特　集　「AI 計算」

A Data Management Infrastructure for Intelligent Systems

Antonello Ceravola (Antonello)　Honda Research Institute Europe GmbH, Europe
Antonello.Ceravola@honda-ri.de, http://www.honda-ri.de/

Frank Joublin (Frank) (same above)
Frank.Joublin@honda-ri.de

Heiko Wersing (Heiko) (same above)
Heiko.Wersing@honda-ri.de

Stephan Hasler (Stephan) (same above)
Stephan.Hasler@honda-ri.de

Behzad Dariush (Behzad) Honda Research Institute, USA
dariush@hra.com, http://usa.honda-ri.com/

Yi-Ting Chen (Yi-Ting) (same above)
ychen@hra.com

Keywords: big-data, data recordings, multi-sensor systems, data management, intelligent systems, AI, automotive, robotics

1. Abstract

In this paper we describe the design principles, implementation choices and general challenges we encountered in the creation of a data management infrastructure for recording data streams from test vehicles, robots and other platforms. The trigger for this data management infrastructure project was twofold: First from the proper setup of new test cars equipped with many sensors, delivering high bandwidth data recordings and second from achieving organized storage of such recordings for the development and testing of intelligent systems operating on the data. After the clearly stated demand of such a data management system from different divisions of our company, we, step by step, conceived it as a very general data management platform targeting different projects with different recording formats and platforms. Recording data from different projects have systematic commonalities, for instance most use time series of data, often from similar type sensors with similar information. However considerable differences exist with respect to data organization in recording sessions, stream formatting or coverage of specific situation/event or environment conditions. Our data management infrastructure targets to support different needs in the data management work-flow. Facilitating recordings visualization, search, inspection, annotation of events/entities present in the data and offline access is among our main targets.

Our approach is first to centralize storage of recordings, avoiding proliferation of copies of them in our network. We give GUI and programmatic access allowing both tool-based-manual-annotation processes as well as automated processes using AI/deep-learning methods. Subsequently we support extraction of events or meta-information from recordings, storing them in a database.. Our infrastructure enables then an efficient search over extracted information for exporting relevant recording segments, used for the creation of automotive or robotics intelligent systems.

2. Introduction

In the last decades the automotive domain progressively focused its attention towards electronics and computer based functionality. The trend here started with usage of custom electronic control units (ECU) supplying new functionality in vehicles and, slowly, moved towards adoption of more general purpose computers-based systems. In this move functionality implemented in cars broaden their spectrum, allowing inclusion of intelligent and robotics systems. At the time of custom electronic control units the functionality implemented were simple and making use of few and limited sensors (with data ranging around few kilobytes per second). The introduction of intelligent and robotic systems required complex functionality, treating many more sensors with higher data transfer (typically several megabytes per second or more).. Moreover, intelligent and robotic systems are recently adopting AI techniques such as deep neural networks that also require a large amount data for training. Such paradigm shift in research, development, and testing of intelligent systems requires handling of sensory data streams in the big-data domain. For this reasons the data acquisition and sensor recordings phase cannot be simply managed together with the development of vehicle functionality. This two packages reached a level of complexity that naturally lead to having *a data management package*, as an own project/concept and *a vehicle functionality package* as separate project/concept. In this perspective we decided to start a project with the target of designing and creating a data management infrastructure that supports the handling of big-data sensory data recordings, used in other projects targeting the creation of intelligent and robotics systems. In this paper we will review the process and address the challenges we encountered in reaching a first release of this data management infrastructure.

3. Working with Data

An important challenge in the development of intelligent systems which makes use of several large sensor data streams involves the organization of the data streams into an accessible format in terms of data size, data structure and packaging. Managing several sensors in real-time requires computer configurations with enough bandwidth, a suitable data grabbing software system and performant computer hardware and storage. We are considering here a system using multiple sources of data, each at different frequency, with different data structures and different sizes (in term of transmitted bytes of information). In the development of intelligent systems, such data can be sourced in different ways:

- As on-line data, feeding data, from sensors, in real-time to a running development/test system;
- As off-line data, by recording sensors data in real-time into a storage and, in a second phase, using it in the development/test systems with a play back functionality;
- Synthesized sensory data, through generation by virtual environment, feeding it to development/test system.

