

3-бөлім

Раздел 3

Section 3




Информатика

Информатика

Computer
Science

IRSTI 28.23.15

DOI: <https://doi.org/10.26577/JMMCS.2022.v116.i4.06>

Zh. Yessenbayev^{1*} , Zh. Kozhirbayev² , A. Shintemirov³ 

¹Atyrau oil and gas university, Atyrau, Kazakhstan

²National Laboratory Astana, Astana, Kazakhstan

³Nazarbayev University, Astana, Kazakhstan

*e-mail: zh.yessenbayev@aogu.edu.kz

DEVELOPMENT OF A COMPUTER VISION MODULE FOR AUTONOMOUS VEHICLES

The favorable geopolitical position and very large transit potential of the Republic of Kazakhstan in the field of land freight traffic between China and Europe makes the transport logistics industry one of the most promising areas for the development of the country's economy. In this context, deployment of unmanned cargo vehicles to minimize the costs of fuel consumption and use of human labor in labor-intensive and routine operations of logistic processes both inside warehouses and during freight transportation on public roads seems natural and efficient as ever.

This paper describes the results of a research work on development of a computer vision module for an autonomous truck prototype. The performed project stages include installation of the necessary equipment, training of computer vision models and development of a mapping between cameras and LIDAR sensor for object classification and localization purposes.

Key words: Computer vision, autonomous vehicle, vehicle trajectory planning, real-time trajectory planning, unmanned solution.

Ж.А. Есенбаев^{1*}, Ж.М. Кожирбаев², А.М. Шинтемиров³

¹Сафи Өтебаев атындағы Атырау мұнай және газ университеті, Атырау қ., Қазақстан

²National Laboratory Astana, Астана қ., Қазақстан

³Назарбаев Университет, Астана қ., Қазақстан

*e-mail: zh.yessenbayev@aogu.edu.kz

Автономды көліктер үшін компьютерлік көру модулін әзірлеу

Қазақстанның оңтайлы геосаяси жағдайы мен Қытай мен Еуропа арасындағы жүк тасымалы саласында Қазақстан Республикасының үлкен транзиттік әлеуеті көліктік-логистикалық саланы ел экономикасын дамыту үшін перспективалы бағыттардың бірі болып табылады. Осы тұрғыда отын тұтыну шығындарын азайту және адам еңбегін пайдаланудың логистикалық үдерістердің ішінде, сондай-ақ қоғамдық көліктердегі жүктерді тасымалдау кезінде пайдаланбайтын жүктердің технологиясын пайдалану табиғи және тиімді болып көрінеді. Бұл мақалада автономды жүк көлігінің прототипі үшін компьютерлік көру модулін әзірлеу бойынша зерттеу жұмысының нәтижелері сипатталған. Орындалған жоба кезеңдері қажетті жабдықты орнатуды, компьютерлік көру үлгілерін дайындау және объектілерді жіктеу және локализациялау мақсатында камералар мен LIDAR сенсоры арасындағы картаны әзірлеуді қамтиды.

Түйін сөздер: Компьютерлік көру, автономды көлік, көлік траекториясын жоспарлау, нақты уақыттағы траекторияны жоспарлау, адамсыз шешім.

Ж.А. Есенбаев^{1*}, Ж.М. Кожирбаев², А.М. Шинтемиров³

¹Атырауский университет нефти и газа имени Сафи Утебаева, г. Атырау, Казахстан

²National Laboratory Astana, г. Астана, Казахстан

³Назарбаев Университет, г. Астана, Казахстан

*e-mail: zh.yessenbayev@aogu.edu.kz

Разработка модуля компьютерного зрения для автономных транспортных средств

Выгодное геополитическое положение и огромный транзитный потенциал Республики Казахстан в сфере наземных грузоперевозок между Китаем и Европой делает отрасль транспортной логистики одним из самых перспективных направлений развития экономики страны. В этом контексте, применения технологий беспилотного грузового транспорта для минимизации издержек от расходования топлива и использования человеческого труда в трудоёмких и рутинных операциях логистических процессов как внутри складских помещений, так и при грузоперевозках по дорогам общего пользования, видится как никогда естественным и эффективным.

В данной статье описаны результаты научно-исследовательской работы по разработке модуля компьютерного зрения для прототипа автономного грузового автомобиля. Выполненные этапы проекта включают в себя установку необходимого оборудования, обучение моделей компьютерного зрения и разработку сопоставления между камерами и датчиком LIDAR для целей классификации и локализации объектов.

Ключевые слова: Компьютерное зрение, автономное транспортное средство, планирование траектории транспортного средства, планирование траектории в реальном времени, беспилотное решение.

1 Introduction

The global transport market is estimated at about 3 trillion USD, which is almost 7% of the global GDP. For example, in Germany this figure reaches 13%, and in Ireland it reaches 14.2%, in Singapore - 13.9%, Hong Kong - 13.7%. This indicates that countries pay special attention to the development of this sector as one of the sources of national income [1]. Favorable geopolitical position and very large transit potential of the Republic of Kazakhstan in the field of land transportation between China and Europe makes the transport logistics industry one of the most promising areas for the development of the country's economy. To this end, Kazakhstan sets the task to increase transit traffic through the country by 10 times by 2050 [2].

