SIMULATION-FIRST: RAPID SPACE ROBOTICS PROTOTYPING AND TESTING

Learn how SoftServe's simulation-first approach harnesses terrestrial technology to accelerate development and reduce associated costs and risks

softserve

EXECUTIVE SUMMARY

The average recurring price for a weather satellite is $290 million, while the price tag for military-grade satellites starts at $390M. Meanwhile, the average development time for a small satellite is 1-2 years, while for a larger, complex mission, it might be 5-10 years or more. With the space economy forecasted to almost triple to $1.8 trillion by 2035, and competition computing for lucrative contracts, companies are looking to drive faster prototyping at a lower cost and with reduced risks.

In this white paper, we describe an approach to combine simulations and space robotics accelerators developed by SoftServe using NVIDIA technology with radiation-hardened x86 compatible-edge hardware from Blue Marble Communications and BruhnBruhn Innvation's *dacreo* software stack. This approach narrows the gap between early prototypes and space-ready applications and enables the reuse of software components already adopted for terrestrial use cases, surpassing the capabilities of traditional radiation-hardened boards.

We show that our simulation-first testing and development approach enables complete software solutions to be developed in commercial environments and be ported to the Blue Marble Communications' Space Edge Processor (SEP). We further demonstrate the success of our approach by evaluating computer vision algorithms on a test unit of the SEP using NASA's POLAR Stereo dataset and photorealistic simulation developed in NVIDIA Isaac Sim.

SoftServe's simulation-first approach leverages the hardware and software of the SEP and *dacreo* to provide:

FASTER TIME-TO-MARKET:
Reduces development time and cost

SIMPLIFIED TESTING:
High-fidelity simulations streamline end-to-end verification

EASY SOFTWARE ADOPTION:
Supports popular robotics and AI frameworks via Linux

ADVANCED AI:
Enables sophisticated algorithms without specialized hardware

LOWER BANDWIDTH NEEDS:
Onboard computational power minimizes data downlink

THE NEED FOR A PARADIGM SHIFT IN THE SPACE ECONOMY

1. THE DEMANDS OF MODERN SPACE MISSIONS

The evolving landscape of space exploration calls for a paradigm shift in software development for modern missions. The key drivers for this evolution are:

ANTICIPATED GROWTH OF THE SPACE SECTOR:

The total value of the space economy has been forecasted to almost triple to $1.8 trillion by 2035. This highlights the need for more accessible technology stacks to make training the next generation of space developers easier.

INCREASED SOFTWARE COMPLEXITY:

Missions with advanced capabilities like robotic surface exploration will require more complex software, high-performance compute, and tighter integrations with simulation environments. Traditional hardware makes this a lengthy and challenging process.

PRIORITIZING SPEED OVER GUARANTEES:

Modern subsystems prefer increased computational power and streamlined development at the expense of the guarantees of traditional hardware. An example of this is commercial-off-the-shelf (COTS) solutions like NVIDIA Jetson, which have gained popularity for their speed and ease of use.

2. THE LIMITATIONS OF TRADITIONAL RADIATION-HARDENED BOARDS

Space-grade onboard computers must balance computational power with resilience to harsh space conditions. Traditional radiation-hardened boards have an extensive flight heritage, making them the go-to choice for critical subsystems in high-orbit and interplanetary missions.

Yet radiation-hardened (rad-hard) boards often lack flexibility due to their limited computing power and dependence on specialized processor architectures like PowerPC. This can lead to longer development time, challenging prototyping, and escalated costs for space projects.

While these boards typically employ real-time operating systems such as VxWorks or RTEMS, they often come with inherent limitations. Additionally, apart from resource constraints, they usually have low support for software ecosystems that are widely used in other domains, such as terrestrial robotics. This has several important consequences:

