2020 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
AUTONOMY, ARTIFICIAL INTELLIGENCE & ROBOTICS (AAIR) TECHNICAL SESSION
AUGUST 11-13, 2020 - NOVI, MICHIGAN

An Open Data Architecture for Ground Vehicle Data-driven Autonomy Development and Validation

Michael Boulet[1], Tate DeWeese[1], Andrew Bird[2], Ryan Kreiter[3], Calvin Cheung[3]

[1] MIT Lincoln Laboratory, Lexington, MA
[2] Neya Systems LLC, Framingham, MA
[3] U.S. Army CCDC Ground Vehicle Systems Center, Warren, MI

ABSTRACT

Modern autonomy development relies on stored data to train and validate the performance of algorithms and models. However, the community developing autonomous ground vehicles for national defense lacks readily available datasets that adequately cover the landscape of anticipated operating environments. We propose the development of an open architecture and supporting infrastructure enabling scalable and effective collection, storage, processing, and reuse of the U.S. Army's autonomous ground vehicle data across numerous stakeholders and programs. This paper presents the proposed architecture's requirements, use cases, and a preliminary design. We also show results of an initial prototype implementation performing a query task on existing ground vehicle sensor data.

Citation: M. Boulet, T. DeWeese, A. Bird, R. Kreiter, C. Cheung, "An Open Data Architecture for Ground Vehicle Data-driven Autonomy Development and Validation", In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 13-15, 2020.

1. INTRODUCTION

Data are increasingly becoming critical assets required to rapidly and efficiently develop advanced autonomy in future ground vehicles. Data are used throughout the autonomous ground vehicle development lifecycle. Offline data, such as raw sensor measurements recorded during field experiments, are often provided as input to software modules to evaluate and improve their performance without requiring time-consuming hardware cycles. Stored data from a diverse set of environments can be used to validate the robustness of an algorithm across anticipated operational conditions. Furthermore, with the emergence of modern machine learning techniques, developers leverage stored data to train deep neural networks to achieve superior levels of performance relative to traditional human-designed algorithms.

Data availability tends to greatly accelerate autonomy innovation and maturation. For example, the publication of labeled images in the academic community facilitated improvements in classification accuracy to achieve parity with humans in a few years [1]. Despite the evident utility of data at the ready, the national defense autonomous ground vehicle community lacks available datasets of relevant environments. While individual programs typically collect significant volumes of autonomous ground vehicle (AGV) data during tests and demonstrations, often at significant cost, the data are rarely shared outside of the program or platform owner.

A present-day example of data-driven development practices highlights current challenges. A Government laboratory researcher may be aware of a potentially useful artificial intelligence capability but is uncertain of its applicability to U.S. Army AGV data. In order to assess the performance of the technology, the researcher must first collect, clean, and transform relevant training and validation data. Without a shared data resource, the researcher might initiate a standalone field data collection campaign which, through significant cost and time expenditure, may generate the needed data. However, the data is likely only a narrow sample of possible operating environments which may lead to flawed analyses. Alternatively, the researcher may seek out previously stored data by contacting colleagues across the community. If found with suitable data rights, the data must be transferred to a hard drives, shipped to the researcher, uploaded to the researcher's computer, assessed, transformed, and cleaned before any AI-based development occurs. Thus, the cost and risk of developing, or even assessing, AI and other data driven algorithms today is significant.

The Autonomous Mobility through Intelligent Collaboration (AMIC) program is developing an open data architecture, guidance, and infrastructure enabling the national defense community to effectively collect, share, and leverage autonomous ground vehicle data. The AMIC open data architecture is aligned with the DoD Modular Open Systems Approach (MOSA) and U.S. Army Data Strategy principals to accelerate autonomy innovation, maximize reuse, and enable competition decoupled from vehicle ownership through shared, readily available, interoperable, protected, and trustworthy data.

1.1. Related Work

Several publicly available datasets, including the KITTI dataset [2], contain LIDAR point clouds, color camera images, global positioning, and other sensor measurements collected from vehicles traveling on public roadways. These datasets are provided as static files with metadata, including a description of the data format and extrinsic calibration information, captured in an accompanying paper. The data volume can be significant, reaching 180 GB for the KITTI dataset. Researchers have leveraged the publicly available data to develop and evaluate a diverse range of capabilities as evidenced by over 4000 references to the KITTI dataset paper.