To achieve this goal, Kazakhstan is actively putting into operation large transport and logistics centers (TLC) in key regions of the country. The development of the transport corridor "Western Europe - Western China" will allow it to become a new Silk Road, which may become a competitor to the maritime route of cargo transportation from the countries of Southeast Asia to Europe. Thus, cargo transportation along the sea route takes an average of 35-40 days, while along the transport corridor "Western Europe - Western China" the time of delivery of goods by road can be reduced by 2-3 times [3]. Such indicators, along with an increase in the volume of transit traffic, can be achieved through the digitalization and automation of the processes of cargo transportation and logistics operations. Part of the measures to develop the infrastructure of transport corridors is planned for implementation within the framework of the state program "Digital Kazakhstan" [4]. Further development of the industry involves the introduction of robotic systems to minimize or completely eliminate the use of human labor and ensure long-term or round-the-clock operation in the TLC by

logistics robots, as well as autonomous cargo transportation by autonomous vehicles along transit highways [5].

Currently, mobile robots and autonomous vehicles are increasingly being used in various sectors of the economy in developed countries. The world's leading automakers such as Tesla, Nissan, Volvo and others are already testing and offering autonomous systems to customers in serial models of cars and trucks [6–8]. The world's leading innovative companies Waymo, Yandex, Uber and others are also developing and testing technologies for autonomous vehicles.

On the other hand, unmanned technologies for the autonomy of trucks are widely used in the mining industry. Volvo was one of the first companies to implement an autonomous transport project in a mine in Norway, where six unmanned trucks operated on a 5 km long route, of which 4.7 km were tunnels. This solution made it possible to increase safety in tunnels and organize round-the-clock work [8]. Since 2015, the British mining company Rio Tinto has been using a fleet of unmanned cargo vehicles in its quarries and mines in Australia, and thereby increased productivity and reduced the cost of the extraction of natural resources [9]. In Russia, ZyfraGroup is actively engaged in the development and implementation of autonomous mining dump trucks in commercial operation [10].

The active promotion of unmanned solutions in the mining industry is associated with the relative ease of ensuring safety measures during the operation of autonomous vehicles by minimizing the presence of a person in the area of operation, the cyclicity and repeatability of operations, etc. At the same time, a much higher level of safety of autopilot systems for trucks is required for facilitating movement of autonomous vehicles on transit roads and TLC territories, along with conventional human-controlled passenger cars and trucks, with pedestrian presence in the traffic area. This area of research is currently under active development in various countries with varying degrees of readiness for testing in real conditions. One of the few examples is the successfully launched North American startup Embark Trucks, which is realizing a project for cargo transportation along the highway from El Paso (Texas) to Palm Springs (California) [11] along a 650 miles (about 1000 km) long route.

In view of the prospects for the introduction of autonomous vehicles technologies in Kazakhstan, Nazarbayev University (NU), together with the Russian company Zyfra (VIST) Group [10], with the support of KAMAZ PJSC, has recently completed a industrial project on development of an autonomous truck on the basis of a modern KAMAZ NEO platform [12].

As part of the project, partners from Zyfra (VIST) Group have equipped a test KAMAZ NEO truck with their own autopilot system for autonomous vehicle movement along a predetermined trajectory or to a specified target position with the possibility of remote control by a person from the control panel (Fig. 1). These technologies have been tested on KAMAZ and BELAZ trucks and will be adapted for a new model of the KAMAZ NEO 5490 truck (Fig. 1) with an automatic transmission. The task of the NU project group was to develop software and hardware modules for computer vision, vehicle trajectory planning with the ability to replan the trajectory in real-time to avoid obstacles and respond to the infrastructure of the traffic system (traffic signs, traffic lights, etc.). The results of the project work on vehicle trajectory planning were reported in [13].

For experimental testing of the developed software and hardware solutions on an experimental KAMAZ vehicle, a test site was created at NU on the basis of an open car parking area on the NU campus.



Figure 1: Autonomous truck based on KAMAZ Neo 5490 platform (left) and Remote control cabin designed by VIST Group (right)

The test site was equipped with road signs, dummy people and cars.

2 Hardware Setup

The hardware part of the computer vision module is a set of interconnected hardware, which is a single system designed to collect, store, process and transmit data from video cameras. The main components are:

1. Video cameras Logitech C922;
2. Wi-Fi/4G router;
3. Laptop;
4. Remote computing server.

Next, we describe the process of integrating these equipment within the truck's cabin.

The bracket and fasteners for the Logitech C922 cameras of the autopilot system inside the cabin of the Kamaz truck were assembled as shown in Fig. 2. The parts were designed in SolidWorks and made on a 3D printer from aluminum profile 40×40 mm. The cameras were attached to the frame, which, in turn, was attached to the regular power fasteners of the Kamaz cabin ceiling sheathing. The stock ceiling mount is a power mount that will not deform due to the added weight of the camera mount assembly and the cameras themselves and the gravitational forces acting on them.

3 Data synchronization between cameras and the LIDAR

The task of localizing objects using RGB cameras, installed inside the test truck cabin, and the LIDAR sensor, mounted in front of the vehicle, involves the transformation from one coordinate system to another. For this, the cameras and LIDAR were calibrated. For automatic calibration, the algorithm needs to find a specific image template. A chamotte board is usually used as the image, since the computer vision algorithm is able to easily find corners.



