ATLANTIS PRESS

Advances in Computer Science Research (ACSR), volume 72

IV International Research Conference "Information Technologies in Science, Management, Social Sphere and Medicine"

(ITSMSSM 2017)

Quadrocopter Control in Auto Tracking Mode*

Andrey Mikhaylov, Polina Kovalenko, Alexander Kataev
Department of CAD
Volgograd State Technical University
Volgograd, Russia
a.mikhaylov.v@gmail.com, p.kovalenko.o@gmail.com,
alexander.kataev@gmail.com

Elena Maksimova, Anna Alekseenko
Department of Information Security
Volgograd State University
Volgograd, Russia
maksimova@volsu.ru, annalekse1@yandex.ru

***Abstract*—In this work the following has been done: ROS configuring, installing of catkin, exploring of Parrot AR.Drone 2.0 application programming interface (API), quadrocopter and computer data exchange through Wi-Fi. Result of work is a project with implemented different object tracking methods: Histogram Back Projection and Template Matching. Mouse listener function to capture the object for tracking has been implemented. Implemented mapping of unknown terrain through monocular vision of quadrocopter (PTAM). Methods of quadrocopter control for moving through optimal path considering vector of tracking object movement has been developed based on implemented video stream object tracking methods.**

Keywords—quadrocopter; ROS; catkin; AR.Drone; object tracking; OpenCV

I. INTRODUCTION

At present, with the development of information technologies in the computer vision area, it is possible to control a quadrocopter without the help of an operator. There are various libraries that allowing to solve particular tasks of computer vision, remote and automatic control of aircraft. But the laboriousness of their usage remains very high, since available algorithms require manual tuning and optimization.

The purpose of this work is solving the problem of visual search and tracking of the object, as well as aircraft control based on it, which engineers and researchers can use to implement their projects.

The whole scope of work was decomposed into separate tasks, which can be divided into two groups. The first is working with visual searching and tracking an object on the video stream received from the camera. The second group includes the quadrocopter control and everything connected with it.

The objects that tracked with quadrocopter were mobile (in motion), and also had unique lineament (features) compared to the scene.

It is important to remember about the information security of the system. It concerns such things as the quadrocopter and computer data exchange through the Wi-Fi network. The scope of quadrocopter usage is not limited to video monitoring or mapping. Quadrocopters can be used to search for victims in places of catastrophe or for military reconnaissance purposes, flying in automatic tracking mode. So they must meet the requirements for safety and integrity of the system.

*The research work carried out as part of the grant to the winner of the program «UMNIK MIPT-autumn 2015» under the contract № 9886ГУ/2015 of 10.03.2016 (code 0020827) with the project «Development of methods for visual search and track targets using quadrocopters»

II. ROS DESCRIPTION AND CONFIGURING

A. Robotics operation system

The Robotics operation system (ROS) is a robot programming framework that provides functionality for distributed work [1]. ROS was originally developed in 2007 under the name switchyard in the Laboratory of Artificial Intelligence of Stanford University for the project (STAIR).

In fact, ROS is a set of various widely known (and not so well-known) libraries, such as OpenCV [2], PCL, Ogre and Orocos.

The main advantage is the client-server architecture of ROS – the developers implemented a mechanism for sending messages between different objects, the ability to build distributed systems, providing bridges to C ++ and Python.

The basic concepts of the ROS file system are the package, stack, node and topic [3]. The purpose of this structuring is completely transparent – increasing usability and reusability. There is an example of a structure on Fig. 1.

ROS has a complex file system. Users are provided with a set of utilities instead of long paths to different directories. It is statics. The dynamics in ROS described by nodes and by topic. A node is a running process, which knows how to communicate with other processes. A topic is a named pipe connected to different nodes. Nodes and topics form an asynchronous data exchange mechanism.

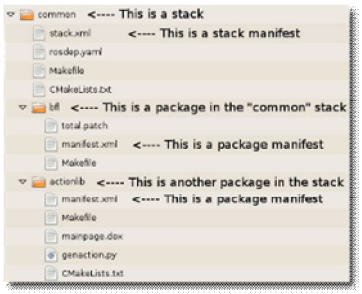

Fig. 1. Package stack structure in ROS

B. Ardrone_autonomy driver

ardrone_autonomy is a ROS driver for Parrot AR.Drone 1.0 and 2.0 [4]. This driver is based on official AR.Drone SDK 2.0.1.

ardrone_driver is an executive node of driver. It can be launched through command "rosrun ardrone_autonomy ardrone_driver" directly or through file to launch with desired parameters.

