Autonomous control system prototype for GAZ A65R32

Tiurikov M. I., Malinin N. V., Evdokimova T. S., Sinodkin A. A. and Fedosova L. O.

1 Nizhny Novgorod State Technical University n.a. R.E. Alekseev, Nizhny Novgorod, Russian Federation
E-mail: stdbreaks@gmail.com

Abstract
The article presents current state of development of autonomous control system prototype for GAZ A65R32. The solutions for mapping and localization, planning, used sensors, as well as the applied methods of computer vision and deep learning and the methods used to transfer data in the system are described.

Keywords: driverless vehicles, ADAS, computer vision, machine learning, mapping, localization, deep learning

1. Introduction
ADAS systems in an actively developing area that gathers the attention of many researchers and corporations. The main problem of current automatic driving systems is the high cost of equipment, mainly due to the use of lidars to obtain information about the environment. In addition, the development of such systems is carried out sequentially according to the levels of driver assistance systems ADAS, so at the moment such control systems represent the further development of adaptive cruise control systems. The main goal of development described in the article is to reduce cost of such systems by replacing lidars by much cheaper video cameras and stereo cameras and to develop level 5 ADAS system from scratch bypassing intermediate levels.

2. Hardware equipment
We equip out prototype with two stereocameras and lidar to provide mapping module with information about objects and landscape around the vehicle. At this stage of development our sensor configuration does not provide full 360 degree field of view for obstacle perception, as of lack of bandwidth provided by on-board computer of the vehicle.

As the workaround for this problem we gain 180 degree field of view provided by Hokuyo UTM-30LX-EW 2D lidar with working range of 30 meters on the front paired with ZED Camera to get additional 3D information about obstacles and ground surface at distances up to 20 meters and 90 degrees field of view without giving up too much precision.

Additionally we have placed Intel RealSense D415 camera at the back of the vehicle to get additional 63 degree rear field of view but with small range of up to 10 meters as on larger distances result point cloud becomes too noisy to provide significant information for mapping module.

To gather data about vehicle’s movement and position the UBlox C94-M8P GPS/GLONASS module is utilized along with Atmel XPlained Pro BNO055 development board featuring Bosch BNO055 inertial measurement unit.

As on-board computer for our vehicle we use a regular PC with following configuration:

- CPU: AMD Ryzen 5 2400G
- RAM: 16 GB DDR4
- GPU: 2 x Nvidia GeForce GTX 1060 6 GB
- SSD: 250 GB SATA

3. Control system modules implementation

3.1 Mapping module
As mentioned earlier mapping module uses two stereo cameras and lidar to gather information about vehicle’s surroundings.

Mapping module consists of three levels of maps as shown in Figure 1:
1. 3D mapping algorithm RTABMap[1] is utilized to create precise 3D map of mostly static objects. As we observed during first tests of mapping system this algorithm is not suitable for highly dynamic environments as it often struggles to fastly clear freed areas of map.

2. 2D global grid map was added to avoid getting leftover parts from moving obstacles as described above by fusing lidar and using higher refresh rates than RTABMap. It is used by global planner to provide vehicle with route from current position to goal state.

3. 2D local grid map is built by getting small area of global grid map. It also has higher resolution and refresh rate and relies heavily on lidar data as it provides much higher refresh rates than can’t be achieved by stereocameras. Is is utilized by local planner that is used to avoid obstacles and provide low level control system with steering and speed commands.

![Figure 1. Structure of mapping module.](image)

3.2 Localization module
Localization module heavily relies on visual odometry received from ZED Camera driver. Frame processing is done on the graphics cards by proprietary algorithms provided by ZED with the driver. The optimal resolution of 3840 by 1080 pixels at 30 frames per second is selected based research presented in [2].

Additionally to increase robustness of localization we utilize Atmel XPlained BNO055 module to achieve high refresh rates of vehicle’s position, direction, velocity and acceleration. To get the global coordinates the UBlox C94-M8P GPS/GLONASS module is installed so we can set target points in global coordinates and track vehicle’s position. The data fusion is made by applying extended Kalman filter.

As a future work we are planning to use Maplab[3] to get visual odometry from cameras as currently used proprietary algorithms from ZED Camera driver requires CUDA capatible GPU to run.

3.3 Planning module
The planning module utilizes the scheme that is common for mobile robotics which consists of global planner to get complete route from current position to goal so local planner do not get stuck in local minima and the local planner to perform obstacle avoidance. The global planner is base on common A*[4] path planning algorithm and the local planner uses Dynamic Window Approach (DWA) [5]. Those algorithms are both simple and use models that do not perfectly describes kinematics of a car, but their simplisity allows us to achieve low latency reaction of the vehicle to dynamic obstacles.
3.3 Machine vision module
As of now machine vision module includes parking lot recognition, signs and traffic lights detection and lane segmentation solutions. Sigh and traffic light detection along with lane segmentation are performed by convolutional neural networks. To train those networks we mostly utilize BDD100K[6] dataset for detection and segmentation tasks paired with Russian Traffic Sign Dataset (RTSD) [7] for traffic sign classification.

3.3.1 Traffic sign and traffic light detection.
Object detection is a computer vision and image processing task related to semantic localization of objects of a certain classes. This submodule implements detection of objects of two classes: traffic sign and traffic light. As a metric of accuracy of different detection neural network architectures Average Precision or mAP is utilized:

\[
AP = \sum (r_{n+1} - r_n) \cdot \text{interp}(r_{n+1})
\]

(1)

where p is precision:

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

(2)

and r is recall:

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

(3)

here TP is true positive detections count, FP is false positive detections count, FN is missed of false negative detections count.

