The Stellar Cartography Project: An Exploratory Study of Spatial Deep Learning Autonomous Models in Low Power Integrated Systems

Phase I SBIR Proposal for NASA Topic H6.22-2882

Proposed by: Adder Mobile Technologies, Inc.
Prepared by: Ian Gerard, CEO & Principal Investigator
3/20/2019

Proposal Summary

The Stellar Cartography Project is a spatial deep learning system for use in deep space exploration and support roles in early colonization and/or resource extraction.

Adder's Phase I SBIR proposal is focused on the machine learning implementations and processing capabilities needed to build a self learning autonomous mapping model capable of running on miniaturized hardware.

The purpose of this innovation would be intended as a cost effective method of deep space exploration via CUBESAT based systems we are designing for a Phase II proposal.

The Stellar Cartography deep learning model is being designed to ingest numerous types of disparate spatial data to extrapolate information and then interpolate insights via an autonomous self guided AI training model.

Our overall goal is to utilize our knowledge and resources in the field of spatial deep learning to build autonomous deep space location, navigation, communication, and processing tools.

The technical objectives of the Phase I proposal are focused on the successful creation of a self guided AI that can interpret and correlate spatial data and gain underlying insight from that data.

Specifically, we intend to have the deep learning model in an autonomous state, ready for limited test deployment on proof of concept hardware designed/assembled by our own internal team of software and hardware developers.

Summary of Work Plan

Identification and modeling of spatial data to be collected
Databasing and communication of datasets
Processing and Coprocessing utilizing neuromorphic methodologies
Test deployment on scaled-down hardware
Early environmental stress testing in regional lab facilities.

Part 1- Table of Contents

Part 2 Identification and Significance of the Innovation

2.1 Proposed Innovation

The Phase I innovation Adder intends to accomplish will be the creation of a deep learning algorithm that can ingest data and create spatial representations of planetary bodies, stars, and other objects of interest in a space exploration role that utilizes self-learning inferential AI models.

This system, dubbed the Stellar Cartography Project internally, or SCP, is envisioned as a system capable of mapping difficult-to-reach terrain on Earth, which uses a deep learning model we're already creating for use in our own mapping systems. We believe that our intended use could be broadened to address some of the numerous challenges associated with space exploration.

The SCP will seek to develop a solution for creating spatial representations of sensor data while in flight utilizing low-power machine learning models designed with these limitations in mind, with our particular focus being map imaging itself for Phase I of this proposal. Phase II will focus heavily on increasing the data types collected while implementing the system in a CUBESAT configuration for use in Earth orbit or in deep space environments.

Stellar cartography CUBESATs will be designed to operate autonomously and independently to collect, analyze and return spatial data to ground stations. In addition, we intend to design the stellar cartography CUBESATs to be able to communicate with each other and operate in a swarm configuration, creating a space-based probe and cloud computing system capable of parallel processing in an autonomous, self-correcting node.

In short, we want to build autonomous mapping probes that can also serve as a basic communications and cloud computing infrastructure for distant worlds.

2.2 Relevance to NASA SBIR topic H6.22

The Phase I innovation Adder will be focused on is the creation of a deep learning algorithm that can ingest data and create spatial representations of planetary bodies, stars, and other objects of interest in a space exploration role utilizing self-learning inferential AI models.

In the process of creating this model, we first are addressing needs related to navigation and data collection. Our approach to this problem will involve the usage of neuromorphic/self learning AI systems and processes to parse, sort, and gain insights from the data to assist in other NASA goals.

In addition, the development needed for this project will require a number of novel approaches in using spatial data with neuromorphic processors and deep learning frameworks. It is therefore possible that we will find ourselves creating a new or updated processor specification for future use in this space-based integrated system.

2.3 Proposed Innovation Relative to the State of the Art

2.3.1 Software

At the time of writing, there is no current model of spatial deep learning system that has been developed for autonomous cartography or stellar navigation. The closest substitute for such a system could begin with a foundation in the Google Earth Enterprise project (GEE), which Adder has previously worked.

In fact, the open source globe framework provided by the GEE project will serve as the initial foundation for the Stellar Cartography Project. By utilizing this open source framework, Adder will save considerable time on the infrastructure side of this project, allowing us to dedicate as many resources possible to the actual neuromorphic algorithms that are the center goal of this proposal.

Our intention is to create a spiking neural network algorithm that can satisfy the NASA goal of autonomous navigation, using activation thresholds across numerous sensor nodes collecting and processing real time spatial data.