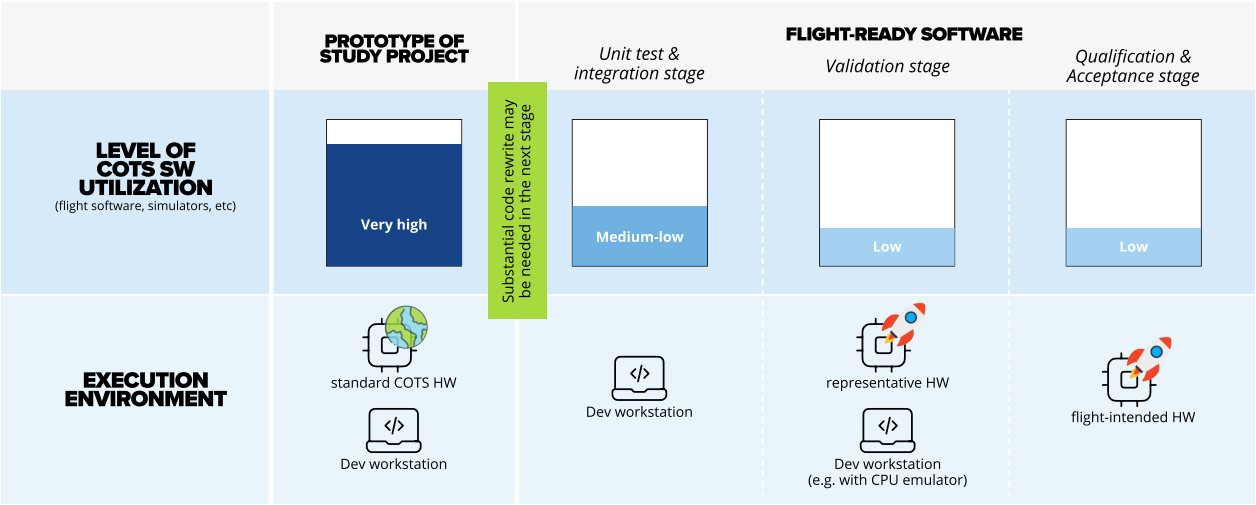
1. **Greater expense through low prototype reusability:** Building initial prototypes on flight-ready hardware is expensive, thus forcing developers to start with more accessible technologies like Robot Operating System (ROS) or SpaceROS, and simulation tools such as Gazebo, Algoryx AGX Dynamics, or NVIDIA Isaac Sim. Later, these need to be converted for space mission use, which vastly extends the development time.

2. **Hardware emulation and cross-target testing requirements:** The development of flight-ready code is often carried out on workstations where more advanced tools for debugging and profiling are available. This requires additional steps to ensure compatibility with the target hardware, including the development of abstraction layers or the need for hardware emulation. All of this increases development effort, makes verification more complex, and creates the need for multiple layers of tests.

3. **Limited access to modern software tools:** Common libraries like PyTorch or ROS are often incompatible with aerospace requirements. That delays development and limits what the final systems can do. Support for widely adopted simulation tools is also limited, leading to the need for specialized simulators or connectors.

4. **Limited hardware-in-the-loop testing capabilities:** Writing flight-ready code for rad-hard boards takes time, and testing advanced algorithms on the target hardware requires additional development that can only be carried out at later stages, increasing overall project risk.

5. **Steep learning curve for developers:** The availability of community support and resources for toolchains and software used in the space industry is often limited compared to large open-source and COTS solutions.



Example levels of COTS software utilization in various test phases for projects employing rad-hard computers. Names of test phases adhere to ECSS standards. COTS software encompasses both flight software and testing tools. Actual utilization levels may vary for each project.

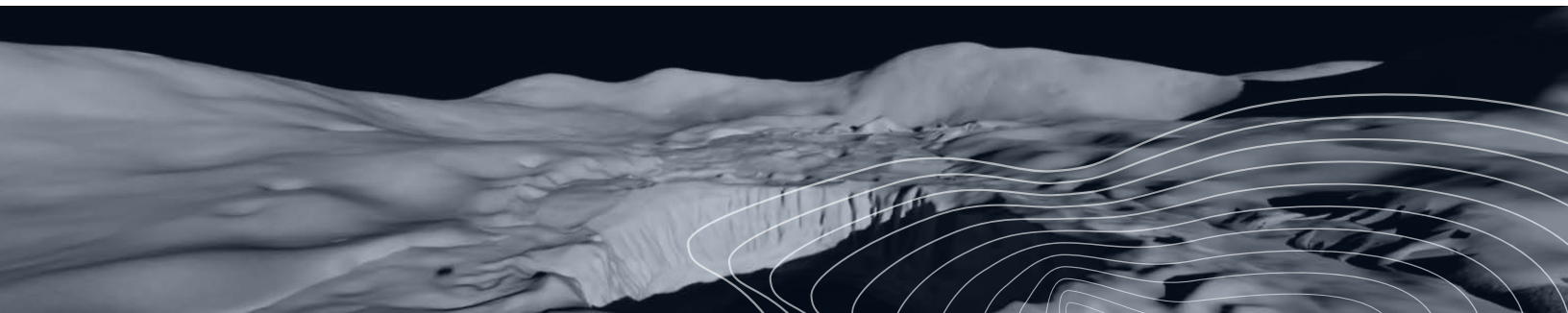
SIMULATION-FIRST TESTING AND DEVELOPMENT WITH MODERN RADIATION-HARDENED BOARDS

Meanwhile, a new generation of radiation-hardened computer boards has entered the market. These boards offer better computational capabilities and leverage mainstream processor architectures such as x86, which is commonly found in laptops, desktop computers, and enterprise servers.