In some cases on-line data is mandatory when real-time and interaction is important for the development/test of a system. However several operations becomes difficult to realize: reproducing particular sequence of sensory data; starting and stopping the system in different cycles, performing test and then stopping and fixing a malfunction in place or running in step by step mode. A solution to most of the previous limitation is the usage of off-line data, where data is recorded in a first phase, and then played back to the system. The issues here are: recordings may require large storage; there is no possibility of real-time interaction; changes in sensory data format may invalidate existing recordings. An optimal solution could be to use a simulation environment from which sensors can be simulated creating synthetic data streams. If simulators can provide the required sensor quality, here the advantages of on-line and off-line can be covered. However, issues here may reside in achieving a sufficient simulation quality and in the complexity of the simulated scenario with its static, dynamic elements and interaction between them.

In this paper we analyze the choice of using off-line data, describing a data management system we have created for handling large recordings.

As a running example, along this paper, we will consider a car-related project where we implement intelligent systems such as advanced driver assistant systems (ADAS) or autonomous driving (AD). The car is equipped with the following sensors:

- 10 cameras (monochrome and color cameras);
- 6 radar sensors;
- 6 laser sensors;
- a GPS sensor;
- IMU unit;
- Several CAN network streams from the car.

Thirty independent data streams are recorded at a rate of approximately 200Mb per second. Challenges here are:

- **Computing Infrastructure**: Setting up a computing infrastructure able to receive all streams in real-time (via USB, Ethernet, CAN, ...);
- **Sensor Synchronization**: Implementing synchronization protocol for labeling each packet/frame of information sent from sensors to recording system;
- **Recording System**: Creating recording software system able to receive and store several streams in parallel;

- **Data Transfer**: Organizing the process for transferring data from recording platform to office network;
- **Play-back System**: Creating a system able to replay recorded data as at recording time.

4. State of Art

In the analysis of existing data management systems supporting management of sensory recordings one of the most used platforms in the automotive domain we found belongs to the ASAM (Association for Standardization of Automation and Measuring Systems) standardization consortium [ASAM]. This consortium, composed of about 30 different companies (OEMs, Suppliers, Software Vendors) mostly from Europe and USA, targets the definition of standards for measurement data in automotive. The main standard is ODS (Open Data Service) modeling a unified representation of measurement data (automotive data streams) for standardized server storages (see Figure 1). One of the aims is the possibility of exchanging data within or between different companies [ODS].

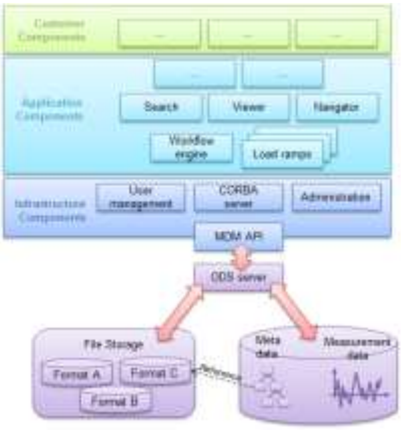

Figure 1: ASAM ODS architecture

In order to promote the adoption of their proposed standard, the consortium published part of the specification and several tools as open source. At the time of our investigation ODS was available with a server solution, a configurable graphical user interface and several modules for handling recordings [OPENMDM]. However it was implemented for low bandwidth data series which did not match our big-data requirement. In the later years, since big-data usage proliferated among automotive companies, such requirements have been taken in account by the consortium. Another data management system, which has been created in relation to a study [CAMP2012] of driving safety, is InSight SHRP2 NDS (Strategic Highway Research Program on Natural Driving Study) carried on by several USA universities and transportation institutions [SHRP]. The purpose of the study is to analyze how driver behavior is affected by driving context (driver, vehicle, roadway and environmental factors). For that objective drivers were invited to install a set of sensors (camera, GPS) with a recording system on their car. Data was collected in a central database and made available through a web interface [INSIGHT]. In this way universities could get access to data, visualize, annotate and search for particular segments on which performing data analysis. A very similar initiative which led to another data management system is carried on by CEESAR, a non-profit organization focused on road safety. They have created the system named SALSA related to the UDRIVE [ROB2014][UDRIVE] consortium (European Naturalistic Driving Study) which cover similar workflows as InSight. These two systems are not targeting the distribution of the data management system itself; instead they offer access of recorded data to 3rd parties. These systems are customized around the data of the respective projects and are not meant to be hosted and populated with data by 3rd parties. Nevertheless they represent a good proof of concept on what we are targeting.