Both AR-Drone 1.0 and 2.0 are equipped with two cameras. One front camera and one vertical camera pointed down. Information about the calibration of the camera is provided either as a set of ROS parameters, either through the files ardrone_front.yaml and/or ardrone_bottom.yaml. Information about the calibration will also be published on the topics camera_info.

C. Working directory catkin

The working directory of catkin is provided by default when installing ROS. Catkin can also be installed from original packages. The installation of catkin through the command:

"sudo apt-get install ros-indigo-catkin".

Catkin is a build system based on CMake and extended with Python [5]. It adds functions to create distributed base codes and is the successor of rosbuild.

CMake is a cross-platform family of open source tools for creating, testing and packaging software [6]. CMake is used to manage the compilation process of the software using simple configuration files and compiler and also to create your own make-files and workspaces that you can use in the compiler environment of your choice. The CMake toolkit was created by Kitware because of the need to create a powerful cross-platform build environment for open source projects such as ITK and VTK.

III. SETTING UP THE PROJECT FOR ROS

A. Project compilation

The project is compiled by assembling the entire working area of the catkin. It can be done by going to the working directory catkin (Fig. 2) and by assembling the workspace (Fig. 3).

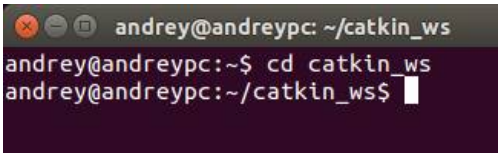

Fig. 2. Transition to catkin working directory

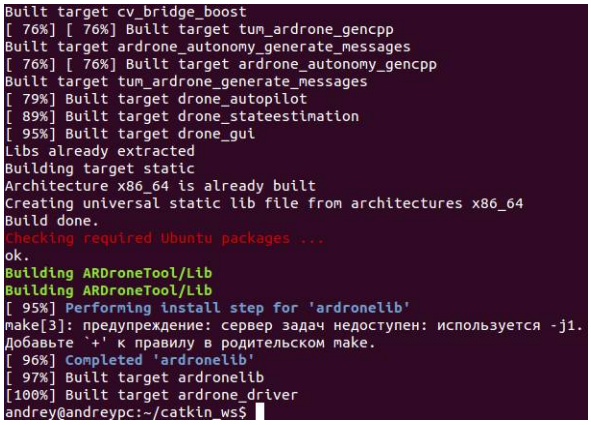

Fig. 3. Build through catkin_make command

B. Connecting the quadrocopter to the computer via Wi-Fi. Secure communication

Connection to Ar.Drone 2.0 quadrocopter is performed through the secure Wi-Fi network using the ardrone_driver package.

In order to establish communication quadrocopter should be turned on by connecting the battery. Then connect the computer to quadrocopter's Wi-Fi. The implementation of these actions means only that the connection between the computer and the quadrocopter is established, but this does not allow data exchange with the quadrocopter.

The launch file, which, using the open api quadrocopter (Fig. 4), allows you to configure access to the data received from the quadrocopter, as well as sending the necessary data from the computer to the quadrocopter. In this launch file, the frequency of the data update is indicated. Data from sensors and cameras will be transmitted in real time.

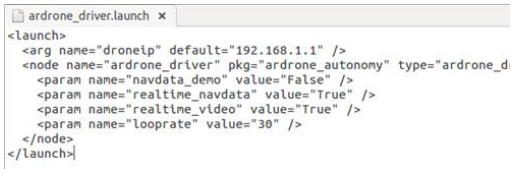

Fig. 4. Content of ardrone_driver.launch file

C. Implementing the mouse listener to capturing the tracking object

For the initial determination of the object, which will be tracked, it is necessary to manually select the object on video stream [7]. To do this, the mouse listener was implemented for the window, in which the video stream received from the quadrocopter camera will be played in real time.

The purpose of this listener is setting the signals with mouse button and performing certain actions with different input factors (left mouse button pressed / left mouse button released / mouse move).

After pressing of left mouse button (LMB) in the window with the video stream while the program is running, then the coordinates of this click will be saved. Moving mouse with pressed LMB draws the bounding box. After releasing the LMB, the coordinate are saved, and rectangle is created by two points. This rectangle defines the area contained the object.