The **Blue Marble Communications Space Edge Processor** (SEP) is an example of this new generation. It features a state-of-the-art space-qualified, radiation-hardened, heterogeneous compute platform running BruhnBruhn's Innovation's (BBI) Linux-based *dacreo* software ecosystem.

SPACE EDGE PROCESSOR **by Blue Marble Communications equipped with *dacreo* AI GPU ecosystem by BruhnBruhn Innovation AB**	
x86 CPU, GPU, and FPGA combined on a single board	**7 nm AMD V2748 APU (64-bit x86 CPU + GPU with VEGA architecture)**
7 nm AMD Versal Prime FPGA	**Dockerized ROCm software stack**
Seamless integration with SoftServe's simulation environment for Hardware-In-The-Loop testing	***Capability to deploy SoftServe's space robotics accelerators to the device***
Optional: **dacreoOS, an optimized embedded Yocto 5.0 LTS derivative Linux OS for lightweight robust containerized application deployment**	

To fully harness the potential of these new hardware and software systems to prototype and ultimately deploy technology solutions for the aerospace industry, it is critical to combine them with the most advanced forms of simulation-centric development and testing methods.



1. REQUIREMENTS FOR SIMULATION-FIRST TESTING

The SEP has integration support for COTS simulation environments such as NVIDIA Isaac Sim. This allows for the adoption of a simulation-first approach and enables continuous verification of evolving software in a fully-fledged testing environment, making “test as you fly” easier to implement.

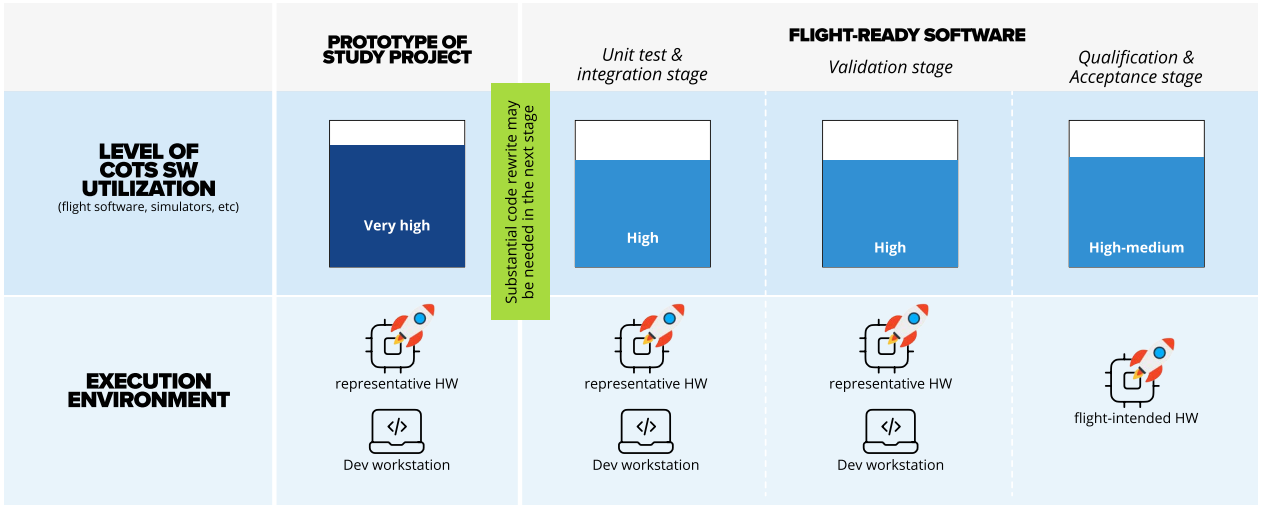
Furthermore, real-time connection to COTS simulation engines allows for on-demand hardware-in-the-loop testing, performance benchmarking, and software verification on the target hardware platform at any project stage. This enhances the representativeness of testing and makes the conformance to guidelines such as ECSS easier to achieve, reducing overall risk.

The SEP can interface with a co-simulation solution developed by SoftServe, which integrates NVIDIA Isaac Sim with Matlab FMI/FMUs. This enables executing complex scenarios, where visually rich simulation engines are integrated with high-fidelity models developed by subject matter experts in Matlab or Simulink and tested directly on the device.