One more European initiative which is targeting the realization of a data management system (see Figure 2) with a major focus on the annotations is the Cloud LSVA [CLSVA]; a Cloud based Large Scale Video Annotation platform. This European project, created in the scope of the innovation program Horizon 2020, consider the full process from real-time vehicle sensor acquisition, transmission of data streams to a cloud server, annotation process, indexing and search on recordings. The target of this European project is the creation of a set of standards for representing sensory data and meta-information, together with the creation of a cloud service that can store, visualize and allow annotation of recordings coming from different companies. This project shares many of our requirements except for the choice of a direct car-to-cloud data transmission, where, instead, we target for copy of full recording sessions to central company file servers.

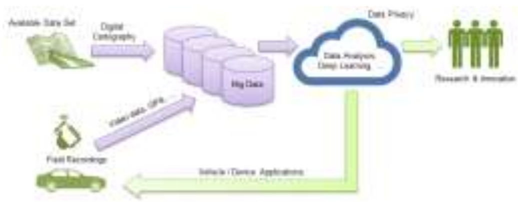

Figure 2: Cloud LSVA targets overview

A commercial solution which is actually designed to handle recordings in a more general way is the DSSC (Distributed Storage and Simulation Cluster)[XCUBEPD], from XCube

[XCUBE]. DSSC is a distributed storage (even geographically) and computation system based on the principle of keeping data to closest accessible nodes from where it has been produced. For computations over data, programs are distributed along nodes (hosting a set of virtual machines) containing targeted data. This principle of code mobility [FUGG1998] through the usage of virtual machine is particularly suited for data management systems handling big-data (as in our case). Although this approach is suitable when using standard computer configurations, it presents a challenge for our applications since our programs may depend on different specific libraries or hardware configurations. Moreover the architecture of the DSSC uses a strong encapsulation of data and programs which make development and test of user applications cumbersome (see **Figure 3**).

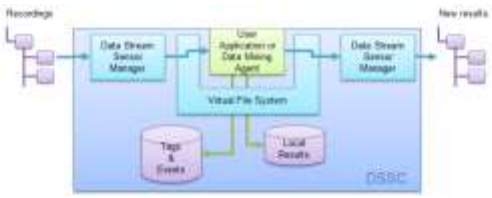

Figure 3: DSSC Node architecture

Generally, one of the most important functionalities of such data management systems is related to the possibility of searching in recordings relevant events or objects, and then using search results for extracting recording segments as verification or test data for developing intelligent systems. Such search functions are based on a process (executed when a recording is imported in the system) able to collect searchable information from recordings, particularly, annotated/labeled data streams.

Recent breakthroughs in machine learning and computer vision have tremendously accelerated progress in image recognition, localization, motion planning, and decision making. In particular, deep neural networks have been shown to learn representations which are generically useful across a variety of visual domains where large-scale datasets with semantically relevant labels are necessary. For automated driving in particular, carefully crafted datasets with ground truth labels at very large scale are important in learning a mapping from an image to meaningful affordance indicators such as relative position, orientation, and velocity of the ego-vehicle with surrounding traffic participants.

Availability of a large-scale semantically labeled data-sets has been limited in part because the process is time-consuming and expensive. Although these factors have slowed the development of new large-scale collections, the research community has nevertheless continued their investment toward creation of various datasets such as ImageNet [DENG2009], the NYU-DepthV2 [SILBER2012], the PASCAL-Context Dataset [MOTTA2014], and Microsoft COCO [LIN2014]. These datasets have certainly accelerated progress in semantic segmentation of indoor scenes as well as recognition of common objects. However, they are not suitable for more specific tasks such as those involved in automated driving.