2.3.2 Hardware

The State of the Art listed for this subtopic is the Boeing Chiplet, with an intended readiness target of 2020. The purpose of this processor is to produce high float point integer operation (FLOP) performance in a low TDP hardware limitation, due to power and thermal considerations. The current hardware supports a dual quad-core ARM processor configuration to allow for parallel processing and fault tolerance.

Our proposal is to develop the SCP models to operate on minimal hardware requirements, however, it is not yet clear if the Boeing Chiplet is up to the task. The Adder team currently believes that advancements in high performance GPU computing will outpace the Chiplet’s capability, and that our system will benefit greatly from leveraging advancements from NVIDIA and Google Machine Learning.

If a development kit model of the Boeing Chiplet is available, we would be thrilled to run benchmark tests of all the potential hardware configurations that are available in order to determine the best chipset for the job.

Part 3 Technical Objectives

3.1 Spatial Deep Learning Model Development

The foundation for this project is the processing of geospatial data using a series of neuromorphic deep learning models capable of ingesting disparate and numerous types of geospatial data. The purpose of these models will be to organize, tag, identify, and otherwise classify geospatial data from low to high density data sets collected from numerous and variegated sources such as satellites, rovers, probes, and other data collection tools utilized by NASA and private commercial space enterprises..

The software and algorithms that are the foundation for our spatial deep learning AI will be the primary focus of the Phase I proposal, with an end goal of building map models from spatial data collected by numerous space based sources, communication of that data to a training database on a localized node, and further databasing to ensure fidelity.

In order to develop such a model, Adder has began working on implementations that utilize Keras and cuDNN that can refine and analyze image data using map tiles from ESRI and other private entities we acquire licensed data from as the foundational dataset. This deep learning model would constitute our current technical readiness level and provide us with an excellent framework to develop further.

In addition to image data processing, the SCP will also address spatial data and organize it based on the open source Google Earth Enterprise globe and its underlying code. Our intention is to use a hidden markov model algorithm to plot out previously unmapped areas utilizing all available data in a particular region.

For example, we collect $2.5x10^9$ GPS coordinate points per day from 50 million US based devices. We can feed this data to our algorithm to understand routing and movement. If Adder develops the proposed system, we can use GPS point data to determine the location of unmapped roads and paths.

AI mapping using existing datasets is already hugely valuable in itself, due to the nature of smartphone data collection, and this technology could solve challenges like mapping roads in countries where there has not been time or money allocated to doing so.

In an Earth-only deployment alone, this model will be of great value to numerous public and private organizations for mapping and logistical purposes. Some companies offer logistic and mapping services, often at a considerable cost to the consumer. We intend to offer our AI map services free of charge if we are successful in the creation of the system. Range finding and other sensor detection systems are a necessity in autonomous vehicles, and the usefulness of such a system for up to date weather, traffic, accident, and infrastructure reporting will be considerable.

It is our goal to complete this earth based model in Phase I, with a testable usage of the mapping system ready for early demonstration within 6 months of the start date.

Our secondary goal is to run this model effectively using a miniaturized computer with compact versions of common neuromorphic processing systems. This would be in preparation for Phase II, where we'd continue development of a system that could utilize these tools in space without large data transmissions to a ground-based station.

3.2 Integrated System POC Development

While our focus for Phase I is the development and training of a neuromorphic guided machine learning model, our Phase II goals will revolve substantially around the potential for deployment of this technology in a space-based environment. In order to reach that goal, Adder will begin preliminary development toward an integrated system capable of spatial data collection, communication of data sets and sensor readings, storage and databasing without external systems, processing, and interpretation.

By the end of Phase I, we plan to have a functioning prototype of this integrated system in order to demonstrate the viability of such a concept. Our test implementations will revolve around image data, however, we'd like to broaden the scope of data collection to include numerous data sources and types into our model for future use (i.e. onboard LiDAR sensors, infrared and UV sensors, etc).

Additionally, we intend to use machine learning to address signal processing should we be limited in our communications capability onboard a CUBESAT deployment of this system.

Part 4 Work Plan

4.1 Technical Approach

Adder's technical approach to the problem will be iterative and cyclical. While we like to avoid 'agile' terminology to prevent confusion, our development process could be described as an agile system. Our cycle starts with research, testing of existing source code bases (if any), development of the initial spatial deep learning models, and then rigorous testing to ensure operational capability.