Figure 2: Finished assembly of fasteners for the cameras

Using the found coordinates of the corners, the matrix of internal parameters K was calculated (Equation 1). The matrix K for a hole camera consists of 5 parameters: (u_0, v_0) - optical center (principal point) in pixels; (α_x, α_y) - focal length in pixels with $\alpha_x = F/p_x$ and $\alpha_y = F/p_y$, where F is the focal length in real units, usually expressed in millimeters (p_x, p_y) is the pixel size in real units; γ is the skew factor, which is non-zero, if the image axes not perpendicular (Fig. 3).

$$K = \begin{bmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

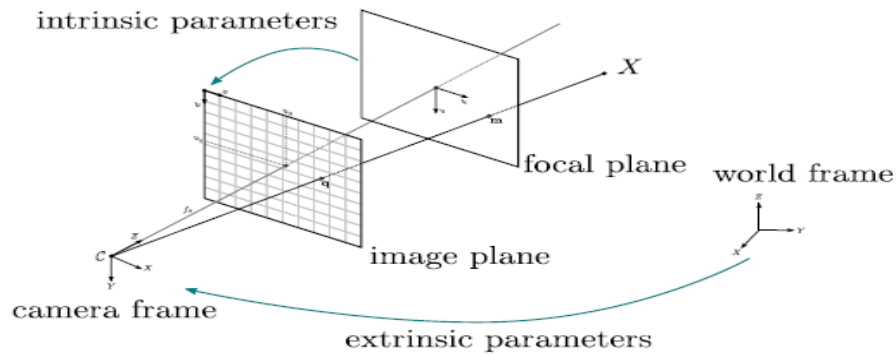


Figure 3: Point camera model

Using internal camera parameters, you can convert points from the real-space coordinate system $R3$ (x, y, z) to the camera coordinate system $R2$ (w, h) .

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R \ T] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2)$$

The Equation 2 shows the formula for converting points in the real space coordinate system to the camera coordinate system, where $[x_w \ y_w \ z_w \ 1]$ is a point in the world coordinate system, $[u \ v \ 1]$ is a point in the camera coordinate system, K are the internal parameters of the camera, z_c is arbitrary scale parameter. $[R \ T]$ - are external parameters that denote the transformation of the coordinate system from the coordinates of the three-dimensional world to the coordinates of the three-dimensional camera. Equivalently, the extrinsic parameters define the position of the camera's center and the direction of the camera in world coordinates. T is the position of the center of the world coordinate system, expressed in the coordinates of the camera-centered coordinate system.

The Zhang's model was used for calibration [14], which is a camera calibration method that uses traditional calibration methods (known calibration points) and self-calibration methods (correspondence between calibration points when they are in different positions). To perform a full Zhang calibration, it requires at least three different images of the calibration object, either by moving the object or the camera itself. If some of the intrinsic parameters are given (image orthogonality or optical center coordinates), the number of required images can be reduced to two.

At the first stage, the approximation of the estimated projection matrix H between the calibration target and the image plane is determined using the DLT (Direct linear transformation) method [15]. Subsequently, self-calibration methods are applied to obtain an image of an absolute conical matrix.

Several 4×8 chessboards were printed on A1 sheets. These chessboards were placed in the field of view of the both RGB cameras, mounted in the truck cabin, to calculate the corresponding internal parameters. Figure 4 shows 4 calibration boards.

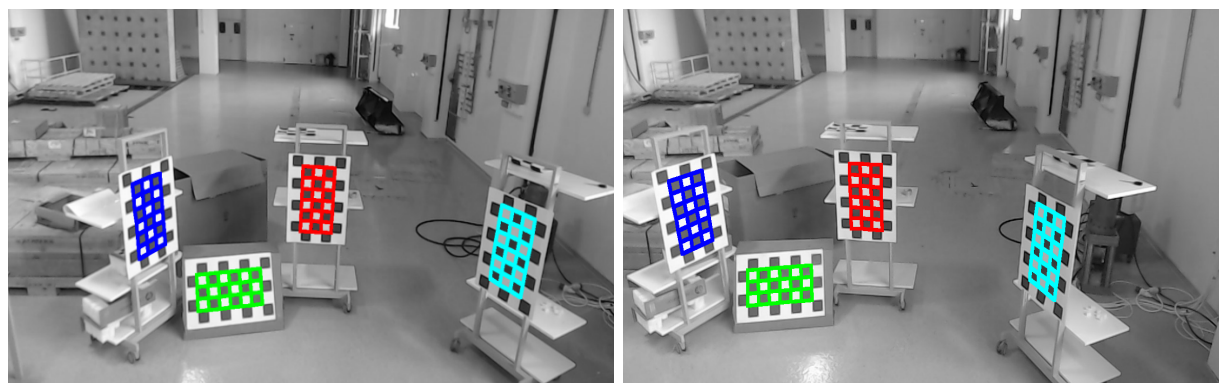


Figure 4: Example of the installed calibration chessboards

At the first stage of calibration the cvlib toolbox was used [16]. This toolbox defines the internal parameters of the cameras and determines the position and rotation between two cameras. To work, several boards were placed ce for the entire size of the frame in different orientations. The figure shows the result of finding the corners of cells in chessboards. Different colors represent different boards found. Figure 5 shows the result of matching the left and right cameras. Thus, the external parameters of both cameras were computed

