

N.A. Yesmagambet

*Ye.A. Buketov Karaganda State University
(E-mail: nurjikyes@gmail.com)*

Implementation GStreamer framework with face detection system for Unmanned Aerial Vehicle

This article provides an experimental development of unmanned aerial vehicle (UAV), and the method by which a video stream from the UAV is transmitted through the pipeline, using GStreamer framework, which in turn takes up in the ground station. Preprocessed data video link will define the face using Viola Jones method through OpenCV library. During research was modeled drone with the flight controller based hardware computing platform «Raspberry Pi» and «Arduino» with complex algorithms based on Open Source projects «ArduQuad, Arducopter, MultiWii» on the C++ programming language, Python. A pipe through which is passed a raw data stream videos via WiFi c UAV ground station network.

Key words: UAV, face detection, opencv, GStreamer.

To date, being developed object recognition systems in the field of computer vision. This article is offered the new option of using an object recognition method, in particular a human face on unmanned aerial vehicles. Unmanned aerial vehicles has a wide range of applications such as a hobby, aerial photography, search, even the delivery of products. It is equally possible variations of applications in various fields of industry and services. Unfortunately due to its unpopularity associated with the complexity of the subject area there is no any complete intelligent operating system which could provide adequate positive result in one or another sector for the UAV, which would work in outdoors without support of the expensive components (for instance vision tracking system). In particular the case if you want to rescue a man missing in the mountains or wilderness, you need to spend a lot of time and resources (both human and technical), while the UAV with human detection module is a potential solution of this problem.

Unmanned aerial vehicle, also known as UAVs are unmanned aircraft that is controlled remotely or autonomously based on pre-programmed flight plans. Typically, these types of vehicles are used for military purposes for missions that are too dangerous for manned aircraft. They are also used in applications such as aerial photography and transportation of various cargoes. UAV type — helicopter has advantages over fixed-wing UAV (planner), they can take off and land vertically, allowing you to hover at a fixed point. In recent years there has been rapid development of autonomous systems, unmanned aerial vehicles (UAVs) and micro aerial vehicles (LAM). Research and development are highly supported at the moment, because their application can be applied to a variety of areas such as search and rescue, military, cinema production, agriculture and others. The main advantages of using UAVs for aerial surveillance, reconnaissance and inspection in difficult and hazardous conditions. In addition, many other technological, economic and political factors contributed to the development and operation of UAVs. Compared with other types of copter, it was decided to choose the type of quadcopter. Quadcopters of its kind - a golden middle, depending on the weight, upgrades, installation of additional modules, power lifting the payload, cost, complexity of installation, maintenance, flight time.

Quadcopters (quadcopter) — is a balanced vehicle with four rotors the same size moving through the speed control motor with propellers (Fig. 1). Through the use of several rotors provides greater flexibility and speed and the ability to hover. Each of the rotors on quadcopters produces thrust and torque. Given that the front and rear motors to rotate counter-clockwise, and the other two rotate clockwise as a result of the torque is close to zero. Until recently, multicopter development was

stagnating. Manage four independent rotors it was incredibly difficult and impossible without the aid of additional computing power. Reducing the cost of modern microprocessors gave impetus to the development of autonomous systems for multicopter, making it possible to use them in commercial, research, military, and even amateur purposes [1]. Practical part of this article considering crafting the prototype of the quadcopter and implement it with GStreamer framework.



Figure 1. Model of quadcopter

Quadcopters (quadcopter) — a powerful framework for building multimedia applications [2], which has adopted the idea of «video pipeline» from the Oregon Graduate Institute. This framework allows you to create applications of different levels of complexity, from simple console player (you can reproduce any file directly from the terminal without having to write any code), the full audio/video players, multimedia editors and other applications(Fig. 2).