We understand that a vast spectrum of unanticipated challenges come with space based operations. With this being a key concern, testing will be a high priority for our team. It will be the primary factor, aside from NASA's requirements, for evaluating system capability.

After testing of software systems has been completed, we will move into POC work for a lightweight, low power processor that can be utilized on the CUBESAT in order to run our AI models in a space environment. The detailed description of an integrated system beyond a POC lab test build can be found in Part 7.

4.2 Task Descriptions

4.2.1 - Software

4.2.1.1 - Data Collection

The first role of the software is to ingest spatial data. Broadly speaking, there are two types of spatial data being considered for the Phase I spatial deep learning model: numeric data and visual data.

4.2.1.1.1 - Numeric Spatial Data Interpretation

We plan to use GPS coordinate points to test numerical data in the model. Coordinate data can be used to not only generate reliable reference points for future use in other spatial datasets, it can also be used independently to gather insights about movement in an area, providing the movement being tracked has a GPS transmitter associated with it.

This would be particularly useful for mapping regions on Earth that are difficult to reach by conventional means, and not valuable enough for image satellite flybys to determine the locations of roads, walkways, and paths. Our ideal deployment to test this system would be in the mountainous regions of India that do not yet have reliable maps, but do have active cell phone towers with devices in the area. With the GPS points collected by devices in the area, road and walkway maps would render quickly and accurately using our model.

In order to utilize this software for smart city and private needs, this AI model would be crucial. It could inform public services where to allocate resources most effectively based on the movement inside a city. For a private entity, this system would be useful for understanding how people react and interact with physical infrastructure.

4.2.1.1.2 - Visual Spatial Data Interpretation

In a similar method to the numerical data model described above, Adder also intends to use images and other visual data to correlate with the numeric data. Telescope and satellite imagery would serve as source data to construct AI map information with, and allow for quick and autonomous exploration of distant worlds and other cosmic bodies. Additionally, we'd like to mount our own cameras and sensors on the SCP CUBESAT to ensure reliability and complete autonomy in collecting and processing this data into our spatial model.

Images of map tiles provided in our agreement with HERE Maps will be used to test image data in the model. This dataset includes several overlapping visual representations of Earth, and they can all be used as reference points/controls. Initially, this will be fit within a framework based on the open-source Google Earth Enterprise SDK, with which we are already experienced.

4.2.1.2 - Communication

In our Phase I development plan, communication with other satellites beyond other containers running our AI models will not be a priority. Instead, we will focus on communication of our data from a CUBESAT platform to a ground station via the NASA Deep Space Network of transmitters and receivers. Our control software will be written to communicate on three different frequencies to handle different tasks.

Further research is needed to determine the optimal frequency needed for satellite-to-satellite communications if we are able to continue into Phase II of the Stellar Cartography Project. We currently believe that JPL's work on the MarCO relay will be instrumental in guiding this research and discovery process.

4.2.1.3 - Databasing

Databasing of spatial data can be a very CPU heavy process, depending on the speed at which data needs to be inserted. In a single CUBESAT configuration, however, we can reliably predict that this will not be a significantly limiting factor.

In the Phase I software development process, we intend to use non-relational database tools that we have developed for the processing of GPS data, and will expand the capability of these tools to suit this application as we move forward into image and other spatial data.

In software testing, we do anticipate challenges shielding the storage media from solar radiation and other interference that may corrupt data, however, there are many existing solutions that could address this problem.

4.2.1.4 - Processing/Training

Initial processing and development will be completed on machine learning workstations at Adder, iterating rapidly to identify weaknesses in the variable set and machine learning model we are using. At the time of writing, we intend to use a hidden-markov chain model with layers of data, finding midpoints between those data layers, and extrapolating those averages.

The results of this extrapolation are then used in the model for repeated interpolation as new epochs of learning begin. This can take considerable time with workstation-grade GPUs and is the reason we need additional GPU infrastructure to avoid untenable cloud GPU fees that would otherwise be accrued in our training process.

When millions of dollars of mission critical data and materials can be at stake, we'd like to have the ability to train our models without being overly concerned with going over budget on Amazon Web Services.

4.2.2 - Fault Resistance & Redundancy

There are several layers of software fault resistance and redundancy systems that will be implemented in order to reduce and eliminate fatal errors, such as kernel panics or invalid memory access. These layers are designed to minimize the number of potential errors that may occur, as well as to coordinate and assist individual components in self-correction when an error has been detected.