In the automotive domain, popular semantic segmentation benchmark datasets include CAMVID [BROS2009] and more recently, Cityscapes [CORD2016], a dataset introduced as training samples for pixel-level and instance-level semantic labeling. Cityscapes is comprised of 5000 images containing pixel-based labels and 20,000 images containing coarse labels collected from different cities in Europe. Although this dataset exceeds previous attempts in terms of dataset size, annotation richness, scene variability, and complexity, the algorithms trained on this dataset do not generalize well to data tested on different traffic scene domains including variations in seasons, lighting, weather, traffic conditions, scene structure, sensor characteristics, etc.

The cost of scaling this type of labeling adequately would require a prohibitive economic investment in order to capture sufficient number of images with the required variability across a variety of tasks and visual domains. To address this issue, promising alternatives have been proposed that utilize synthetic imagery that simulate real urban scenes in a vast variety of conditions and produce the appropriate annotations [ROS2016]. In particular, the SYNTHIA dataset was generated with the purpose of aiding semantic segmentation and related scene understanding problems in the context of driving scenarios. SYNTHIA consists of a collection of photo-realistic frames rendered from a virtual city and comes with precise pixel-level semantic annotations for 13 classes. It has been shown that inclusion of SYNTHIA in combination with publicly available real-world urban dataset during the training stage significantly improves performance on the semantic segmentation tasks.

In lieu of manual annotation, researchers are exploring methods to automatically generate large-scale semantically labeled data-sets. Their approach can create 2D labels for large video datasets from street scenes, resulting in automatic or semi-automatic processes for generating semantic and instance image annotations.

5. Motivation

Introduction of a data management system started in relation with the development of experimental vehicles equipped with several sensors to research intelligent systems in

ADAS/AD domain. The experimental vehicles are equipped with several high definition cameras and LiDAR sensors, producing much higher volume of data than our previous vehicles.., In addition, our past experience in managing recordings by copying data in multiple locations on our network was no longer feasible. A managed and streamlined handling, storage, and sharing of data storage across our global newtwork was necessary. In the initial design of our data management system, the majorrequirements that we identified included:

- A system capable of handling big-data recordings (several hundreds of Gb per recording).
- Ability to support single and multiple regions. .
- System and data accessible through a graphical user interface, command line, and programmatically.
- Ability to treat video, , point cloud, and time series data.
- Support and management of different recording formats produced from different platforms such as vehicles, robots, etc. .
- Data streams must be time-synchronized to enable development of algorithms that require sensor fusion.
- Manual and automatic annotation of events and objects classes with a structured taxonomy must be supported. .
- All annotations must be indexed and searchable at each site or collectively to allow sharing between sites.
- Recordings can be exported in full or in segments.

In our previous experience on managing recordings used for our intelligent systems we left much freedom to each project in deciding where recordings should be stored, how they should be organized, and how to visualize and access data. That organization worked well as long as recordings were small, interaction with other projects was minimal and not much reuse of recordings over time necessary. However as soon as the sizes of recordings grows or reuse within/between projects raises, such freedom becomes an obstacle. For instance, since we had no particular policy on where recordings should be stored, we found that usual practice was to keep several copies of recordings stored in different locations of our network. That was leading to a large number of copies of recording, which maybe have been used only one time and never again. Moreover, another emerging issue has been a proliferation of recordings with much different organization, both in structure and formats. Recordings structure and data formats could differ in terms of:

- **Recording organization structure**: how files and directories are used to organize/distribute recording information;
- **Modularity**: some recordings are based on a single main file (containing all sensor streams), others use single file per sensor stream or different files, each for a sensor frame/packet;
- **Time synchronization**: some recordings relay on implicit time synchronization, others encode synchronization information in file names, in index files or along with sensor data;
- **Stream format**: some streams are recorded in human readable format (e.g. ASCII, json) others use binary encoding requiring specific libraries for access.

With so much variability it is not possible to target for a single common recording format. Particularly because recording formats are, in some cases, linked to the actual streaming performance of recording platforms. For instance, in a vehicle platform setup (sensor-network-computer) for a video camera sensor it is not possible to store data by creating single image files (either compressed or raw) for each camera frame due to file system overhead in creating files at that frequency, leading to many frame-drops. While, instead, storing all frames in a raw format on a single file leads to no frame-drops. With such types of constraints, we had to consider creating a data management infrastructure that would allow original recordings to be stored in different formats.
In deciding on the architecture of the data management infrastructure we had to consider several requirements. For instance:

- Infrastructure should suit needs of different divisions in different locations;
- Each division should have a fast access to recordings;
- A division would work mostly on local recordings, however sharing recordings should be possible;
- Each division, and sometimes each project, may have a different recording format;
- A mostly consistent data management workflow across all divisions/projects.