In order to achieve accurate results of object recognition, the algorithms for overlaying 2D segments on 3D data from the LIDAR system were developed. To transform the coordinate

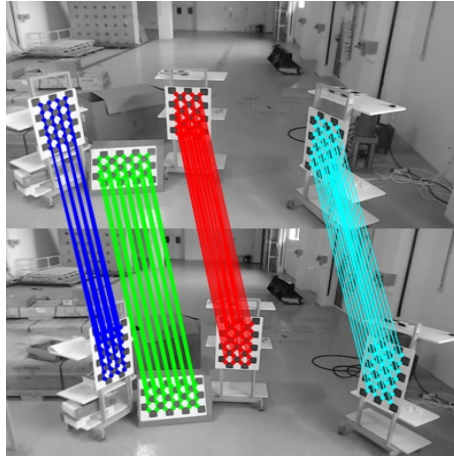


Figure 5: The result of matching two frames of the left and right cameras

system, it was necessary to calibrate the cameras and the LIDAR in order to build the transformation matrix.

Transformation matrices allow arbitrary linear transformations to be displayed in a consistent format suitable for computation. It also makes it easy to combine transformations (by multiplying their matrices).

Linear transformations are not the only ones that can be represented by matrices. Some transformations that are non-linear in the n -dimensional Euclidean space R_n can be represented as linear transformations in the $(n + 1)$ -dimensional space R_{n+1} . These include both affine transformations (such as translation) and projective transformations. For this reason, 4×4 transformation matrices are widely used in 3D computer graphics. These $n + 1$ -dimensional transformation matrices are called affine transformation matrices, projective transformation matrices, or, more generally, non-linear transformation matrices. As for an n -dimensional matrix, an $n + 1$ -dimensional matrix can be described as an augmented matrix.

In the physical sciences, an active transformation is one that actually changes the physical position of the system and makes sense even in the absence of a coordinate system, while a passive transformation is a change in the description of the coordinates of the physical system (change of base). The distinction between active and passive transformations is important. By default, by transformation the mathematicians usually mean active transformations, while physicists can mean both.

Calibration was done using the Aruco calibration toolbox [17] which provides a graphical interface for interacting with 2D and 3D images. Calibration takes place in 2 stages. The first stage is to select points in the 2D image and their corresponding points in the 3D LIDAR image. This step must be repeated several times to increase the accuracy of the results.

At the second stage, the algorithm calculates the transformation matrix from the LIDAR coordinate system to the camera coordinate system. This matrix is also known as the external parameters of the camera. An example of a calibration process we made is shown in Fig. 6.

The complete calibration scene is shown in Fig. 7. In the center, data from the LIDAR system is presented in Point Cloud format, where the intensity of each reflected beam is shown in color. The lower left picture shows data from the left camera fixed inside the truck



Figure 6: Aruco calibration toolbox example

cabin. The lower right picture shows data from the right camera fixed inside the truck cabin.

Points with the LIDAR are filtered based on the visibility area, which are set as parameters. We consider only points in front of the test truck. Then, they are transferred to the camera coordinate system using the matrix obtained earlier. Based on the segments obtained during object detection, the points are filtered again. The resulting points are the basis for the final clustering.

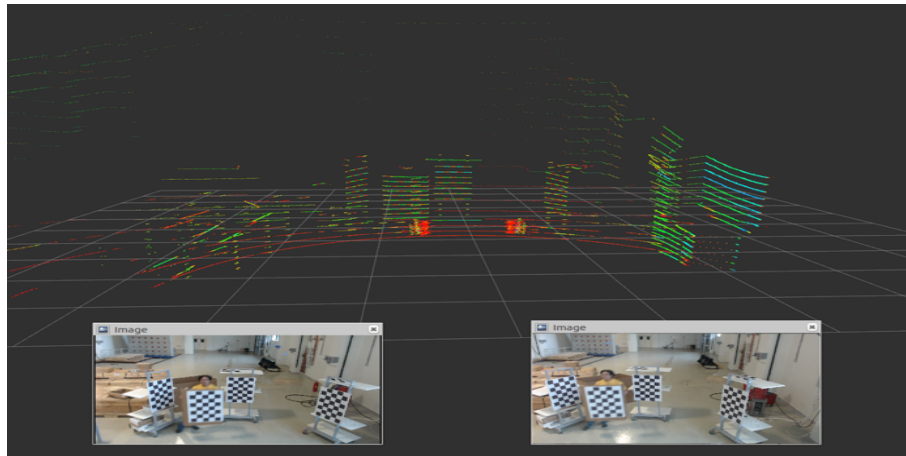


Figure 7: Full calibration scene

4 Datasets

One of the most important parts of machine learning using artificial neural networks is the collection and processing of large amounts of data. To train the object detection and localization model, it was decided to use COCO dataset [18]. The COCO (Common Objects in COntext) dataset is a dataset for training models based on different tasks: object recognition, localized objects, keypoint identification, segmentation, etc. It contains 80 classes, 80,000 training images, and 40,000 images to test the accuracy of the model (Fig. 8).

The dataset consists of several parts. The first part is the images themselves containing the objects. The second part depends on which problem the dataset is applied to. In our case, we use annotations to localize objects. Each object instance contains an annotation with a number of fields, which include a class identifier and a designation of the object's boundaries in the format of x, y coordinates, width and height.