In [3], Nelson identifies the static nature of these existing datasets as a key challenge and describes a data management framework enabling efficient access and additions to the dataset. An implementation of the data framework is used to store 20 TB of sensor measurements, yet also return specific data values with sub-millisecond access times.

Key distinctions between the prior work and the AMIC system include the need to accommodate a wide variety of data sources, as the system must accept data from platforms that are controlled by external entities, and a focus on data from anticipated military operating environments, including data collected from navigation of unimproved roads and off-road terrain.

2. REQUIREMENTS

The design of the AMIC data architecture is informed by the data challenges currently faced by AGV algorithm developers, anticipated future use cases, and U.S. Army data policy.

Scalable: To capture a representative dataset for military AGVs, thousands of hours of sensor data must be collected in the many environments and conditions in which future AGVs may operate. Currently, a typical autonomous ground vehicle will produce tens of megabytes of sensor data per second with compression. Therefore, the data architecture must be able to support a data volume reaching tens of petabytes.

Accessible: The collected data must be readily available to authorized developers and stakeholders with minimal friction and overhead. Efficiently using the collected data requires an ability to index, query, and retrieve specific data records to avoid an expensive linear search through the accumulated data store.

Reusable: Extending the use of data beyond the specific purpose for which it was collected requires capturing semantic information and descriptive metadata associated with each data record, such as calibration parameters. Additionally, the results of any processing applied to raw data shall also be available for reuse to avoid repeating expensive computations.

Validated: As the quality of data directly impacts the level of effort required to extract utility from it, the AMIC framework must provide metrics for assessing data readiness levels per Lawrence [4]. Additionally, the data architecture must support data lineage and traceability.

Interactive: The data architecture shall support rich visualization and processing flexibility to enable iterative query and algorithm development.

Protected: Data, while at rest and when transmitted between systems, must not be accessed or interfered with by unauthorized entities.

2.1. Data Types

Supporting the development of a diverse set of AGV capabilities, including future applications that are not yet defined, requires a heterogeneous data architecture capable of storing, querying, accessing, and processing a variety of data types.

Time series vehicle-hosted sensor measurements are a core data type considered by the AMIC framework. The system must support common sensor data types and formats, e.g. frequently used Robot Operating System (ROS) messages, and provide capability for interpreting unknown data types with custom processing. The architecture must also contain information needed to transform and process vehicular sensor measurements, such as the sensor's configuration, position and orientation relative to the host platform, and calibration. Anticipated use cases may also require access to relevant contextual data, including the host vehicle class and configuration, mission tasking, crew communications, and, for autonomous host vehicles, autonomy mode and other internal state data.

While a repository of raw spatiotemporal ground vehicle sensor and related data would provide standalone utility, important use cases require the storage and access of additional types of data. For example, a machine learning engineer developing an algorithm to match ground vehicle sensor data to overhead imagery would also need access to a dataset of overhead images. Additional types of data with potential AGV relevance include terrain elevation maps, images with associated object detection or semantic segmentation annotations, point clouds from UAV-based photogrammetry, weather history, or road networks. Furthermore, the framework must also support storage of processing output, such as learned features or data transformed into alternative types, to facilitate reuse of potentially expensive computing steps.

2.2. Use Cases

While AMIC will provide utility to a large variety of applications, several specific use cases serve as motivating examples for assessing and testing initial AMIC prototypes.

Formation Control: Developers will leverage AMIC to apply machine learning technologies, such as inverse reinforcement learning, to develop single and multi-vehicle planning algorithms from vehicle control and sensor data collected from human-driven traversal of complex environments.

UAV-based Perception for Ground Navigation: Developers will use AMIC to build algorithms that assist ground vehicle navigation tasks, for example planning beyond the ground vehicle's sensor range and GPS-denied localization, from data collected by airborne platforms.