Another consequence of providing freedom for data management projects resulted in different teams to create specialized data management tools for each specific activity. t. Such tools were highly customized for a specific project instance and generally provided functionality for browsing and annotating recordings together with the possibility to play-back recordings. . That structure created the following challenges:

- **Managing recording copies**: difficult for users to manage recordings distributed along network storage (keeping track of obsolete recordings, differences, relevance, ...);

- **Usage of recordings**: Usually only a portion of each recording actually used for the development/test of the target system;
- **Data access**: It was often necessary to have a fast access of recordings (in terms of play-back).

6. Approach

The main constituents of the data management system include:

Recording structure

The organization of the recording structure plays a crucial role in the resulting performance we can achieve for accessing, operating and sharing the recordings. . Project members often keep sets of recordings in their local computers, feeding them to their target systems during implementation and testing. Our solution is to organize the recording structure into directories according to the types of information as follows:

- **.../Recording Name/**
  - **On-line streams/**: here we store all streams as recorded from the recording platform. The term "on-line" means that streams are the only one available to an application running in the car for a live execution. Once created, this directory is locked against writing;
  - **Off-line streams/**: here all streams generated a-posteriori from the on-line streams or even from other off-line streams. Those streams can be generated by programs (post-processing, automatic annotation, recognition algorithm) or generated by humans (manual annotations);
  - **Information streams/**: all streams that helps to abstract from original recording data, it contains meta-data files describing each streams available in the on-line and off-line directories as well as calibration, or platform information;
  - **Preview streams/**: sub-sampled representation of all streams in a simple visualizable format. This function provides quick recordings preview by the graphical interface to quickly visualize recording contents;
  - **Searchable information/**: here we store streams of tags. Tags are descriptive entities with attached properties which can be stored in database and searched via queries. Examples of a tag in the context of a vehicle platform can be driving speed, GPS location at a given time or the presence of a pedestrian in the image for a time-period.

With this structure, each recording is fully contained in a directory, organized with the listed sections/sub-directories. The choice of this organization has the purpose of separating original information coming from recording phase (On-line streams), from data subsequently added (describing/augmenting recoding data). This recording structure is applied around the original recording data at import time and stored in a central recording storage.

Central storage architecture Our storage solution considers a geographically distributed central storage for each division.. This gives possibility of having fast access for local users (local division), keeping the possibility to share recordings by sending copies of recordings to other divisions when necessary. This solution gives the possibility to use different storage solutions between divisions, depending on specific needs in each location. Some divisions need a basic NFS access to the recording and data integrity management. In such case the central recording storage uses CEPH [CEPH] architecture with replication of data and CephFS access. In other divisions, several computational process on large recordings are required. In that case, the central storage can be based on the Hadoop [HADOOP] architecture where a cluster of computers handles parallel access on the same recordings through Map-Reduce processes.

Recording's import process

In order to create a systematic storage process for new recordings, we defined a recording import procedure. Each new recording is encapsulated in the previously described recording structure from which the computation of the different off-line streams based can be done. This import process source data from the original recording, therefore, depending on the recording format and platform, a different data access functions is used. For that reason, the import process is based on a set of plugins, each able to treat different recording or stream formats.

Visualization of recordings

Our user studies showed that users often need visual inspection of recordings to evaluate the quality of recordings or finding particular recorded events/conditions. Several information may need to be visualized together with a recording, such as: : date and description of the recording, location and play-back of video or other streams. For such functionality we have designed a set of meta-data files where storing recordings meta-information. Together with this, at import time, we create a set of preview files for most of recording's streams. Such information can be browsed in a direct way form file system or via a web-based graphical user interface, giving easy access to all available recording present in the storage.