The overall system is made up of several individual components, each of which is responsible for a very specific task. This specificity simplifies the design of each component, and allows for very thorough error handling. Any given component will be able to correct its state, in the case of a non-fatal-error, buffering additional input while the component recovers, and using systems such as journaling and write concerns to ensure that any buffered input is properly processed upon resuming normal operation. In the event of a fatal error (i.e. an error which places the component in an unknown, unrecoverable state), each component will thoroughly log all relevant data (including its state at the time of the fatal error), for later analysis. Once the logs are written, the component immediately restarts itself, ensuring that it is in a stable state before it continues normal operation.

By the design principle of modularity, no two components will be able to communicate directly to each other. Instead, they will utilize a series of message queues to communicate in a general fashion. Should a message queue become unavailable, only components that are subscribed to that queue will be unable to perform their functionality, and will enter a sleep mode until they detect that the queue has recovered from its invalid state. The usage of message queues rather than a monolithic controller minimizes the likelihood of the overall system entering an unknown state, simplifies error identification, analysis, and correction, and eliminates the inefficient and useless process of state persistence and reinitialization of components that are still functioning properly.

While each component will be capable of recovering from most errors on its own, there are fatal errors which may occur that a subsystem cannot recover from on it's own. We plan to implement a series of 'watchdog' daemons, which periodically assess a portion of the systems state to ensure it is valid, and coordinate with each other to correct any issues. These daemons will be designed as simply as possible, to reduce the potential errors that may occur, and will coordinate and assist components in recovering from fatal errors.

Who watches the watchers? In order to ensure the daemons maintain a consistent state, a master daemon will run in the background, constantly assessing the condition of each daemon. Should the master daemon determine that a single daemon (or series of daemons) are in an invalid state, the master daemon will correct the issue and allow the system to continue on. Due to the simple nature of each individual watcher daemon, the possible combinations of errors that may occur become quite small, enough so that the master daemon will be able to quickly and easily recover. In the event that the master daemon cannot determine a proper course of action, the master daemon will initiate an immediate emergency shutdown of individual subsystems, if it can be done without adversely affecting the data integrity of the overall system. Should the master daemon fail at correcting the state of the subsystem, it will initiate an emergency shutdown and restart of the entire system into a safe mode, verifying and recovering the state to the best of its ability, analyzing and storing log information about unrecovered errors or possible data corruption, before restarting into a normal operational mode.

Each component, message queue, watcher daemon, and the master daemon will have additional redundancy systems running, and will ensure state consistency via journaling, read/write concerns, and other backup mechanisms. Should one system fail, a myriad of other systems will be able to ensure continued operation while the system restarts itself. It will take a very fatal error, one capable of causing a multitude of systems to fail nearly simultaneously, in order to bring the system to an unknown state and force a reboot into safe mode. We anticipate that with the proper hardware design, the software will be able to operate for an extended period of time before a routine maintenance reboot will be required.

4.2.3 - Testing

4.2.3.1 - Functional Tests

Functional Testing will cover all aspects of system functionality, including all of the aforementioned subcomponents that comprise the SCP CUBESAT.

When the SCP and system control algorithms have been implemented, each component will undergo heavy unit and stress testing to ensure that the logic is working in a proper fashion. Each component will be highly analyzed and carefully designed to handle all errors that may occur within itself, and will contain logic which allows each component to self-correct without placing additional strain on other components. In the event of a fatal, unrecoverable error (i.e. an error which places the system in an unrecoverable unknown state), the component will signal the controller that it has failed, and restart itself.

4.2.3.2 - Planned Phase II Tests

Random Vibration
Radiation/EM Exposure
Frequency

Pyroshock Test
Thermal Cycling
Thermal Vacuum
Database Fidelity Testing

4.2 Meeting the Technical Objectives

Our team is familiar with all of the proposed Phase I work laid out, with an exception to building the hardware into a CUBESAT itself. We have already developed deep learning models, lightweight databases, and unique sensing hardware for use in harsh environments. We have previously completed design work under subcontract (for IRST, contracted by the USGS) to add GPS location, 3G communication, and atmospheric sensors onto a submersible water monitoring device.

For Phase II work, we have hired additional staff to help address concerns in orbital mechanics and movement, as well as additional hardware designers for research work on new and unique processor designs/architectures.

4.3 Task Labor Categories and Schedules

Tasks will be separated into focus areas for our internal development teams. In our current plan, work focus areas are split into machine learning, software design, hardware deployment, and testing/diagnosis.