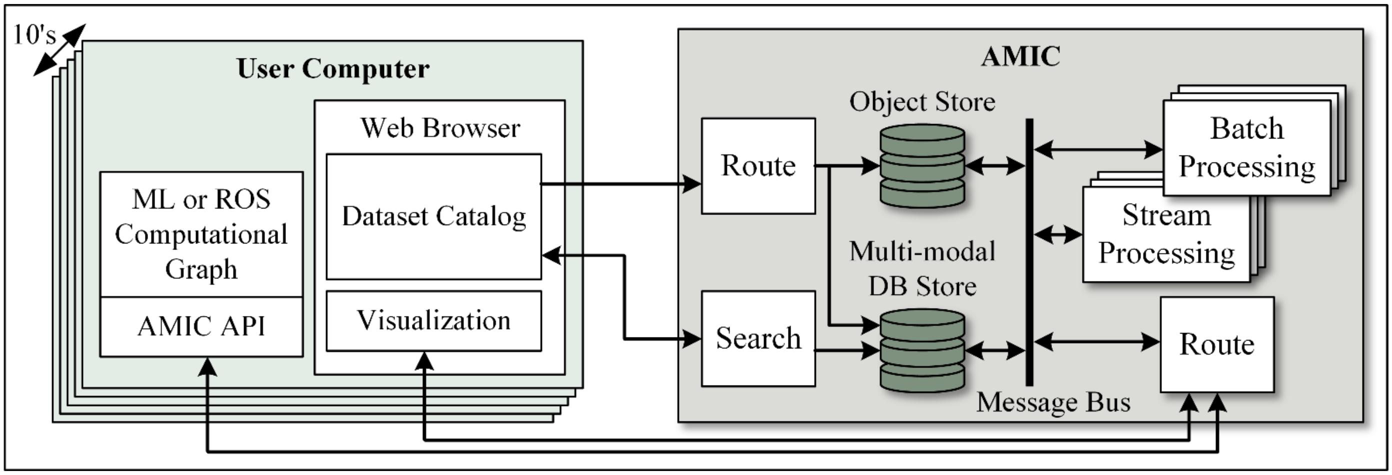

Figure 1 *The AMIC open data architecture consists of data storage, processing components, and tooling supporting flexible and efficient data –driven algorithm development and validation.*

3. ARCHITECTURE

Analysis of prior work, available technologies, and the described use cases and requirements informed the development of an AMIC storage and processing architecture described below and shown in Figure 1. The architecture is intentionally aligned with both open source and commercial big data product lines to facilitate implementation flexibility.

3.1. Data Stores

Data storage and access is a core function of the AMIC framework. A poly-store architecture with characteristics of both an unstructured data lake and structured data warehouse provides for the high degree of variety inherent in AGV data.

Unstructured Data Store

Most autonomous ground vehicle sensor data, by volume, are inherently unstructured. That is, the range readings from LIDAR sensors or pixel values from on-board cameras do not typically provide information that can be meaningfully used in relational tables or similar schemas without additional processing. Even when unstructured data can be cleaned and exported to a structured format, retaining the native unstructured form allows for future reprocessing. Researchers may, for example, improve processing algorithms or identify new applications for the stored raw data. Furthermore, AGV data are frequently recorded to disk as serialized sequences of atomic records, such as bag files in the ROS ecosystem. Natively storing these files as unstructured data is space efficient and enables the use of existing robotics-specific tooling for data inspection and replay.

The AMIC framework stores unstructured data in a distributed object store service. Each binary object is assigned a unique ID that is associated with content type, ownership, and additional metadata maintained in the structured databases described below. The system supports objects of arbitrary type, including data formats that are not yet supported by other AMIC processing tools. Users may upload data to the object store using a web-based interface or through a REST API. Additionally, algorithms executed on the AMIC processing platform may store results back to the object store.