Annotation and labeling tools

In the development process of intelligent systems, data annotation is a crucial step. Here ground truth information or, higher level representation of recorded data can be created (off-line streams). For this reason, we designed a set of manual and automatic annotation tools together with allowing the possibility to use 3rd party tools in that phase. Our web graphical interface includes manual annotation tools, allowing users direct creation of annotation in recordings. However, manual annotation is an expensive process, error prone and requiring validation and quality management. Therefore, we are also developing automatic annotation sub-systems of different types. In the context of traffic scenes , it is very import to have annotations algorithms that dynamically capture the 3D representation of the world providing the possibility to generate accurate labels for higher order semantic descriptors such as 3D position, orientation, and velocity of traffic participants. These higher order descriptors are not easily attainable from manual annotation of images. We work here on a novel machine learning framework based on LIDAR-camera fusion to automatically assign semantically meaningful tags to an image to accelerate construction of benchmark datasets relevant for ADAS/AD. In our approach we take as input synchronized image and point cloud data streams, and produces semantic annotations in image and point clouds, including a semantically segmented image, as well as 3D dynamic reconstruction of street scenes, traffic participants, and their velocity profiles (see Figure 4).

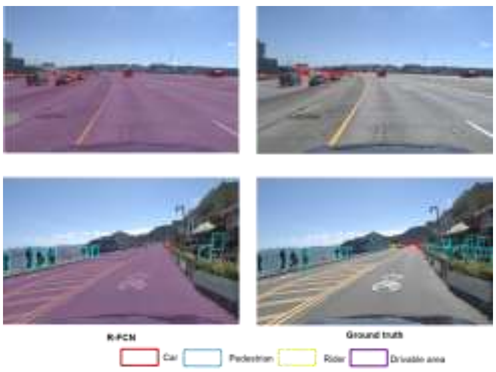

Figure 4: Example of scene annotation

Tag extraction & Indexing process

Given that recordings may contains hours of recorded data and a huge number of events or annotations, a direct search of information by direct access of file-streams is computationally too expensive. We batch-post-process all searchable information (like meta-data, annotations, labels) transforming them in tag streams. Such streams represent information in textual format with time stamps, suitable to be stored in a database. Tags do not have a predefined structure, therefore we store them in NoSQL database, flexibly modeling the different relations tags have, together with gaining a fast access to them. In the database we additionally create structures that group tags according to time-stamp (in some case with also duration). This structure allows a more performant search when looking for recordings-segments where particular events or annotation are valid.

Recording search

By storing tags in a database we are able to have a preformat search, finding segments of recordings on which a particular condition is met. On top of the database, we have created a simplified query syntax, which, at execution is then translated into the specific query language. Together with this we have in the database specific structures storing statistics on the used tag-values and types. This allows users to know in advance what can actually be searched in recordings.

In the analysis of search requirement, we identified three major types: search for existence (e.g. finding recordings where a car pass by a crossing or drives in rainy conditions), search for values (e.g. the speed of the car is higher than a value or the car is in a particular GPS area) and search for complex conditions (e.g. the car is overtaken by a truck on the left lane or the car is slowing down due to a faster car passing by). Where the first two types of search can be built with our query language, the third type of search requires access to recording streams and additional interpretation of events and recording context. We currently have the first two types of search; the third has not been yet addressed in our work.

Search results

Search results coming from the execution of a search query produces a list of recordings segments (with start and end time) matching the query. Search results can be used in different ways. They can be used to visualize the segments found for a manual inspection; can be saved into a file and used programmatically in a target system for accessing the recording segments directly (via central recording storage); or they can be used with exporting functions to create copies of found segments. This last case allow users to get copies of recording results in local machines, playing them back in the target system with high performance.

7. The EMI System

Design and development of our data management system (Experiment Management Infrastructure, EMI), has been done in a cooperation project between divisions in Europe and United States.. Both divisions shared requirement for handling vehicle recordings with large number of sensors and store them in a centralized way.

In Figure 5 the list of steps and tools supported in our recording management workflow.

