The use of open and machine vision technologies for development of gesture recognition intelligent systems

Kirill V Cherkasov¹, Irina V Gavrilova², Elena V Chernova², Andrey S Dokolin³

¹ ANDROID TECHNICS, 6, Herzen street, Magnitogorsk, 455000, Russia
² Nosov Magnitogorsk State Technical University, 38, Lenin ave., Magnitogorsk, 455000, Russia
³ Magnitogorsk secondary school №28, Carl Marks Ave., 141/4, Magnitogorsk, 455038, Russia

E-mail: Irina.V.Gavrilova@yandex.ru

Abstract. The article is devoted to reflection of separate aspects of intellectual system gesture recognition development. The peculiarity of the system is its intellectual block which completely based on open technologies: OpenCV library and Microsoft Cognitive Toolkit (CNTK) platform. The article presents the rationale for the choice of such set of tools, as well as the functional scheme of the system and the hierarchy of its modules. Experiments have shown that the system correctly recognizes about 85% of images received from sensors. The authors assume that the improvement of the algorithmic block of the system will increase the accuracy of gesture recognition up to 95%.

1. Introduction

At present, robotics is one of the fastest growing branches of science and production. Many problems related to the development of robotic systems nodes have been left behind, and the problem of developing intelligent robot control systems has emerged, which is especially important for anthropomorphic systems because their purpose is often of a social nature. Increasing the effectiveness of training anthropomorphic robots movement process and their interaction with the surrounding objects is now one of the most urgent tasks of robotics, the rapid development of which causes the need to develop new effective approaches to the training of robotic systems.

Traditional training of anthropomorphic robot is a rather long process of creating scripts that describe the movement of its nodes in space. Of course, a person without special training and without specialized programming environments will not be able to cope with this task. It would be simpler and faster if a system that recognizing a person's gesture in a video stream would independently make a program that repeats it, and remember its meaning – in this case, for training anthropomorphic robot is not needed highly qualified programmer.

The study showed that today there are no developed approaches to the effective use of heterogeneous sensory systems, supplying the information caused by various reasons.
2. Technologies

The developed gesture recognition system processes two-dimensional image obtained as a result of
the work of the LIDAR sensors installed on the robotic system designed for laser scanning of three-
dimensional space. Laser LIDAR is a technology of receiving and processing information about
remote objects with the help of active optical systems, using phenomena of light and its dispersion
reflection in translucent environments. The hardware platform for the development of the system was
made up of computers equipped with Nvidia graphics cards based on nuclei CUDA (Compute Unified
Device Architecture). The modular system "Bundle" was used as the emulation medium, which allows
unifying the control process by various robotic devices regardless of the low-level software loaded
into the robotic system. For image recognition was chosen as an open source library OpenCV and its
subsequent revision. Among available open source solutions in the field of machine learning was as
follows:

- Open Source Machine Intelligence library TensorFlow from Google;
- Caffe is an open-source deep learning framework from the laboratory of Artificial Intelligence
  Studies at the University of California Berkeley;
- scientific computing framework with open source Torch created by a small group of experts in
computer vision;
- Theano is an open library of machine learning at the University of Montreal, written in
Python. Allows to define, optimize and calculate mathematical expressions with effective
involvement of multidimensional arrays;
- Microsoft Cognitive Toolkit (CNTK)-a set of tools for open source learning from Microsoft;
- MXNet – developed by the community of deep machine learning framework for the
construction and training of neural networks with open source.

The choice of the system was based on the following criteria: priority development languages C++ and
Python, open source, good documentation, ease of development, support for key points – the system
must have built-in support for working with the image. The results of the analysis are summarized in
table 1.

Table 1. Comparative characteristics of machine learning systems

| Parameter/System                      | TensorFlow | Caffe | Torch | CNTK | MXNet | Theano |
|---------------------------------------|------------|-------|-------|------|-------|-------|
| Open source                           | +          | +     | +     | +    | +     | +     |
| Documentation                         | +          | +     | -     | -    | -     | +     |
| Ease of development                   | +          | +     | -     | -    | +     | +     |
| Key points                            | -          | +     | -     | +    | -     | -     |
| Working on a video processor          | +          | +     | +     | -    | +     | -     |
| Programming language                  | Python, C++| C++, | C++, | C++, | R, C++ | Python |
|                                       | Java       | Python | lua   | R    | perl, | Python, |
|                                       |            |        |       |      | Scala |       |

Multi-GPU capability                   | +          | +     | +     | +    | +     | -     |

However, the importance of the system choice had a different setting – the speed of real-time
(frame/sec.). For this purpose, a number of trial tests were carried out on the working machine to
calculate the speed of the frameworks in a simple problem of classification and localization of the
human face image. To train the neural network was used public dataset SCface - a database of random
faces of people taken on surveillance cameras, containing 4160 static images of 130 people shot in the
infrared and visible spectrum, as well as data coordinates for these images, with coordinates X and Y
points of the centers of the eyes, nose and mouth.