2. REQUIREMENTS FOR ACCELERATED SOFTWARE DEPLOYMENT

BBI’s *dacreo* AI ecosystem is an adapted version of AMD ROCm high-performance compute stack, adjusted to run on the SEP’s VEGA GPU architecture. This ecosystem adds accelerated compute capabilities to the device and enables access to emerging technologies widely adopted in artificial intelligence and machine learning communities, such as OpenCL, Tensorflow, or Pytorch.

Additionally, it allows for writing low-level GPU code and enables the compilation of CUDA applications with relatively low porting effort. *Dacreo* running on SEP opens a possibility to easily optimize many applications, without the need for specialized hardware, reducing risks and overall cost of the end solution. These applications are built on top of established and emerging frameworks in the space domain such as cFS, ROS, OpenCV, and others.



Possible levels of COTS software utilization in different test phases, for modern rad-hard boards such as Space Edge Processor. The levels outlined in the diagram are just exemplary and the actual numbers depend on various aspects such as mission profile, system criticality, power budget, etc.

3. INITIAL VALIDATION OF SEP'S SIMULATION-FIRST READINESS

SoftServe has already developed solutions like Lunar Drone Accelerator or Lunar Robotic Excavator on the SEP, demonstrating its compatibility with Linux and the processor architecture of desktop computers. That implies that various algorithms and applications could be deployed and tested on the representative hardware from the project outset with off-the-shelf frameworks and packages. Moreover, large portions of prototype code could potentially be reused in the final flight-ready software.

Furthermore, we were able to deploy a unified development toolchain for the developer workstation and the Space Edge Processor. This allows for leveraging tools such as Valgrind for dynamic code analysis directly on the representative unit and makes both profiling and troubleshooting easier. Additionally, real-time remote debugging tooling present within the Linux ecosystem enhances troubleshooting capabilities, improving productivity and software quality.

Taken together, SoftServe's approach to development and simulation using the SEP offers several potential advantages over traditional radiation-hardened boards, further emphasizing why our simulation-first methodology stands out:

FASTER TIME-TO-MARKET:

Leveraging a combination of hardware and software reduces the time and cost needed to develop space robotics applications.

EASIER VERIFICATION AND TESTING:

The high-fidelity simulation environment developed by SoftServe with online access to hardware makes end-to-end testing more straightforward.

EASIER OFF-THE-SHELF SOFTWARE ADOPTION:

Facilitated by the Linux environment and software accelerators integrated with widely adopted robotics and AI frameworks.

ADVANCED AI CAPABILITIES:

High computational power with an optimized software stack enables sophisticated AI algorithms and applications without the solution-specific hardware.

LOWER BANDWIDTH REQUIREMENTS:

Enhanced computational capabilities onboard mitigate the need for downlinking data to Earth for processing.

CASE STUDY: BENCHMARKING STEREO MATCHING ALGORITHMS ON THE SEP

1. CONTEXT, AIMS, AND METHODOLOGY

New space missions are increasingly reliant on advanced robotics systems. To ensure safety and effective autonomy, robust and rapid robotic perception implementation is essential. Accordingly, there are substantial advantages to being able to run sophisticated perception pipelines without the need for specialized hardware.