Figure 5: EMI workflow steps

Workflow starts with a recording session (raw data acquired from a recording platform i.e. a car, robot ...) which gets copied from the platform to a local storage in the company network. The first activity is a preprocessing of the recording data, cleaning it and cutting unnecessary parts. General information describing the purpose of the recording is added as metadata in this phase. The next steps include importing, reshaping recording structure into uniform organization, adding streams computed on the basis of original streams and creating previews. At this stage manual or automatic annotations can be generated. Subsequently, , tags are extracted from all streams and imported into the searchable database.

At this stage, users can directly access recordings in full length or use search query to find segments matching a given condition. Search results (interval list file), can be used to access recording segments directly or export them as sub-recordings, copying them to user's local computers. In Figure 6 some example of search queries.

Search query	Description
EmiDbSearch.py -r "project=TFF OR project=PAZ"	Get recordings from project 'TFF' or project 'PAZ'
EmiDbSearch.py -r "country=eu" -s "ego_vehicle.mean_speed > 15"	Get intervals from country 'eu' where ego vehicle is faster than 15m/s
EmiDbSearch.py -s "traffic_objects_extended.objects.most_typename=car AND traffic_objects_extended.objects.mean_velocity_abs_x>20"	Get intervals with a traffic object that is a 'car' and a traffic object with a speed over 20m/s
EmiDbSearch.py -s "[traffic_objects_extended.objects|most_typename=car AND mean_velocity_abs_x>20]"	Get intervals with a traffic object that is a 'car' with a speed over 20m/s

Figure 6: Example of EMI search query

From the architecture point of view, we target a data management system with a fast data access for each division. That requirement leads to a design architecture that supports a central data management system per division with a distributed architecture (each division having a server instance). This architecture allows high performance access to local recordings (see EMI Back-End in Figure 7) with a central access view of recordings through the web interface (see EMI-UI in Figure 7). Here, each server, can have a set of local choices in order to handle specific recording formats, different access to recordings (CEEPH, Hadoop ...), faster search with local indexes or specific project annotation services.

Figure 7: Overall EMI architecture

In order to investigate performances of our system, we have executed several tests with recordings of different duration. On average 1 hour of recording correspond to a size of 680 Gbytes for 20 sensor streams. Figure 8 illustrates some example of execution timing.

Functionality	Execution Time
Import of recording (1h recording)	30 minutes
Computation of Tags (1h recording)	1 hour
Database Update (1h recording)	2 minutes
Execution of search query (6h recording on DB)	2 seconds
Execution of search query (133h recording on DB)	7-12 seconds
Export of recording (1h recording)	30 minutes

Figure 8: Example of execution time for EMI functionality

8. Conclusions

In this paper we introduced the needs and challenges AI scientists are confronted with when dealing with systems and multi-sensor platforms producing with large training data for learning algorithms. We described a solution we designed and created to store, share and use such type of big-data sets in a multi-site organization structure where no a-priory constraints can be set on recording formats. Presence of several international projects targeting to solve similar problems shows that such experience can be valuable for AI communities. The next steps are focused on improvement and finalization of parts of the web-based user interface and several low level functionalities operating on streams for annotation and tag generations. We are in a first deployment phase, integrating user’s feedback in further development planning.

References ◇

[ASAM] ASAM WIKI, access 2017, https://wiki.asam.net

[BROS2009] G. J Brostow, J. Fauqueur, R. Cipolla, "Pattern Recognition Letters, Semantic object classes in video: A high-definition ground truth database", Vol 30, No. 2, pp. 88-97, 2009

[CAMP2012] K. L. Campbell, "The SHRP 2 Naturalistic Driving Study", TR News 282, September-October 2012, https://insight.shrp2nds.us/documents/shrp2_background.pdf

[CEPH] CEPH, access 2017, http://ceph.com

[CLSVA] Cloud LSVA - Large Scale Video Annotation, access 2017, http://cloud-lsva.eu

[CORD2016] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding", In proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), 2016

[DENG2009] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, "A large-scale hierarchical image database", In proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), 2009

[FUGG1998] A. Fuggetta, G. P. Picco, and G. Vigna, "Understanding code mobility", IEEE Transactions on software engineering 24.5: 342-361, 1998

[HADOOP] Apache Hadoop, access 2017, http://hadoop.apache.org

[INSIGHT] InSight Data Access Website - SHRP2 Naturalistic Driving Study, access 2017, https://insight.shrp2nds.us