4.3.1 - Machine Learning Design/Training

The machine learning category will have the most time allocated to it, and will be a continuous process into Phase II due to the nature of AI training. AI training of existing models is a task that our entire team is capable of completing.

Brendan Elliott, MS, will be leading the machine learning team, and Ian Gerard will handle day to day devops. Jeremy Fox, Brad Hoffman, Kevin Wiley, and Tyler French, MS, will also be key staff in designing protocols, APIS, and scripts. Dr. Timothy McDonald, Dr. Rajiv Uttamchandani, Lillie Beiting, and Ian Gerard will be tasked with database structuring, information architectures, and collection/sourcing of training data.

4.3.2 - Software Design

Software design will begin 2-4 weeks after the start of the project, following the research necessary to implement the machine learning models in the power constraints set by CUBESAT solar panel limitations. POC software will be completed within 90 days of the start date to give our teams time to then deploy the spatial deep learning models onto the proposed hardware for benchmarks and testing.

4.3.3 - Hardware Development

Hardware deployment testing will begin 4-6 weeks after software design begins, with a

POC system taking roughly 1-2 weeks to assemble. The hardware development and testing process will become iterative at this point, as hardware and software teams zero in on the necessary requirements and changes that need to be made to the POC system.

4.3.4 - Testing

Testing and diagnosis are ongoing processes that begin at the start of software development, and will continue until the completion of Phase I. For specifics on what tests will be completed, see section 4.2.3.

Part 5 Related Research/ Research and Development

From our initial research and development work with mini-GPU high performance computers in Phase I (NVIDIA's Jetson, for example), we believe that the NVIDIA CUDNN will be an optimal neuromorphic training environment for our spatial data. As we get further into the process of containerizing the models, we'll be able to determine key limitations and issues with this hardware solution.

As these limitations are uncovered and further understood, we intend to begin designing our new chipsets to address issues with the current state of the art, with a key focus on fault tolerance, redundancy, and shielding from cosmic rays, dust, and other astronomical particulates. It is generally accepted that ARM or NVIDIA will likely be the end manufacturer of whatever designs we intend to prototype.

We also intend to work in power optimizations that will utilize specialized parameters set by our designers to make the most of what power is available on a CUBESAT. Additionally, there are several Kentucky-based startups that are working in graphene battery design to increase efficiency and storage capability, which could be very useful for this project if the weight to performance ratio works to our advantage.

Communications optimizations will be a late stage concern that we likely will not start work on until Phase II, given the ability to use the NASA DSN which already has established protocols.

Part 6 Key Personnel and Bibliography of Directly Related Work

Ian Gerard - Project Leader

Related Work: Ian Gerard worked in geospatial data plotting and cartography with Leica Geosystems/Hexagon before founding Adder Mobile Technologies. His largest project managed to date is the recently completed Adder 2.0 project which involves massive geospatial workload processing and analysis using GPU and CPU high performance computers purpose built and specialized for our applications.

Ian Gerard also designed Adder prototype hardware including the DCDS I & II -- a series of mobile screens designed to change states when in different geographic locations. Gerard also designed and built Prometheus I & II -- a set of specialized spatial deep learning workstations that utilize cryogenic cooling methods to achieve incredible performance on low cost hardware.

Brandon Bush - Software Development Lead (Master's Degree)

Related Work: Brandon Bush was the lead developer on the Adder 2.0 GPS analytics backend and overall project, with an emphasis on databasing and hardware operations pertinent to the geospatial data collected. Breakthroughs on the project included near-instant querying of data sets with over 2.5 billion GPS coordinates and analysis/inference of nearby device locations in a non-relational architecture. Bush also created the software needed for the DCDS I & II screen technology to be spatially aware and properly oriented.

Lillie Beiting - Data Analysis Lead (BA)

Related Work: Lillie Beiting has specialized in ROI attribution models across multiple software systems, with an emphasis on data integration and database architecture. Her largest project managed to date involved serving as the chief architect in the restructuring of a $5billion federally-regulated database in preparation for the application of Machine Learning.

Dr. Rajiv Uttamchandani - Neuromorphic Processing Lead (PhD)

Dr. Rajiv Uttamchandani is an astrophysicist with a research emphasis on the relationship between sunspots and magnetic fields in the sun’s photosphere. With a focus on alternate materials for dielectrics in modern transistors, Dr. Uttamchandani has provided insight into low-cost, high-performance capacitor optimization for use in novel processor architectures for Intel.