During architecture search for detection module we considered only single shot detectors such as SSD[8], YOLO[9] and RetinaNet[10]. Most development time was spend on this stage. RetinaNet was selected as a base for out detection module giving 58% mAP score compared to significantly smaller 40% mAP for SSD and 45% mAP for YOLO. The training time for RetinaNet in our case was around 30 hours on a single GTX 1080ti.

The results are present on Figure 2 below.

![Figure 2. Combined results of sign detection and recognition module and lane segmentation module.](image)

3.3.2 Lane segmentation.
Semantic segmentation is a process of division of every pixel in an image into a specific classes of objects. Semantic segmentation is used to highlight different objects in an image. The deep learning is utilized for this task as of classical computer vision algorithms such as WaterShed or MeanShift is not capable of providing such level of accuracy as deep learning approaches.

To compute precision of semantic segmentation neural networks we used Intersection over Union metric or IoU:

\[
\text{IoU} = \frac{S_O}{S_U}
\]

(4)

where \(S_O\) is area of intersection between true area of object on image and neural network predicted area for this object and \(S_U\) is union of true area of object and neural network predicted area of image for specific object.

As in object detection task multiple architectures of neural networks for semantic segmentation were reviewed such as: PSPNet[11], ENet[12] and LinkNet[13]. LinkNet were selected as a base for this module giving us 81% IoU with about 9 times higher frame rates than PSPNet with 89% IoU. To achieve 81% IoU we trained LinkNet for around 25 hours on a single GTX 1080ti.

3.3.3 Parking lot detection.
To detect parking lots for automated parking as of now we utilize classical computer vision algorithms as there is no suitable datasets developed yet for this task to train neural networks.
The overall pipeline consists of performing Bird’s View transform on image from camera, color thresholding followed by detection of vertical and horizontal lines with angle tolerance of 10 degrees, then rectangle is fitted to cover parking lot area and parking lot keypoints are calculated relative to camera position.

The results are present on Figure 3 below.

![Figure 3. Parking lot detection results visualization.](image)

3.4 System components integration

Integration of described modules is done via Robot Operating System (ROS). ROS includes libraries to exchange messages between modules using TCP sockets, which is implemented as Publisher-Subscriber scheme with support for themes for message topics and some basic message types to ease further development of control systems for robots.

At present moment message exchange system implemented in ROS does not support data encryption and client authentication. So to develop a basis for future secure message exchange system of autonomous vehicle we implemented a NATS based message system to exchange messages between high level control system (local planner in particular), low level control system driver and smartphone used for teleoperation. NATS is a high performance message broker which supports authentication, message encryption and transfer over TLS protocol and is supported on wide variety of platforms. Simplified data flow scheme is present of Figure 4.

Integration of data from machine vision modules is now in active development. Current mapping module does not support any convenient methods to integrate data other than obstacle positions. Our solution of this problem is to reimplement mapping module using multilayered maps provided by Grid Map[14] library. This will allow us to use a dedicated layers to map results of different vision tasks with obstacle information and further consider it in cost functions used by global and local planners. Figure 5 illustrates integration of different sensors to global multilayered 2d maps.

![Figure 4. Data exchange between different modules of autonomous driving system](image)
4. Conclusion and future work

Proposed autonomous vehicle prototype was tested on “ROBOCROSS-2019” open test of unmanned vehicles and aircrafts where it took a third place. At this stage our prototype is able to autonomously avoid obstacles and most of future works is related to integration of vision modules data into mapping and planning modules.

As one of future research areas we are targeted at development of highly integrated hardware vision modules based on neural network processors. As part of this research we are planning development of efficient convolutional neural networks for image processing on embedded hardware.

Acknowledgement

This research done with the financial support from Ministry of Science and Higher Education of the Russian Federation (the unique project identifier is RFMEFI57717X0268).

References

[1] Labbé M., Michaud F. RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation 2019 Journal of Field Robotics. 36(2)

[2] Luis E.O., Elizabeth V.C., Luiz M.G. Depth Data Error Modeling of the ZED 3D Vision Sensor from Stereolabs 2018 Electronic Letters on Computer Vision and Image Analysis. 17(1)

[3] Schneider T., Dymczyk M.T., Fehr M., Egger K., Lynen S., Gilitschenski I., Siegwart R. maplab: An Open Framework for Research in Visual-inertial Mapping and Localization 2018 IEEE Robotics and Automation Letters

[4] Hart P.E., Nilsson N.J., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths 1968 IEEE Transactions on Systems Science and Cybernetics 4(2)

[5] Fox D., Burgard W., Thrun S. The dynamic window approach to collision avoidance 1997 IEEE Robotics & Automation Magazine 4(1)

[6] Yu, F., W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling 2018 arXiv e-prints. arXiv:1805.04687.

[7] Shakhuro V.I. Russian traffic sign images database 2016 Computer optics 40

[8] Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg SSD: Single Shot MultiBox Detector 2015 arXiv e-prints. arXiv:1512.02325

[9] Redmon, J., S. Divvala, R. Girshick, and A. Farhadi You Only Look Once: Unified, Real-Time Object Detection 2015 arXiv e-prints. arXiv:1506.02640.

[10] Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár Focal Loss for Dense Object Detection 2017 arXiv e-prints. arXiv:1708.02002

[11] Zhao, H., J. Shi, X. Qi, X. Wang, and J. Jia Pyramid Scene Parsing Network 2016 arXiv e-prints. arXiv:1612.01105.

[12] Paszke, A., A. Chaurasia, S. Kim, and E. Culurciello ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation 2016 arXiv e-prints. arXiv:1606.02147.

[13] Chaurasia, A. and E. Culurciello LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation arXiv e-prints. arXiv:1707.03718

[14] Fankhauser P., Hutter M. A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation 2016 Robot Operating System (ROS) – The Complete Reference