The processing process is divided into several steps. At first, the image must be scaled using an interpolation algorithm. Secondly, it is necessary to use algorithms for data augmentation. We used a random image cropping algorithm to obtain different regions. Also, some images were mirrored horizontally. To annotate the data that was collected manually, the YoloMark tool is used [19].

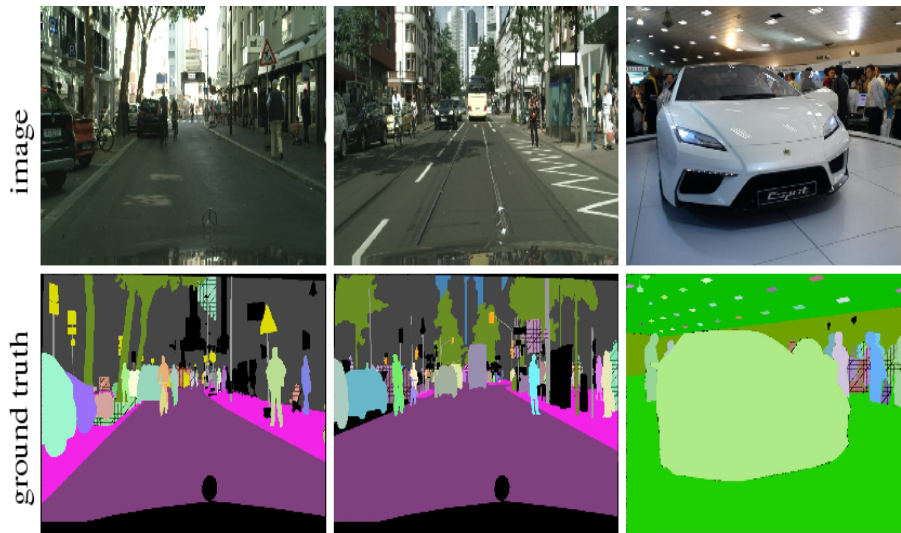


Figure 8: Examples of images in the COCO Dataset database

For a full-fledged computer vision system that will be able to move along the roads of Kazakhstan in real time, it is necessary to collect additional data on road signs. By training the traffic sign recognition model, the computer vision module can detect traffic signs and send the position of the sign relative to the vehicle to the trajectory planning module. Next, the planner must take certain actions that were prescribed for a particular sign.

After searching and analyzing existing databases, it was decided to use the RTSD road sign database [20]. The database has similar road signs as in Kazakhstan. There are 156 traffic sign classes and 104358 sign images in this dataset.

As you can see in Figure 9, the RTSD dataset has very diverse seasons and also different times of day, which of course helps retrain the model on the same type of data and increases the accuracy of sign detection.

5 Development of an object detection model

During the development of the object detection model, two models were used: YOLOv3 [21] and CenterNet [22]. Both models were trained on COCO public data.



Figure 9: Examples of images in the RTSD database

Figure 10 shows the result of the YOLOv3 recognition algorithm on data collected with an analog camera. It was revealed that the both models were able to constantly find objects in the image for classification (detection stage). A distinctive feature of these methods is that they perform detection and classification at the same time, thereby eliminating the need for re-classification. This allows the models to be suitable for real-time tasks.

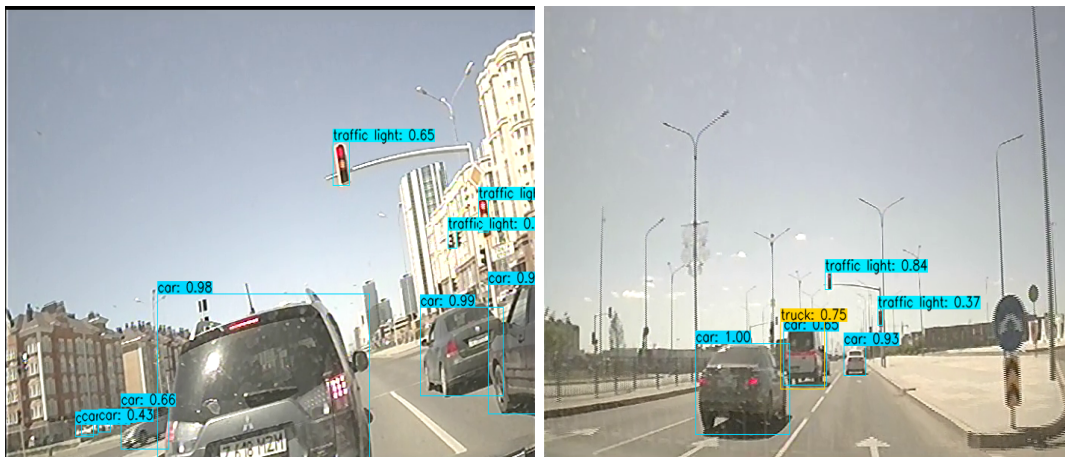


Figure 10: The result of the classification algorithm

The models were trained on the COCO dataset using data augmentation algorithms for recognition of people, cars, fire hydrants and US-style road signs. The “training” of the machine learning models were carried out on the DGX-1 deep learning cluster from NVIDIA, which consists of 8 video cards with a total video memory of 64 GB. Both models were able to recognize all the necessary objects that are found on the roads, namely cars, trucks, buses, people, signs, traffic lights, etc.