**Figure 1.** Diagram of machine learning framework test results.

CNTK showed undeniable advantages in speed, especially when using two computers. As a result, it was decided to focus on the Computational Network Toolkit system, despite the complexity of development and poor documentation.

3. **Implementation**

The main functions of the developed system are data collection from the camera, data processing and matching, subsequent recognition. Data collection can be performed either in real time with the video camera connected, or with a recorded avi or mp4 file. In the first case, you need to specify the parameters of the camera, in the second – the name and path to the read file.

As new data packets are received, a signal is generated that initiates pre-processing and mapping.

Preprocessing mainly involves filtering the image to reduce the parasitic effects of individual data groups on the matching process. If necessary, it is possible to analyze the passableness, as well as the correction of the point cloud using the data of the inertial navigation system (but only when working with the laser sensors LIDAR).

After the preprocessing is complete, the data is mapped using the key point algorithm. The algorithm core is executed on the GPU of a personal computer, so when you run the algorithm, you must place all the necessary data arrays in the graphics card memory. In the case of a successful mapping, the desired transformation is returned, which is added to the corresponding array of transformations used in the subsequent recognition and definition of the human skeleton.
At the stage of recognition, the presence of a person or several people in the image is determined, and in the positive case, their skeleton is compiled. If necessary, the user can turn off the visualization of individual or all components to reduce the consumption of computing resources.

The developed system consists of several interacting modules in JSON format, each responsible for its area of operation and is associated with other modules within the system. With the model of the system can be found in figure 3.

Through the interface module "Bundle" the system receives video information from the cameras installed on the robot. This information is then transferred to the video processing unit, where it is processed by image adaptation algorithms. Here in the picture noises are removed, brightness, color balance and contrast are regulated. The image is converted to a format suitable for the recognition module. Further, in parallel with the training module, in the module recognition is pattern recognition, and dynamic classification of the object, because the system applies machine learning algorithms and is in a constant process of learning. Then the information about the recognized object is passed to the...
"Bundle", which decides how to behave with the received object.

A model or processed image obtained from robotic platform (RTP) cameras, which, as a rule, has natural or artificial boundaries, is a mathematical description of the image features, such as wall lines in corridors, shapes and coordinates of different people and objects, recorded and approximated data of all image objects, with the training movement of RTP, etc. With the joint use of RTP cameras and LIDARs, a three-dimensional map of the space surrounding the work required for RTP autonomous trajectory control is constructed. A priori information contained in it is constantly compared with the current data of the system, on the basis of which the robot localizes itself on it, builds a forecast of future events, calculates and implements the necessary control program. The approximation allows us to obtain an adequate mathematical model of the transient characteristic. The solution of the general approximation problem consists of three separate stages. At the first stage it is necessary to choose and justify the functional structure of the approximating mathematical expression. At the second stage it is necessary to calculate the approximation coefficients for the selected mathematical function according to the available experimental data. At the third stage it is advisable to assess the accuracy of the approximation. The algorithm of approximating the 3D rendering works as follows: upon receipt of a scanned point cloud, select the point group of the free and that is a mathematical treatment of the least squares method. Then, during the error detection phase, the condition is checked: if the error does not exceed the threshold, then the nearest points must be added. If the error exceeds the threshold, the condition is met by a sufficient number of points. If the number of points is insufficient, it is necessary to once again choose the point group of the free, if the number of points is sufficient, it is necessary to calculate the boundaries of the plane and divided into polygons, and then to visualize the image.

Tests have shown that the developed intelligent gesture recognition system accurately identifies up to 85\% of objects with the construction of the skeleton and the position of the brushes.

4. Conclusion

Thus, the use of open source libraries for pattern recognition is not enough to obtain acceptable accuracy of the gesture recognition. It is necessary to provide possibility of more exact adjustment of the algorithm of training, and also the module allowing to correct results of a nickname. In any case, even 85\% of correctly identified gestures increase the speed of learning the system tenfold, since entering one gesture manually on average requires about 20 instructions to set the initial position and up to 50 instructions to describe its dynamics. In future we plan to improve the quality of recognition up to 95\% due to the introduction of additional heuristics in the learning algorithm. Of course, this would not be possible with the use of closed source technology.

References

[1] Robert Laganière 2011 OpenCV 2 Computer Vision Application Programming Cookbook. (Packt Publishing)
[2] Coelho L P and Richert W 2015 Building Machine Learning Systems with Python
[3] Gaurav S, Frederic J, Cordelia S Expanded Parts Model for Semantic Description of Humans in Still Images, IEEE Transactions on Pattern Analysis & Machine Intelligence 39 87-101
[4] Popova I V Subochev A V 2011 Development of the trained specialized information retrieval system, Software and Systems 3 22