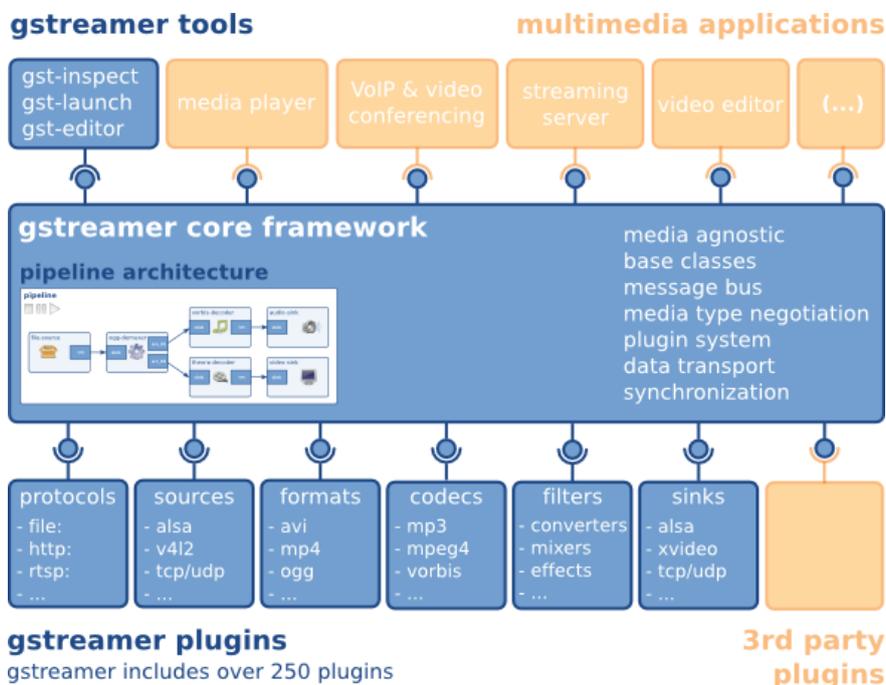


Figure 2. GStreamer framework

Elements. An element is the most important class of objects in GStreamer. You will usually create a chain of elements linked together and let data flow through this chain of elements. An element has one specific function, which can be the reading of data from a file, decoding of this data or outputting this data to your sound card (or anything else). By chaining together several such elements, you create a pipeline that can do a specific task, for example media playback or capture. GStreamer ships with a large collection of elements by default, making the development of a large variety of media applications possible. If needed, you can also write new elements. That topic is explained in great deal in the GStreamer Plugin Writer's Guide.

Pads. Pads are element's input and output, where you can connect other elements. They are used to negotiate links and data flow between elements in GStreamer. A pad can be viewed as a «plug» or «port» on an element where links may be made with other elements, and through which data can flow to or from those elements. Pads have specific data handling capabilities: a pad can restrict the type of data that flows through it. Links are only allowed between two pads when the allowed data types of the two pads are compatible. Data types are negotiated between pads using a process called caps negotiation. Data types are described as a GstCaps. An analogy may be helpful here. A pad is similar to a plug or jack on a physical device. Consider, for example, a home theater system consisting of an amplifier, a DVD player, and a (silent) video projector. Linking the DVD player to the amplifier is allowed because both devices have audio jacks, and linking the projector to the DVD player is allowed because both devices have compatible video jacks. Links between the projector and the amplifier may not be made because the projector and amplifier have different types of jacks. Pads in GStreamer serve the same purpose as the jacks in the home theater system.

For the most part, all data in GStreamer flows one way through a link between elements. Data flows out of one element through one or more source pads, and elements accept incoming data through one or more sink pads. Source and sink elements have only source and sink pads, respectively. Data usually means buffers (described by the GstBuffer object) and events (described by the GstEvent object).

Bins and pipelines. A bin is a container for a collection of elements. Since bins are subclasses of elements themselves, you can mostly control a bin as if it were an element, thereby abstracting away a lot of complexity for your application. You can, for example change state on all elements in a bin by changing the state of that bin itself. Bins also forward bus messages from their contained children (such as error messages, tag messages or EOS messages).