Table 1: Results of comparing the YOLOv3 and CenterNet models

Model	Size of input frame	Time to process	Average accuracy on dataset MS COCO (AP)
YOLOv3	416x416	40ms	31.0
CenterNet	511x511	60ms	37.4

During the experiments the main characteristics of the two above-mentioned algorithms were identified. The comparison result is presented in Table 1. CenterNet showed a more accurate classification. YOLOv3, in turn, is the faster model of the two presented.

6 Model training

To recognize objects with an RGB camera, it was decided to use a convolutional neural network with the CenterNet architecture (Fig. 11). CenterNet uses a system similar to YOLO. This network is a state-of-the-art architecture capable of classifying and localizing objects in 2D RGB images. CenterNet accepts a fixed-size 2D image as input, and an interpolation algorithm is used to resize the images coming from the camera. As output, CenterNet provides a 2D bounding box of objects found in the image. Unlike YOLO, CenterNet is based on a central point. The implementation of CenterNet for PyTorch was used.

CenterNet consists of several parts. The first part is preprocessing, in which the images are converted to the required format that the neural network accepts. At this stage, the image is interpolated to the dimensions 511×511 . Each of the three RGB channels of the image is normalized according to the parameters of the dataset on which the network was trained.

In the second step, the algorithm uses an autoencoder/decoder-based neural network architecture to perform semantic segmentation. The hourglass architecture with 54 convolutional layers is used as a basis.

At the third stage, the corners of the bounding box are found in parallel using the cascade pooling algorithm and the centers are found. After that, a heatmap is built for the found centers and angles. Heatmap contains information about the probability of each pixel to contain the corners and centers of objects. The data from both heatmaps are combined to get the final bounding box.

The network was trained and tested on MS COCO datasets with 50 classes of everyday objects and KITTI dataset [23] with three classes: cars, pedestrians, bicycles, collected using a car with 2 RGB cameras and equipped with a LIDAR system and designed for training and testing machine algorithms with self-government.

7 Development of methods and algorithms for localization of detected objects

After converting the points from 2D to 3D and segmenting them, it was necessary to select clusters that correspond only to the objects that need to be defined. The first step in achieving this was the segmentation of the plane that corresponds to the ground. For this, parameters were found to describe the plane. All points that lie below a certain distance are filtered. This distance can be set as a separate parameter. The remaining points are passed on. To determine the parameters of the plane, the formula was used:

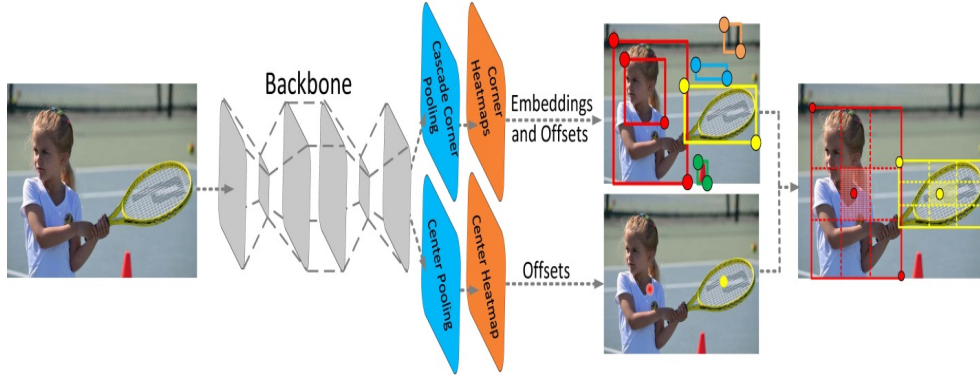


Figure 11: Stages of object detection in the CenterNet architecture

$$fit = (A^T * A)^{-1} * A^T * b \quad (3)$$

where A - x and y coordinates are a set of points belonging to the plane, b - z coordinates are a set of points belonging to the plane, fit - parameters of the plane in the format:

$$ax + by + cz + d = 0 \quad (4)$$

The remaining points can be grouped together to form separate objects. We use the Euclidean Clustering Extraction algorithm to separate points into clusters based on their proximity to each other. The algorithm uses the Euclidean distance to determine if the points belong to the same cluster. Points are considered to be in the same cluster if they are within radius r from each other. The radius r is set as a parameter. By changing it, we can control the size of the clusters. It should be noted that in our case, r must be larger than the voxel size used in the VoxelGrid downsampling step. Brute force search of points within a radius is very expensive, so the point cloud library uses the Kd tree structure to optimize the algorithm. The modified version creates a kdtree from all input points. The tree is used to find the closest points and check their relative distance. This eliminates the need to check all points in the set. At the end, the algorithm extracts a set of clusters that contain points for each feature. The clustering method divides the disorganized point cloud model P into smaller parts, so that the overall processing time for P is greatly reduced. A simple approach to data clustering in the Euclidean sense can be implemented by using a 3D grid subdivision of space using fixed-width blocks or, more generally, an octree data structure. Such a particular representation is very quick to construct and is useful in situations where either a three-dimensional representation of the occupied space is needed, or the data in each resulting 3D block (or octree leaf) can be approximated with a different structure. More generally, however, we can use nearest neighbors and implement a clustering method that is essentially similar to the flooding algorithm. Let's assume we've given a point cloud with a table and objects on top of it. We want to find and segment individual point clusters of an object that lie on a plane. Assuming we are using a Kd tree structure to find nearest

neighbors, the algorithmic steps for this would be:

1. Create a tree view Kd for the input point cloud dataset P ;
2. Initialize an empty list of clusters C and a queue of points Q to be checked;
3. Then for each point $\mathbf{p}_i \in P$, do the following;
 - (a) add \mathbf{p}_i in the current queue Q ;
 - (b) for each point $\mathbf{p}_i \in Q$ perform:
 - i. find a set P_i^k of point neighbors \mathbf{p}_i in the sphere of radius $r < d_{th}$;
 - ii. for each neighbor $\mathbf{p}_i^k \in P_i^k$, check if the point has already been processed and if not add it to Q ;
 - (c) when the list of all points in Q has been processed, add Q to the list of clusters C and reset Q to an empty list;
4. The algorithm terminates when all points $\mathbf{p}_i \in P$ have been processed and are now part of the list of point clusters C .

For each segment, the algorithm finds several clusters. By choosing the largest cluster, i.e. the cluster with the most points is the desired object. Using these points, you can find the position of the object relative to LIDAR. The position is found using the cluster centroid. The centroid is the average value of all the coordinates of a given cluster. To test the algorithm, data was collected from the test KAMAZ truck: images from the two installed RGB cameras and Point Cloud data from the LIDAR sensor. The data was written using the rosbag utility that comes with the Robot Operating System (ROS) robotics software development framework. To visualize the work of the algorithm for converting data from a 3D LIDAR coordinate system to a 2D coordinate system, an algorithm was written that, using the calibration parameters, overlays the LIDAR data and the image from the camera. An example is shown in Fig. 12.

Fig. 15 shows the visualization result of object detection and classification. In this example, the pedestrian classes have been filtered out. Fig. 16 shows the visualization of the result of the object localization algorithm. The objects are localized based on the segments shown in Fig. 13. The dots of different colors show the centers of the segmented objects.

8 Conclusion

In this paper we developed a computer vision module for an autonomous vehicle prototype based on a KAMAZ NEO chassis which was provided by our industry partners. The module is aimed at object detection and localization using an integrated system with two video cameras and a LIDAR sensor. The first task was to design and install the necessary hardware equipment. Thus, the cameras and the fasteners were installed inside the cabin whereas the LIDAR sensor with its brackets was installed on the front bumper outside the cabin. Next,

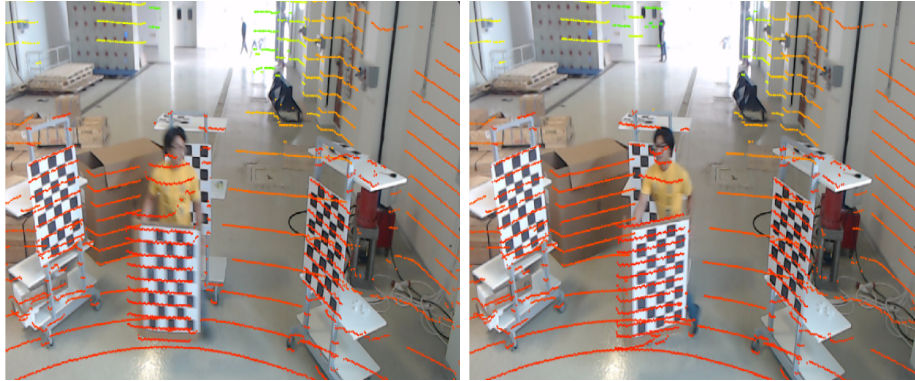


Figure 12: Visualization of the overlay algorithm

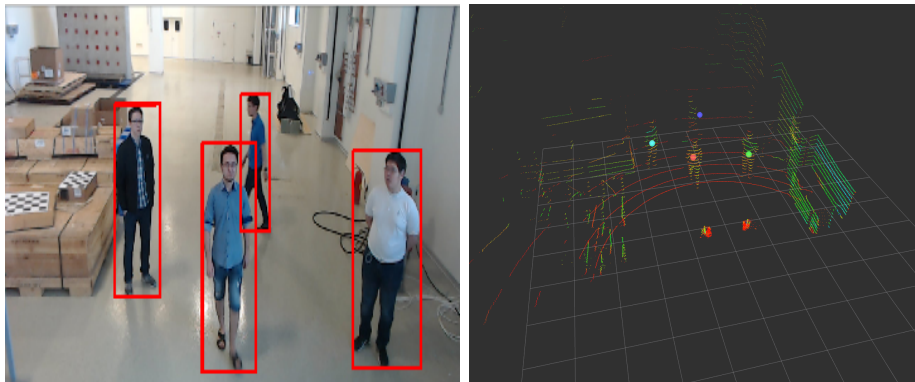


Figure 13: Visualization of the detection (left) and localization (right) algorithms

we performed the tasks of calibration and synchronization of two independent data streams, namely, 2D images from the cameras and 3D point clouds from the LIDAR sensor. Once the data streams were fully calibrated and synchronized, we developed the algorithms and trained the models for object detection in the 2D images and localized them in 3D space using information coming from LIDAR.