Dr. Timothy McDonald - Data Procurement Lead (PhD)

Related Work: Dr. Tim McDonald has an extensive background in data acquisition.

Brendan Elliott - Geospatial Machine Learning (Master’s Degree)

Related Work: Brendan Elliott served seven years as a senior developer for ALK technologies (today: Trimble Maps) in geospatial mapping and route optimization, eventually rising to Director of Software Development for their company. Brendan also worked in machine learning and large scale fleet logistics while with ALK.

Kevin Wiley - Hardware Development Lead

Related Work: Kevin Wiley is highly experienced in working with x86 and ARM architectures at the kernel level, giving our team the ability to operate as close ‘to the rails’ as possible. His extensive work in blockchain will also be useful in developing hardware that can self correct and validate in complex tasks such as mapping and navigation.

Tyler French - Machine Learning & Dev Ops (Master’s Degree)

Related Work: Numerous Kubernetes and Docker implementations make Tyler French a critical component of helping our team work on the right AI and deep learning models. He assists in the maintenance of version control and virtualization of our own models, we ensure maximum training efficiency.

India Stewart - Astrophysics Specialist (Master's Degree)

Related Work: India Stewart holds a degree in astrophysics, and will be instrumental in assisting the team with operational considerations that are outside the scope of hardware engineers and deep learning specialists.

Brad Hoffman - Parallel Processing Developer

Related Work: Brad is currently an MS Candidate in Computer Science at the University of Louisville and has contributed heavily to the Adder 2.0 GPS analytics backend and overall project, with an emphasis on databasing and hardware operations pertinent to the geospatial data collected. Brad Hoffman specifically has a substantial interest in developing novel programming applications and approaches, along with Brandon Bush and Kevin Wiley.

Alex Gerard - Orbital Mechanics Advisor

Related Work: Served 10+ years as a pilot and instructor in the United States Air Force. During his service, Alex Gerard went on to complete advanced-level coursework in orbital mechanics and space based navigation. He has designed numerous orbital simulations and can provide essential insight into the challenges of deploying the mapping and navigation software we intend to create.

Jeremy Fox - Communications Hardware

Related Work: The Adder 2.0 GPS analytics backend and overall project. Jeremy Fox has also worked extensively in hardware troubleshooting and fault testing from his time as a circuitry technician with Harmon/Kardon.

Part 7: Relationship with Phase II or other Future R/R&D

This Phase I proposal is meant to serve as an R&D test period for our team to complete our technical objectives that are essential to continuing to a Phase II proposal. With the Stellar Cartography Project deep learning models in a workable state, we intend to have completed prototype, lab testable software and an early hardware solution capable of running the SCP deep learning models in low earth orbit, operating within the Cubesat's limitations on power and thermals.

Our Phase II proof of concept hardware considerations as this system develops can be separated into five distinct categories: main board, antennas, sensors, storage, and power supply.

3.3.2.1 - Main Board

The main board, or motherboard, will consist of a processor and necessary inputs to match or exceed the capabilities of the Boeing Chiplet, the current SOA.

For POC Hardware implementations, we plan to test the Raspberry Pi Model 3, Arduino Uno, NVIDIA Jetson TX1, NVIDIA Jetson TX2 & TX2i, NVIDIA AGX Xavier, NVIDIA Jetson Nano, ARM Mini and the Google Coral board.

The key concern will be the ratio of power to performance while operating within set CUBESAT parameters. These parameters can be very restrictive, which is why it is of critical importance to run the least intensive software while maintaining mission capabilities of data collection and deep learning interpretation of spatial data.

3.3.2.2 - Antennas

In order to effectively route communications and to leverage a network of SCP CUBESAT platforms, multiple communication bands will be needed to complete the required tasks. In our initial design plan, we will need three communication nodes for three distinct purposes: earth to space instructions/space to earth data downloads, space to space, and interruptions/emergencies.

Utilizing NASA's Deep Space Network (DSN) will solve a majority of communication challenges if we are limited to only one antenna per unit. Ideally, with a two antenna configuration, a communications node similar to JPL's MarCO Relay could be established for co-processing/parallel processing/cloud computing between spatial learning units.

It is also feasible, and potentially very cost effective, to consider using the MarCO relay or a similar system as a centralized node to link any data feeds to the SCP CUBESAT.