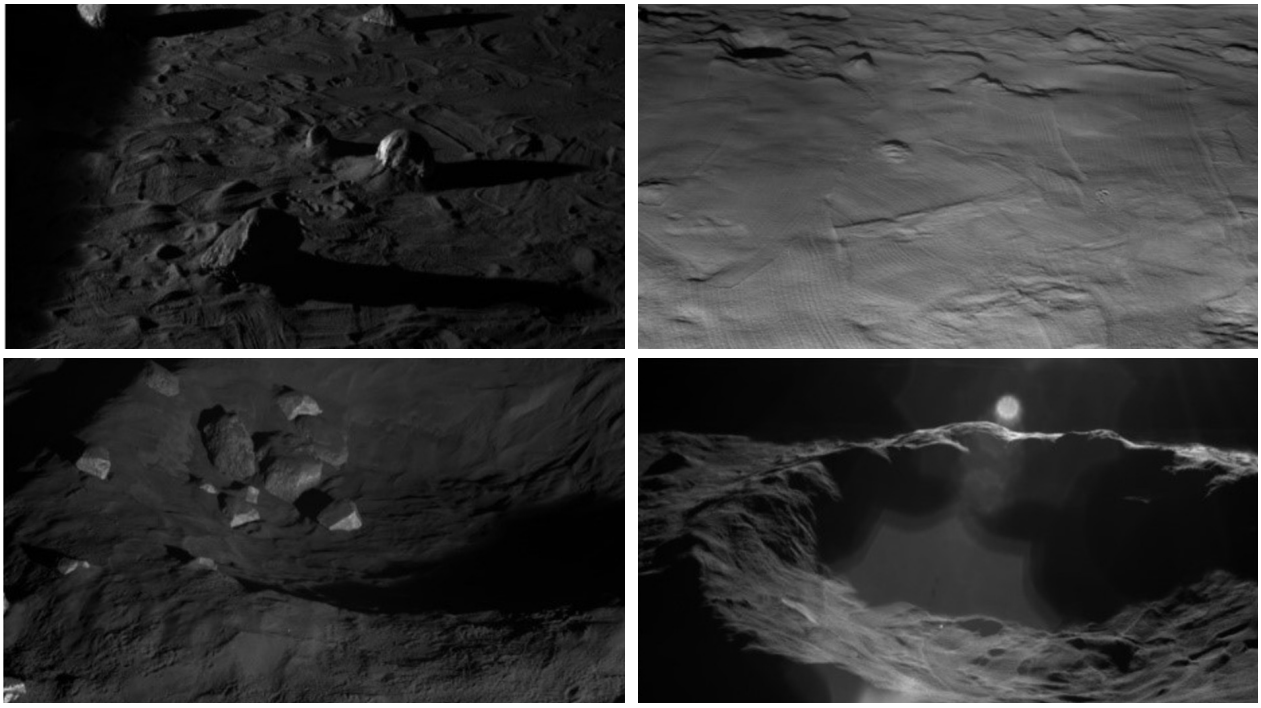
However, not all algorithms are equally conducive. By leveraging SoftServe's simulation-first approach to development, we evaluated and benchmarked several classical and AI-based stereo-matching algorithms on a test unit of the SEP.

The benchmark is aimed to evaluate the performance of the AMD 7nm ZEN 2/VEGA APU devices found on the SEP and the suitability of various stereo algorithms for space applications. To achieve this, we selected several stereo-mapping methods of varying computational complexity. Each algorithm underwent three tests:

1. Measurement of interference time on a reference system using AMD 7nm APU
2. Evaluation of algorithm quality using NASA's POLAR Stereo Dataset — an analog dataset generated in the laboratory, aiming to recreate the conditions at the poles of the Moon
3. Evaluation of algorithm quality using a synthetic image dataset created in NVIDIA Isaac Sim, imitating the visual conditions of the lunar south pole[2]

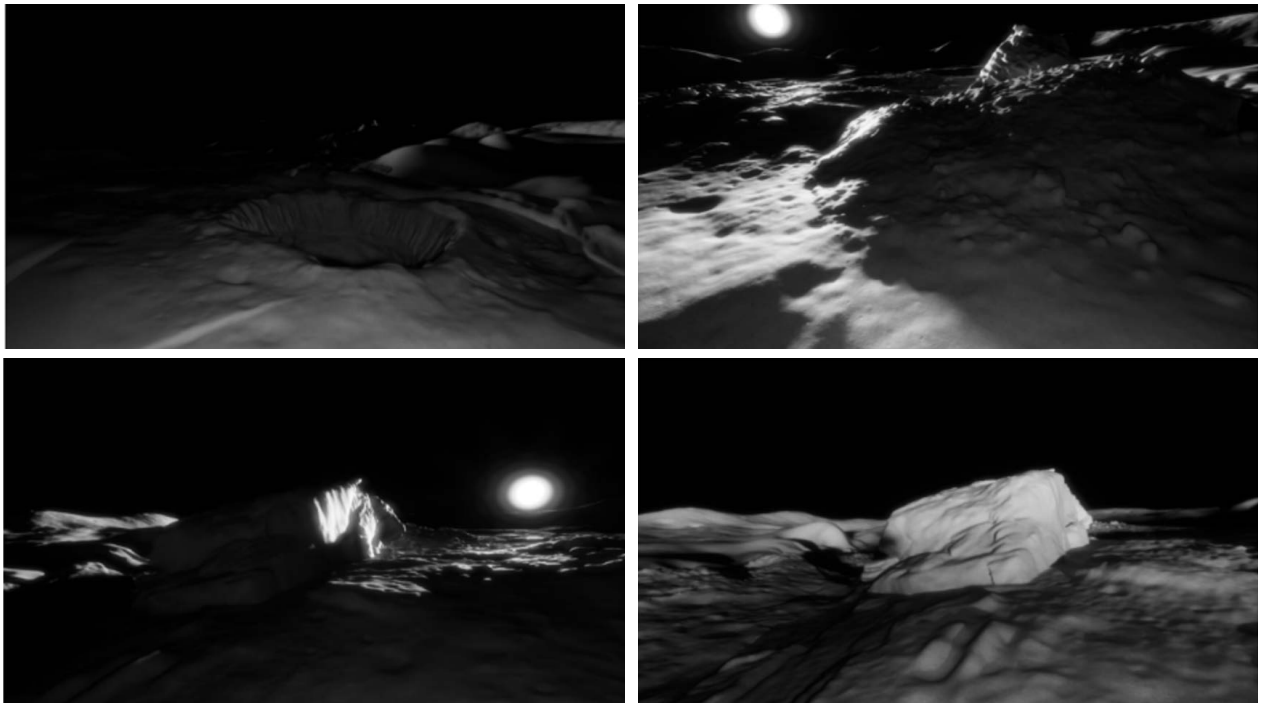
WE CONSIDERED THE FOLLOWING ALGORITHMS:[1]