IV. TRACKING METHODS IMPLEMENTATION

A. Histogram Back Projection

This method is used for image segmentation or searching objects that differ in color from the background. Histogram Back Projection (HBP) creates a mask according to the probability of the pixels belonging to the object, and by the size, which equal to input image in single-channel mode. Each pixel corresponds to the probability of belonging to the desired object. The output is a probability map of the object location on the image. Pixels from white to black indicate this probability – white shows that the probability of belonging to a given object is high, black – low probability [8].

B. Template Matching

The template matching is a high-level method that determines the details the user needs in the image, according to the previously selected template. The methods of matching with the template are flexible and easy to use. The method Template Matching (TM) [9] can be used in detecting large and complex patterns, but in this case, their applicability is limited by the computing power of the equipment.

At the input of the template matching method needs a frame from the video and the template (reference) of the object that you want to find in the image.

C. Implemented tracking methods validation

The marker (Fig. 5) has been chosen for validation.

This test was conducted on the basis of implemented object tracking methods (HBP and TM). To test the program it is necessary a mp4 video file, a text file containing information about the location of the object on each frame of this video, which has 4 numbers in each line: "X-coordinate Y-coordinate width height" (Fig. 6). The program can calculate the statistics of the selected object tracking method with this set of input data. If necessary, only to see the work of object tracking methods without method work statistics, you can do that without a video file and a text file, just by connecting a webcam.

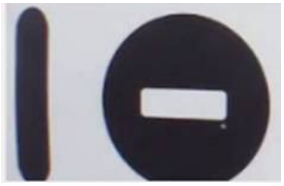

Fig. 5. Marker

```
185 242 123 76
183 242 117 75
174 242 124 79
176 243 114 79
168 243 127 80
169 243 119 78
166 244 118 72
155 239 118 76
151 239 124 77
148 237 123 77
```

Fig. 6. Fragment of text file content

The quality of the work of an algorithm will be evaluated according to the following metrics:

- Average executing time of the algorithm;
- Maximum executing time of the algorithm on one frame;
- Standard deviation of the object's coordinates from the real coordinates represented in the text file;
- Maximum deviation of the object's coordinates from the real coordinates of the object.

The algorithms executing time is estimated in seconds, the maximum deviation of the coordinates of the object from real coordinates is estimated in pixels.

The frame from the recorded video of the experiment is shown in Fig. 7, which shows the scene received from the camera quadrocopter. Tracking marker is highlighted in green, which indicates the successful tracking of the object in real time using the quadrocopter camera.

Comparative statistics of Histogram Back Projection and Template Matching methods work are presented in Table 1.

TABLE I. COMPARATIVE STATISTICS OF METHODS WORK

Quality metrics	Histogram Back Projection	Template Matching
Standard deviation of the X	2.144	2.391
Standard deviation of the Y	1.441	1.693
Max. deviation of the X (in pixels)	10.500	9.500
Max. deviation of the Y (in pixels)	11.000	6.500
Average executing time (in seconds)	0.005	0.075
Maximum executing time (in seconds)	0.008	0.112

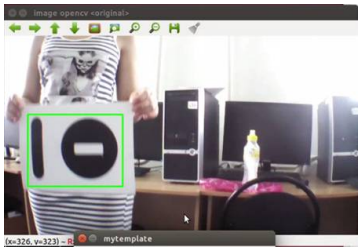

Fig. 7. Frame from experiment video

V. THE APPLICATION OF MONOCULAR SLAM (PTAM)

Simultaneous Localization and Mapping (SLAM) is a method used in mobile standalone tools to build a map in an unknown space or to update a map in a previously known space while monitoring the current location and the distance traveled. The popular methods for the approximate solution of this problem include a particle filter and an extended Kalman filter [10].

To build an environment map using AR.Drone, you can only use the front camera. This directly leads to the using of monocular vision, namely monocular SLAM.

The most of monocular visual odometry algorithms for small UAVs rely on parallel tracking and mapping (PTAM) technology [11]. PTAM, in turn, is based on SLAM, which provides reliability by tracking and displaying hundreds of control points. It works in real time and simultaneously performs the tasks of displaying destination points and estimating the movement, based on an effective correction based on image processing from different viewing angles [12].

VI. THE REALIZATION OF METHODS FOR CONTROLLING THE QUADRUPTER FOR MOTION ALONG THE OPTIMAL TRAJECTORY

After processing each frame of the video coming from the camera of the quadrocopter, the system receives the calculated position data of the object, which returns a particular implemented tracking method. Using this information, the conclusion is made about changes in the size of the object relative to the original, and also about the deviation of the object from the center of the scene [13].