3.3.2.3 - Sensors

Sensor data does not need to be collected by any deployed version of the SCP deep learning/AI model in order to utilize its capabilities. Instead, data can be sourced from other collection devices and transmitted to the SCP CUBESAT for storage and processing on site.

In a trial run, however, it seems wise to test basic sensors and cameras on the same physical CUBESAT as the data processor/main board component to ensure completion of the primary objectives. This added redundancy to the system will decrease overall mission risk and provide additional failsafes should there be unforeseen issues, with communications, for example.

3.3.2.4 - Storage

Storage devices on the CUBESAT will operate on the lowest power requirements possible to allow for other processes to run more efficiently. Weight is also a factor worth considering, as multiple terabytes of data stored on disk drives would add up quickly.

Our plan is to operate NVMe storage or another SSD based solution if it can be shielded properly to ensure database fidelity and quick accessibility by the spatial deep learning models.

3.3.2.5 - Power Supply

As with many other CUBESAT configurations, we are operating within limiting parameters provided by EnduroSat's solar panels on a 6U configuration. While a 6U configuration may be oversized for internal component storage, the power generation capabilities of the 6U CUBESAT solar panels is needed in our current calculations for onboard power.

We will spend part of our research phase investigating graphene battery and other high efficiency storage solutions that could work for this proof of concept. Because our Phase II plan focuses heavily on CUBESAT hardware, these concerns will be addressed in that plan.

Additional testing or testing that could not be completed in Phase I will be completed in Phase II. See the work plan in section 4 for more details.

Part 8: Company Information and Facilities

Company Information

Adder Mobile Technologies, Inc., is a Kentucky corporation with 16 employees. Adder was founded on June 1, 2017 by Ian Gerard, with previous experience working for Hexagon/Leica Geosystems, and Collin Taylor. The Adder team has grown considerably since then, now including specialists in machine learning and hardware development. Adder also has an extensive professional network to draw talent and insight from, should additional resources be needed for this project.

Our non-SBIR company research and development focuses on real time and post-processed GPS data insights and analytics and how they relate to customer behavior and non-digital, outdoor advertising measurement. We monetize our advertising related services in a number of models and sales channels.

Adder is also beginning to develop projects that are intended for open source distribution, with a strong interest in providing open source access to our insights and models pertinent to cartography and GPS location finding.

Facilities Available to Adder

Our primary facility is located at 10482 Bluegrass Parkway, Louisville, Ky, 40299. We co-locate our hardware and software development in this facility, with a ~1000 sq. ft. production warehouse for our prototype and small-run production. At this location, we can provide cloud services via our fiber-enabled servers, 3d printing, hardware prototyping, and small run circuit assembly.

Specifically, the Adder HQ facility features internal and external gigabit connections, over 225 TB of onsite data storage, and roughly 500 teraflops of high performance computing ability. These machines have room to be scaled up and are at about 50% capacity. We also have hardware development capabilities tailored toward independent microcontrollers such as the Raspberry Pi, Arduino, Nvidia Jetson, and others. This facility is optimal for software development and AI training on the base data sets.

The secondary location in Richmond, Ky, serves as the base of operations for PCB production, reflow/SMD circuits, non-essential servers. This location can also fulfill 3d printing needs, and hosts backup storage for some of our servers.

Our facilities do not currently feature certified clean rooms, however, we have access to two clean room facilities that would be cost effective for us to lease. We do not anticipate a continuous need for clean room areas after this project, and do not currently intend to adapt our HQ to include a clean room. If a clean room environment is required in a Phase II proposal, we have drawn up plans and budgets for an improvement to our facility at 10482 Bluegrass Parkway, Louisville, KY.

If this facility cannot be brought to the standards that our engineers require for fulfillment of a potential Phase II SBIR proposal, we will seek a solution that can fulfill this need, however, we do not anticipate this outcome.

Part 9 Subcontracts and Consultants

Our SBIR Phase I scope of work and development plan does not currently involve any outside contractors. All work done on this project will be completed by the employees of Adder Mobile Technologies, Inc.

We will consider and research the possibility of using outside contractors on this project for a Phase II implementation. We will also pull from outside resources in our professional network if we encounter challenges that require additional support in Phase I. We believe that the areas of databasing and processor chipset design will require additional manpower in Phase II, but for Phase I we are confident the team we have assembled can complete the technical objectives.