Local Block Matching (LBM)	Cascade SGBM
Semi-Global Block Matching (SGBM)	Hitnet
Cascade LBM	Fast-ACVNet+
IGEV	



Sample images from POLAR Stereo Dataset

The quality of algorithms was assessed using metrics including detected pixel percentage, average pixel error, and misclassified pixel percentage, calculated for both synthetic and POLAR datasets. In the results, the misclassified pixel percentage is denoted as “Bad N%”.[3]



Sample images from the synthetically generated dataset

2. RESULTS

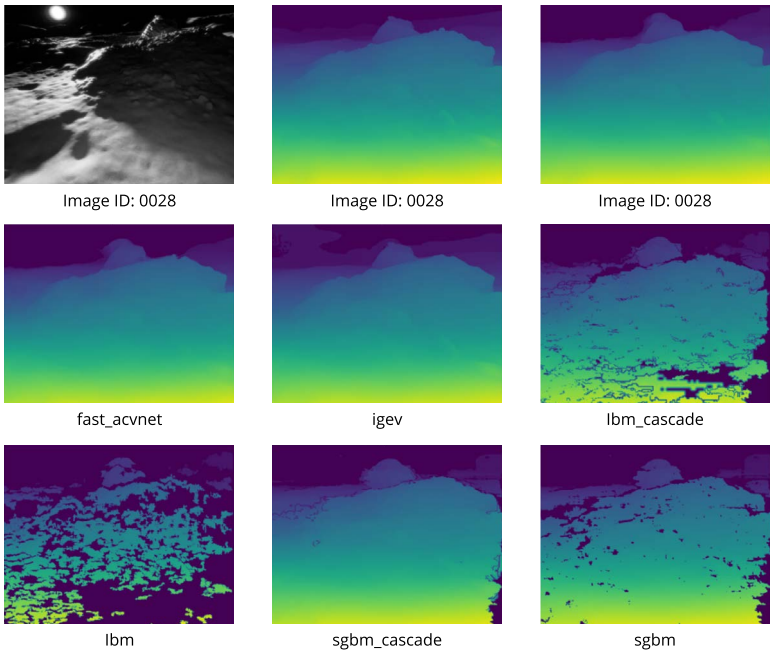
The resulting processing times and metrics capturing the quality of different algorithms are summarized in the table below:

METHOD	IGEV	HITNET	FAS ACVNET	SGBM CASCADE	SGBM	LBM CASCADE	LBM
ALGORITHM SPEED ON AMD RYZEN 7 4800U DEVICE							
Avg Inference time [ms] ↓	6320	1530	1060	82	64	35	24
ALGORITHM SPEED ON AMD RYZEN 7 4800U DEVICE							
Detected Pixels [%] ↑	99.87%	99.30%	99.29%	87.63%	77.95%	76.95%	56.71%
Average Error [px] ↓	2.32	3.92	4.78	3.98	1.19	5.97	1.16
Bad 1% [%] ↓	9.0%	15.2%	16.0%	9.3%	2.7%	15.1%	5.3%
Bad 2% [%] ↓	5.5%	8.9%	9.6%	6.3%	1.2%	8.6%	0.5%
Bad 4% [%] ↓	3.5%	6.4%	7.0%	5.0%	0.8%	7.9%	0.4%
ALGORITHM QUALITY ON SYNTHETIC LUNAR DATASET							
Detected Pixels [%] ↑	97.66%	97.58%	97.31%	93.82%	75.20%	74.43%	39.47%
Average Error [px] ↓	8.89	13.87	15.36	31.11	8.74	64.30	7.59
Bad 1% [%] ↓	27.6%	32.0%	32.7%	47.6%	38.6%	51.4%	32.4%
Bad 2% [%] ↓	14.5%	16.2%	16.7%	27.8%	17.6%	33.9%	13.5%
Bad 4% [%] ↓	8.3%	8.6%	8.2%	14.8%	6.2%	23.2%	5.4%
False positives [%] ↓	5.46%	2.44%	1.46%	2.18%	1.02%	2.09%	0.99%

The table with results for 3 experiments. ↑ means higher values are better, while ↓ indicates that lower values are better.