Structured and Semi-structured Data Store

In addition to the object store service, AMIC integrates relational and document-oriented database systems for rapid query and analysis of AGV data that fits within structured or semi-structured schemas. For example, the message database maintains a record of the object ID and byte offset for each ROS message contained within

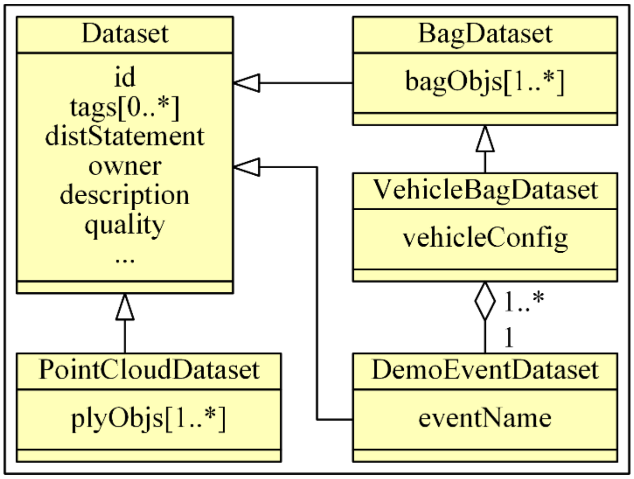

Figure 2 *A hierarchy of dataset classes allows users to search on common metadata and then refine the query based on derived class parameters, such as the name of a particular demo event.*

the set of bag files in the object store. Data consumers, such as an algorithm processing LIDAR scans, can use the message database to rapidly retrieve messages of a specific type from the object store without needing to scan through every bag object. Data ingest and other AMIC processors may create new databases, alter existing database schemas, or append records to existing instances. An image classification algorithm would link detected labels to images with a row in the annotations table, for instance. Additional information maintained by the system includes object metadata, datasets, jobs, and data lineage.

3.2. Datasets and Dataset Catalog

A core concept for organizing and presenting data within AMIC is the “dataset”. An AMIC dataset associates a collection of data with contextual, ownership, and quality metadata. Required metadata fields include a dataset description, distribution statement, list of tags, and data readiness level. To aid rapid search, some datasets may also contain summarization data, such as the temporal and spatial boundaries of data contained within the dataset.

In order to support the many types of data used in AMIC, a class hierarchy of dataset types are maintained in an object-oriented database. As shown in Figure 2, the abstract dataset type contains required metadata and the derived classes include specific types of data, such as reference to bag file objects and additional metadata. Processing pipelines leverage the structure provided by derived dataset types to facilitate reuse. For example, a visual inertial odometry algorithm may leverage calibration information contained in the vehicle configuration structure along with images contained in bag files from the object store.

The dataset catalog provides a web-based user interface for browsing, searching, and selecting datasets based on user-defined query parameters. The interface also provides features to visualize the data to help users assess whether the dataset fulfills their objectives.

An important use case for datasets is the creation and sharing of high-quality curated data. Researchers performing processing and analysis to transform, clean, and label raw sensor measurements may collect the resulting data into a dataset marked with a high data readiness level. Other researchers may then leverage the dataset catalog to search for datasets meeting a minimum quality rating.

3.3. Query and Data Processing Engine

Data stored in the AMIC framework may be returned to the user for downloading and processing on their local system. However, the time required to transmit data over the network may impose a significant limit on capability development velocity. Therefore, the AMIC architecture provides a platform and API for executing processes on computing infrastructure co-located with the data stores. Researchers and other users may construct processing chains in a high-level language that integrates custom algorithms with AMIC data access functions. Submitted tasks are then scheduled and executed on the AMIC-hosted computing infrastructure with internal message busses providing high-bandwidth low-latency access to stored data.

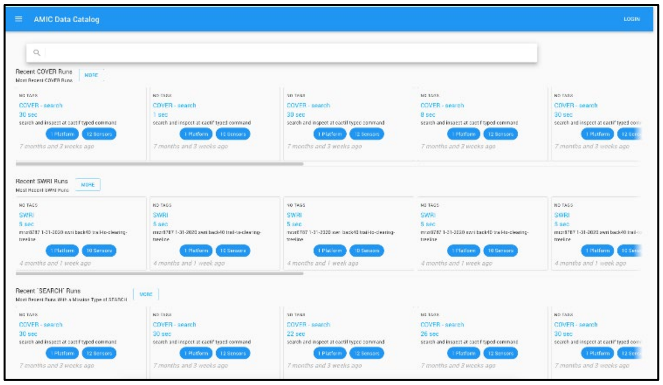

Figure 3 *The AMIC dataset catalog prototype allows users to search and view data using a web browser.*