There are two primary reasons for this decision:

1) The team we have established is diverse and multi-disciplined, with skills that compliment all aspects of this project. We have already started work in machine learning approaches that can help sort and gain insight in the fields of GPS, navigation, and spatial learning.

2) In our previous experience contract firms and laborers have lacked the quality and work documentations standards demanded by Adder. We’ve also had numerous issues with contractors missing deadlines and would like to work with our own internal team and add advisors on as needed for high-level concept work.

Part 10 Potential Applications

10.1 Potential NASA Applications

Our general work in spatial deep learning, packet compression, and miniaturization of these processes has substantial overlap with NASA’s interests. In fact, we began outlining the concepts related to this proposal before we were even aware there was an SBIR focus area addressing deep learning applications in space.

More specifically, there are a number of applications for a swarm-based neural network deployed in orbit around numerous planets. On Earth specifically, AI interpolated spatial data can be used in numerous analytical models.

In space, however, the potential is almost as endless as the cosmos itself. A CUBESAT swarm of these systems could serve as a survey probe system to evaluate a planet or asteroid for colonization/resource extraction. Also, as previously mentioned, if our system can be paired with a high energy communications relay such as the JPL MarCO, the system could serve as a communication satellite and cloud computing infrastructure.

This system could also be used in high earth orbit to detect and track debris and objects of interest, assisting in navigation and cleanup of defunct hardware.

10.2 Potential Non-NASA Commercial Applications

Non-NASA applications could include mapping of viable space travel routes and launch windows in an open-source capacity. This means that we would offer our mapping services (APIs and SDKs) to private and public groups, spurning development and innovation in space navigation and mapping technologies, offering complete access in exchange for any relevant spatial data fed into our neural models.

More specifically, we believe that this open source mapping system could be used to solve problems in transportation, logistics, resource access, and more by giving underserved regions tools to accurately map and plot terrain that has previously lacked those resources.

Part 11 Similar Proposals and Awards

11.1 Previous Awards

Adder has not previously submitted any previous proposals, nor has Adder sought any SBIR/STTR awards up to this point. Adder has not been issued any grant funding whatsoever and is coming into this SBIR Phase I process with a clean slate and very few complications to address.

11.2 Other Awards Being Sought

Adder will be working diligently to submit additional proposals related to this topic and to research/development areas our team has experience and interest in. Specifically, we intend to pursue SBIR interest areas related to swarm communication, blockchain, image processing, and GPS/spatial data mapping.

If any additional proposals are submitted, we will update this proposal immediately after doing so.

References

Coral Beta. (n.d.). Retrieved March 29, 2019, from https://coral.withgoogle.com/

NASA. (2017, October). *CubeSat101 Basic Concepts and Processes for First-Time CubeSat Developers* [Press release]. Retrieved March 29, 2019, from https://www.nasa.gov/sites/default/files/atoms/files/nasa_csli_cubesat_101_508.pdf

Google. (2019, March 28). Google/earthenterprise. Retrieved March 29, 2019, from https://github.com/google/earthenterprise

Jetson Nano Developer Kit. (2019, March 28). Retrieved March 29, 2019, from https://developer.nvidia.com/embedded/buy/jetson-nano-devkit

Mission Introduction to the DSN [DSN Mission Support Definition & Commitments]. (n.d.). Https://deepspace.jpl.nasa.gov/files/Intro_4_Mar20151.pdf.

NVIDIA JETSON AGX XAVIER MODULE A NEW AI MILESTONE FOR AUTONOMOUS MACHINES [PDF]. (n.d.). NVIDIA.

NVIDIA Jetson TX2System-on-ModulePascal GPU ARMv8 8GB LPDDR4 32GB eMMC WLAN/BT[PDF]. (n.d.). NVIDIA Corporation.

NVIDIA® JETSON™ TX1 SYSTEM-ON-MODULE FULL-FEATURED MODULE FOR INTELLIGENT DEVICES [PDF]. (n.d.). NVIDIA.

THERMAL TESTING and VERIFICATION [This lesson will explain the various kinds of thermal testing performed on spaceflight hardware and the objective of each type of test.]. (n.d.). Https://nescacademy.nasa.gov/review/downloadfile.php?file=ThermalTestingPart1.pptx&id=243&distr=Public.

Thermal-Vacuum Testing [Description of Thermal-Vacuum Testing Facilities available at Johnson Space Center]. (n.d.).
Https://www.nasa.gov/centers/johnson/pdf/639724main_Thermal_Vacuum_Testing_FTI.pdf.