Knowing that the object has moved away or has approached and where it has moved, the system transfers the necessary information to the quadrocopter, so that it moves in a certain direction - in the direction of the object's displacement. In order to minimize the loss of the object sought and go beyond the scene, it was decided to first center the object, and only then approach it or move away, depending on the input data.

VII. RESULTS OF WORK

As a result of the work, the configuration of the project under ROS on the Ubuntu operating system was studied. After studying the api of the available quadrocopter, it was possible to establish a secure connection with the Parrot Ar.Drone 2.0 quadrocopter via Wi-Fi with a computer, and also the exchange of data between them.

The principles of OpenCV library methods operation were revealed both on the Ubuntu operating system, and when writing projects for ROS, and ways of interaction of this computer vision library with them.

A project was compiled with the processing capabilities, it was obtained from the real-time quadrocopter using the OpenCV library, and object tracking methods such as Histogram Back Projection and Template Matching were implemented, which showed stable operation with real input data. Comparative statistics of the work of tracking methods shows that the HBP method works more accurately and faster than the TM method, but the TM method surpasses it in stability. In this case, the selection of the tracking object is realized through the listener of the mouse on the window with the streaming of the frames from the camera of the quadrocopter.

Also, within the framework of the work done, the Ar.Drone 2.0 quadrocopter is implemented in automatic tracking mode by sending commands to the motors of the quadrocopter.

VIII. CONCLUSION

This work can be used in security structures and organizations, whose powers include the implementation of security, and it can be useful to zoologists, with the safe tracking of wild animals, journalists, in the filming of reports from the hard-to-reach places. The methods of tracking the object and controlling the quadrocopter realized in this work make it possible to use a quadrocopter in the automatic flight mode for a mobile target.

The proposed solution for implementing the tracking system of the selected object by a quadrocopter in an off-line mode differs from the existing approach to the development of this system, the use of an add-on in the form of the ROS operating system, which opens wide opportunities for interaction with various robots, including Parrot Ar.Drone, which was used in this work.

REFERENCES

[1] ROS. Available at: http://www.ros.org/.

[2] OpenCV. Available at: http://opencv.org/.

[3] Basics of work in Robotic Operating System. Available at: https://geektimes.ru/post/128024/.

[4] Ardrone_autonomy / Official. Available at: http://ardrone-autonomy.readthedocs.io/en/latest/index.html.

[5] ROS.org / catkin. Available at: http://wiki.ros.org/catkin#Installing_catkin.

[6] CMake. Available at: https://cmake.org/.

[7] V.V. Myasnikov, N.I. Glumov and V.V. Sergeev, “Methods of detection and recognition of objects in digital images,” Samara: Samara Publishing House. state. aerospace. Un-ta; 2006, p. 168

[8] T. Gevers, J. van de Weijer and H. Stokman, “Color Features Detection,” University of Amsterdam, CIP, 2006, Chapter 1.

[9] Template Matching. Available at: http://docs.adaptive-vision.com/current/studio/machine_vision_guide/TemplateMatching.html.

[10] A. G. Finogeev, D. S. Parygin and A. A. Finogeev, “The convergence computing model for big sensor data mining and knowledge discovery,” Human-centric Computing and Information Sciences, 2017, vol. 7, Art. no. 11. Mode of access: https://hcis-journal.springeropen.com/articles/10.1186/s13673-017-0092-7

[11] I.A. Kalinov, Creation of a small autonomous unmanned aerial vehicle on the basis of a multi-rotor using the algorithm of a semi-line visual odometry, Intelligent systems, control and mechatronics-2016: mater. Vseros. Научн.-технич. Conf. Sevastopol: Publishing House of SevGU, 2016, pp. 150-154.

[12] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” IEEE and ACM Int. Sympos. on Mixed and Augmented Reality, Nov. 2007, pp. 1–10.

[13] N. Sadovnikova, D. Parygin, M. Kalinkina, B. Sanzhapov and Trieu Ni Ni, “Models and methods for the urban transit system research,” CIT&DS 2015 : Proceedings of the First International Conference on Creativity in Intelligent Technologies & Data Science, Volgograd, Russia, 15–17 September 2015, Springer IPS, 2015, CCIS 535, pp. 488–499.