40.

[14] Brian Wilcox and Tam Nguyen. *Sojourner on mars and lessons learned for future planetary rovers*. Tech. rep. SAE Technical Paper, 1998.

[15] E. J. Wyrwas. "Proton Testing of nVidia Jetson TX2". In: *NASA Technical Report* (2019).