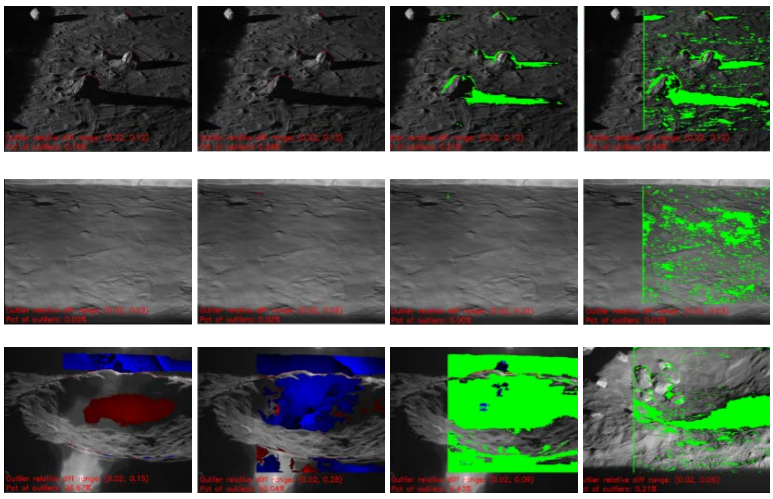
Classical methods achieved real-time performance on the Space Edge Processor, while deep learning methods suited scenarios with higher latency tolerance.[4] All algorithms utilized the integrated GPU, freeing CPU resources for critical tasks.

Deep stereo algorithms showed promising inference times, though certain operations created bottlenecks.[5] Results indicated that more computationally demanding methods produced dense stereo maps with the highest number of matching pixels. Deep learning methods generated more accurate depth maps on lunar-like datasets compared to classical methods, which struggled with texture-less regions.[6]



Comparison of algorithm outputs for a single image from Synthetic Lunar Dataset

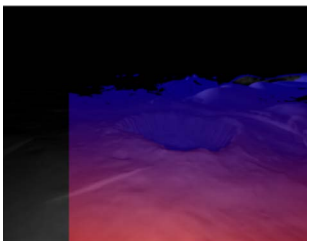
For POLAR Stereo Dataset, we observed that deep stereo algorithms usually surpass classical approaches for images with no direct sunlight pointing toward the camera. For images with sunlight, these algorithms tend to generate more outliers than classical methods, but they provide much denser stereo maps from a single image pair.



Stereo mapping results for 4 chosen algorithms, on 4 chosen images from POLAR Stereo Dataset. Green pixels show where depth couldn't be determined. The red pixels show where the predicted depth was lower than the actual depth. Blue pixels show where the predicted depth was higher than the actual depth.

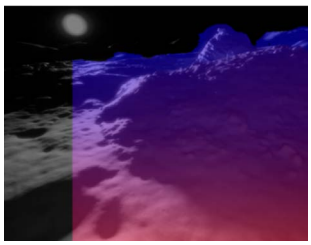
While the IGEV algorithm performed the best in detecting pixels on the synthetic dataset generated in NVIDIA Isaac Sim, it often misclassified lunar sky patches as scene objects, particularly when the sun was visible in the image, leading to false positives. Other algorithms, such as Hitnet or Fast-ACVNet+, resulted in fewer false positives on average on the synthetic dataset. We also observe that while IGEV performs the best on average, other algorithms outperform it for certain images.

Original image

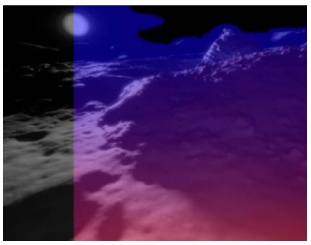


Disparity map blended on image

Sample disparity map obtained with IGEV algorithm for a synthetic image. The red spots are closer to the camera than the blue pixels.



Disparity map for Hitnet



Disparity map for IGEV

The IGEV algorithm produces false positives around the sun. Pixels in the sky are incorrectly classified as objects in the scene.

3. CONCLUSION

By deploying our simulation-first method, we determined that the processing power of Blue Marble Space Edge Processor allows for running more advanced, AI-based algorithms for stereo matching directly on the edge. We conclude that classical stereo matching algorithms, such as Local Block Matching and Semi Global Block Matching can run on the BlueMarble hardware with real-time performance, with only minor tweaks to publicly available code.