All the tasks were successfully completed. Currently, our computer vision module is able to detect and classify people, cars, road signs and traffic lights as well as to identify the distance to the detected object. It should be noted that the developed algorithms are suitable for any type of vehicle or mobile robotic systems that employ RGB cameras and a LIDAR sensor as their primary sources of visual information.

As a future work, we plan to develop the algorithms for obstacle avoidance, i.e., we need to replan the trajectory or stop the truck based on the situation on the road. This work was partially done but still needs to be improved.

9 Acknowledgement

The project team was partially supported by the Ministry of Education and Science of the Republic of Kazakhstan (grant project IRN AP08052091), Nazarbayev University CRP project (project №11022021CRP1502) and the industrial grant funded by Zygra (VIST) Group.

References

- [1] Iskalyev Y., “Transport logistics today сегодня is the key component in implementing of the State program Forced Industrial Development”, accessed: 07.07.2022, URL: <http://portal.kazlogistics.kz/analytics/95/708/>.
- [2] Nazarbayev N., “State program «Strategy «Kazakhstan — 2050»: new political course of a developed country”, accessed: 07.07.2022, URL: <https://online.zakon.kz>.
- [3] “What is the importance of logistics development for the economy of Kazakhstan”, accessed: 07.07.2022, URL: <https://forbes.kz/finances/markets/birthday/>.
- [4] “Official website of the State program «Digital Kazakhstan»”, accessed: 07.07.2022, URL: <https://digitalkz.kz>.
- [5] “Transport logistic centers of Kazakhstan, what was done?”, accessed: 07.07.2022, URL: <http://atameken.kz/ru/articles/27077-transportkazakhstana>.
- [6] “Official website of Tesla”, accessed: 07.07.2022, URL: <https://www.tesla.com/autopilot>.
- [7] “Nissan tests fully autonomous prototype technology on streets of Tokyo?”, accessed: 07.07.2022, URL: <https://newsroom.nissan-global.com/releases/release-1fc537356ae3aaf048d0201b77013bf9>.
- [8] “Autonomous trucks in real operation”, accessed: 07.07.2022, URL: <https://www.volvotrucks.com/en-en/news/volvo-trucks-magazine/2019/feb/bronnoy.html>.
- [9] “Mining automation: The be all and end all?”, accessed: 07.07.2022, URL: <https://www.australianmining.com.au/features/mining-automation-the-be-all-and-end-all/>.
- [10] “Robotized technique for mining and industrial companies”, accessed: 07.07.2022, URL: <https://vistgroup.ru/solutions/robotizirovannaya-tehnika/>.
- [11] “Self-Driving Trucks Are Now Delivering Refrigerators”, accessed: 07.07.2022, URL: <https://www.wired.com/story/embark-self-driving-truck-deliveries/>.
- [12] “A new project for development of an autonomous vehicle based on KAMAZ NEO platform started at Nazarbayev University”, accessed: 07.07.2022, URL: <https://nu.edu.kz/ru/news-ru/v-nazarbaev-universitete-startoval-proekt-poszdaniyu-robotizirovannogo-avtomobilya-kamaz-neo>.
- [13] Abilkassov Sh., Nurlybayev A., Soltan S., Kim A., Shpieva E., Yesmagambet N., Yessenbayev Zh., Shintemirov A., “Facilitating Autonomous Vehicle Research and Development Using Robot Simulators on the Example of a KAMAZ NEO Truck”, *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, (2020):1-8.
- [14] Zhang Zh, “A Flexible New Technique for Camera Calibration”, *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11), (2000):1330-1334.
- [15] Abdel-Aziz Y.I., Karara H.M., “Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry”, *In Proceedings of the Symposium on Close-Range Photogrammetry*, 8 (1971):1-18.
- [16] “cvlib - a simple, high level, easy to use, open source Computer Vision library for Python”, accessed: 07.07.2022, URL: <https://www.cvlib.net/>.
- [17] “ArUco: a minimal library for Augmented Reality applications based on OpenCV”, accessed: 07.07.2022, URL: <https://www.uco.es/investiga/grupos/ava/node/26>.
- [18] Lin T.Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollr P., Zitnick C.L., “Microsoft COCO: Common Objects in Context”, *European conference on computer vision*, (2014):740-755.

-
- [19] “YoloMark - Windows and Linux GUI for marking bounded boxes of objects in images for training Yolo v3 and v2”, accessed: 07.07.2022, URL: https://github.com/AlexeyAB/Yolo_mark.
- [20] “Russian Traffic Sign Dataset”, accessed: 07.07.2022, URL: <http://graphics.cs.msu.ru/en/research/projects/rtsd>.
- [21] Redmon J., Divvala S., Girshick R., Farhadi A., “You only look once: Unified, real-time object detection”, *In Proceedings of the IEEE conference on computer vision and pattern recognition*, (2016):779-788.
- [22] Duan K., Bai S., Xie L., Qi H., Huang Q., Tian Q., “CenterNet: Keypoint Triplets for Object Detection”, *In Proceedings of the IEEE/CVF international conference on computer vision*, (2019):6569-6578.
- [23] Geiger A., Lenz P., Stiller C., Urtasun R., “Vision meets robotics: The KITTI dataset”, *International Journal of Robotics Results*, 32(11)(2013):1231–1237.