4. DATA COLLECTION

While AMIC is designed to accept data from a variety of sources, dedicated data collection kits deployed on human-operated ground vehicles will provide a significant volume of initial data. Processing embedded within the kit receives, timestamps, compresses, and stores sensor measurements on local storage as the host vehicle travels through the environment. Timestamps are GPS-synchronized to facilitate association with data collected from other sources. Additionally, each data collection kit embeds calibration information and other mission metadata into the data stream. Data from each vehicle are offloaded to a large-scale on-site data lake upon the vehicle's return to a central depot.

AMIC will also leverage physics, visual, and sensor simulation engines to augment data collected in the physical world. Simulation offers precise control of conditions and scenarios, can be less costly than field data collections, and, by leveraging many computers, can easily scale to produce a large data volume.

5. DEPLOYMENT AND ACCESS

The AMIC architecture is designed to be hosted on horizontally scalable computing systems, such as those available from commercial cloud providers. Organizations may also instantiate a local instance of the AMIC system on their local network to reduce latency or satisfy data locality requirements. AMIC components support full or partial replication of stored data to partner sites.

An authentication and authorization service manages permissions to AMIC resources. Once users successfully provide valid credentials, the system provides security tokens that facilitate access with minimal security friction.

6. INITIAL PROTOTYPE

The AMIC program has developed an initial prototype of the data architecture to inform requirements and assess scalability. The current prototype is capable of ingesting autonomous vehicle sensor data stored in ROS bag formats. The AMIC prototype system was seeded with 2 TB of lidar, inertial, and GPS position sensor data collected during the CoVeR EET in October of 2019.

6.1. Dataset Catalog

The prototype Data Catalog includes an ingestion pipeline that scans each ROS bag file and generates a representative metadata record. Metadata records are added to the Data Catalog via standard REST API services. The database is periodically indexed by search services, to provide an optimized search interface for users. Users are able to execute search queries on the metadata through a web-based interface to identify specific data to return. The prototype also supports a connection to a web-based visualization engine to display streaming lidar, camera, and other sensor data. A screenshot of the prototype Data Catalog is shown in Figure 3.

6.2. Data Query and Processing

The initial prototype's query and processing performance was tested by executing two tasks. Although simple, the tasks are representative of possible analyses performed by a test director or algorithm developer. In task A, the user seeks to retrieve the 90% percentile value for vehicle roll rate for each platform in the dataset. In B, the user is tasked with returning all laser scans collected within 5 meters of a designated geographic coordinate. The duration needed to execute the query using AMIC verses the time needed to execute the task by linearly searching through 431 GB of bag files is shown in Table. 1. All tests were performed on virtual machine equipped with 4 processing cores and 32 GB of RAM with data stored on mechanical hard drives.

Table 1 Bag vs AMIC Query Performance

Task	Bag Search	AMIC
A.	6.5 hr	2.7 s
B.	6.5 hr	15.3 s

7. CONCLUSION

Collecting U.S. Army AGV data, during dedicated campaigns or through program demonstrations, is expensive and time consuming. The AMIC open data architecture will provide a mechanism to maximize the value of AGV data and create opportunities for enhanced data-driven innovation.

8. CONTRACT ACKNOWLEDGEMENT

This material is based upon work supported by the Department of the Army under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Army.

1. REFERENCES

[1]O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015.

[2]A. Geiger, P. Lenz, C. Stiller, R. Urtasun, “Vision Meets Robotics: The KITTI Dataset,” International Journal of Robotics Research, vol. 32, no. 11, pp. 1231-1237, 2013.

[3]P. Nelson, C. Linegar, P. Newman, “Building, Curating, and Querying Large-scale Data Repositories for Field Robotics Applications,” Field and Service Robotics, 2016.

[4]N. Lawrence, “Data Readiness Levels,” arXiv preprint arXiv:1705.02245, 2017.