[LIN2014] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C. L. Zitnick, "Microsoft COCO: Common objects in context", In proceedings of European Conference on Computer Vision (ECCV), 2014

[MOTTA2014] R. Mottaghi, X. Chen, X. Liu, N.G. Cho, S.W. Lee, S. Fidler, R. Urtasun, A. Yuille, "The Role of Context for Object Detection and Semantic Segmentation in the Wild", In proceedings of Computer Vision and Pattern Recognition (CVPR), 2014

[ODS] ASAM ODS Base Model, access 2017 http://www.highqsoft.de/download/ao_base.htm

[OPENMDM] OpenMDM toolkit, 2017, https://www.openmdm.org/about-openmdm/asam-ods

[ROB2014] E. Rob, et al. "UDRIVE: the European naturalistic driving study", In proceedings of Transport Research Arena. IFSTTAR, 2014

[ROS2016] G. Ros, L. Sellart, D. Vazquez, A. M. Lopez, "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes", In proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), 2016

[SHRP] Strategic Highway Research Program (SHRP) 2 - Revised Safety Research Plan, access 2017, http://onlinepubs.trb.org/onlinepubs/shrp2/RevisedSafetyResearchPlanMarch2012.pdf

[SILBER2012] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, "Indoor segmentation and support inference from rgbd images", In proceedings of European Conference on Computer Vision (ECCV), 2012

[UDRIVE] UDRIVE - European Naturalistic Driving Study, access 2017, http://www.udrive.eu

[XCUBE] XCube, access 2017, http://www.x3-c.com

[XCUBEPD] XCube Product Description, 2017, access 2017, http://www.x3-c.com/wp-content/uploads/2017/02/XCube-Products-Description_Jan2017.pdf

Authors' Profile

Antonello Ceravola (Non Member)
He studied Computer Science at the University of Pisa, Italy. He worked in the field of IT software for five years dealing with multimedia systems, large scale software infrastructure for telecommunication systems, multi-tier applications and workflow engine for process management systems. From 2001 he joined Honda Research Institute Europe, Germany, currently Principal Scientist. His research interest includes software components, middleware, large-scale systems and integration environments.

Frank Joublin (Non Member)
He received a European Ph.D. degree in neurosciences from the University of Rouen (France), in 1993. From 1994 to 1998 he was postdoctoral research fellow at the Institute für Neuroinformatik, university of Bochum, Germany. From 1998 to 2001 he was customer project manager at Philips Speech Processing Aachen. Since 2001, he is principal scientist at the Honda Research Institute Europe, Germany. His research interests include developmental robotics, semantic acquisition and data mining.

Heiko Wersing (Non Member)
Heiko Wersing received the Diploma in Physics in 1996 and a Ph.D. in science in 2000 from Bielefeld University, Germany. In 2000 he joined the Honda R&D Europe GmbH in Offenbach, and is now working as a chief scientist at the Honda Research Institute Europe GmbH, Offenbach Germany. In 2017 he was awarded an honorary professorship at Bielefeld University, Germany. His research interests are incremental & online learning, personalization, adaptive HMI, and computer vision.

Stephan Hasler (Non Member)
Stephan Hasler received a PhD degree in Engineering from the University of Bielefeld (Germany) in 2010. He is currently Senior Scientist at the Honda Research Institute Europe, Germany. His research interests include models for visual perception, lifelong learning, and human-machine interaction.

Yi-Ting Chen (Non Member)
Yi-Ting Chen received a PhD degree in Electrical and Computer Engineering from Purdue University with emphasis in imaging system and its image quality analysis. After graduation, He moved to University of California at Merced and visited Google to work on computer vision researches and projects. He is currently research scientist at Honda Research Institute USA, where he is working on computer vision and machine learning. Specifically, he is working on 3D dynamic scene understanding for different applications in automated driving.

Behzad Dariush (Non Member)
Behzad Dariush received a PhD degree in Electrical Engineering from The Ohio State University with emphasis in robotics and control systems. He is currently Chief Scientist at Honda Research Institute USA, where he is managing the Knowledge Discovery group to support automated driving and robotics projects in global Honda. His past research activity includes humanoid robotics, wearable-technologies, and human motion analysis.