This may remove the need for FPGA ports, speeding development time and reducing risks. Moreover, the available computing capacity enables the utilization of multiple stereo pairs in real time, for increased robustness and scene coverage.

Although the inference time for AI-powered methods is higher than for classical algorithms, the coverage and quality of created disparity maps surpass classic cascade approaches, making them a promising alternative for certain scenarios. AI-powered algorithms may be suitable for use cases where the quality and coverage of the created map are significantly more important than latency. Such use cases may include near-real-time creation of maps of environments for science, or mid-range path planning.

CONCLUSION: SIMULATION-FIRST DEVELOPMENT FOR RAPID TESTING AND PROTOTYPING

The rapid expansion of the space economy demands innovative and efficient approaches to technology development and deployment. SoftServe's simulation-first prototyping and testing method lets you bridge the gap between early-stage prototypes and fully space-ready applications, enabling the reuse of terrestrial software solutions. That lowers costs, accelerates development, and minimizes risks for high-value space economy projects.

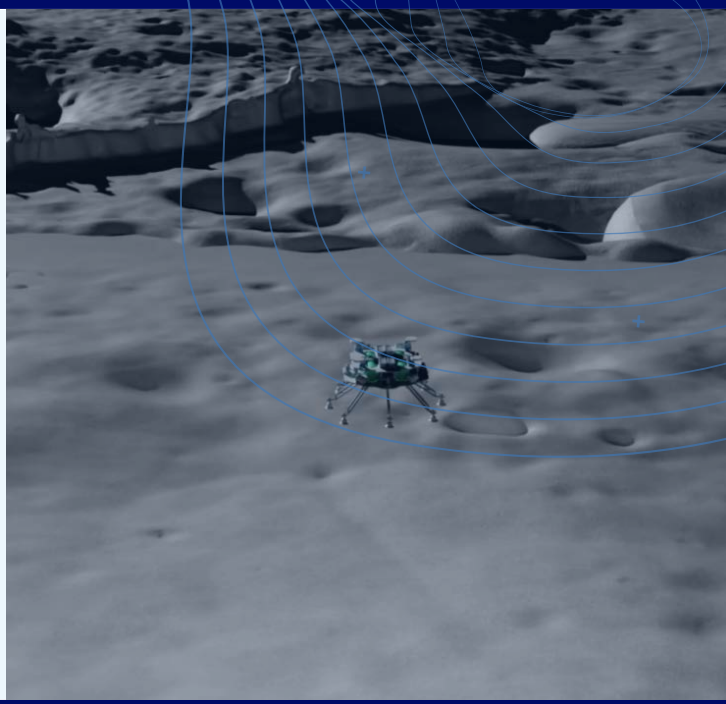
SoftServe partners with collaborators like BruhnBruhn Innovation and Blue Marble Communications to bring state-of-the-art technology and expertise to the forefront of space exploration. By integrating advanced simulations with space robotics accelerators utilizing NVIDIA technology, combined with Blue Marble Communications' SEP and BruhnBruhn Innovation's *dacreo* software, our method demonstrates unparalleled efficiency and adaptability.

Do you want to leverage our simulation-first method and accelerate the development of space-ready technology solutions?

Let us help you transform your vision into reality swiftly and efficiently.

Learn more about what SoftServe's Robotics Team can do for you here on

OUR DEDICATED PAGE



For this whitepaper, "SIMULATIONFIRST: RAPID SPACE ROBOTICS PROTOTYPING AND TESTING," the Appendix is provided on a dedicated page online. To explore this supplementary data, additional resources, and detailed information, please visit this page.

ABOUT US

SoftServe is a premier IT consulting and digital services provider. We expand the horizon of new technologies to solve today's complex business challenges and achieve meaningful outcomes for our clients. Our boundless curiosity drives us to explore and reimagine the art of the possible. Clients confidently rely on SoftServe to architect and execute mature and innovative capabilities, such as digital engineering, data and analytics, cloud, and AI/ML.

Our global reputation is gained from more than 30 years of experience delivering superior digital solutions at exceptional speed by top-tier engineering talent to enterprise industries, including high tech, financial services, healthcare, life sciences, retail, energy, and manufacturing. Visit our website, blog, LinkedIn, Facebook, and X (Twitter) pages for more information.

NORTH AMERICAN HQ

201 W. 5th Street, Suite 1550
Austin, TX 78701
+1 866 687 3588 (USA)
+1 647 948 7638 (Canada)

EUROPEAN HQ

30 Cannon Street
London EC4 6XH
United Kingdom
+44 333 006 4341

info@softserveinc.com
www.softserveinc.com

softserve