A pipeline is a top-level bin. It provides a bus for the application and manages the synchronization for its children. As you set it to PAUSED or PLAYING state, data flow will start and media processing will take place. Once started, pipelines will run in a separate thread until you stop them or the end of the data stream is reached. *Viola-Jones approach* The Viola-Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection [3].

The problem to be solved is detection of faces in an image. A human can do this easily, but a computer needs precise instructions and constraints. To make the task more manageable, Viola-Jones requires full view frontal upright faces. Thus in order to be detected, the entire face must point towards the camera and should not be tilted to either side. While it seems these constraints could diminish the algorithm's utility somewhat, because the detection step is most often followed by a recognition step, in practice these limits on pose are quite acceptable.

The algorithm has four stages:

1. Haar Feature Selection.
2. Creating an Integral Image.
3. Adaboost Training.
4. Cascading Classifiers.

A face detector (Fig. 3) has to tell whether an image of arbitrary size contains a human face and if so, where it is. One natural framework for considering this problem is that of binary classification, in which a classifier is constructed to minimize the misclassification risk. Since no objective distribution can describe the actual prior probability for a given image to have a face, the algorithm must minimize both the false negative and false positive rates in order to achieve an acceptable performance. This task requires an accurate numerical description of what sets human faces apart from other objects. It turns out that these characteristics can be extracted with a remarkable committee learning algorithm called Adaboost, which relies on a committee of weak classifiers to form a strong one through a voting mechanism. A classifier is weak if, in general, it cannot meet a predefined classification target in error terms.

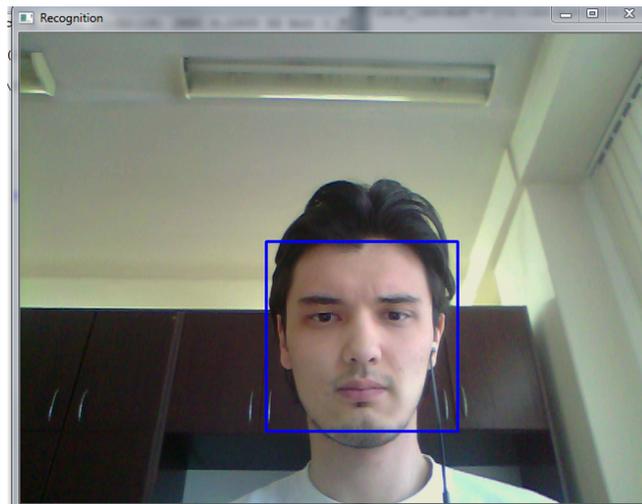


Figure 3. Face detection using Viola Jones method

An operational algorithm must also work with a reasonable computational budget. Techniques such as integral image and attentional cascade make the Viola-Jones algorithm highly efficient: fed with a real time image sequence generated from a standard webcam, it performs well on a standard PC.

For the implementation of the task, it was decided to use the Raspberry pi 3 (Fig. 4). The Raspberry Pi is a series of credit card-sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation [4] with the intent to promote the teaching of basic computer science in schools and developing countries. Board feature a Broadcom system on a chip (SOC), which includes an ARM compatible CPU and an on chip graphics processing unit GPU (a VideoCore IV). CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. Secure Digital SD cards are used to store the operating system and program memory in either the SDHC or MicroSDHC sizes. Most boards have between one and four USB slots, HDMI and composite video output, and a 3.5 mm phono jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I2C. Some models have an 8P8C Ethernet port and the Pi 3 has on board WiFi 802.11n and Bluetooth. The Raspberry pi software consists of a OS, environment (IDE) and the core libraries. The core libraries are written in C and C++ and compiled using avr-gcc and AVR Libc. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on the computer (e.g. Flash, Processing, MaxMSP.) The boards can be assembled by hand or purchased preassembled. The Raspberry pi Foundation support Debian and Arch Linux ARM distributions and promotes Python as the main programming language.



Figure 4. GStreamer framework

During the research working, for solving the physical question, mainly in hardware issues, we found out that we can use bunch of Arduino Uno [5] board and raspberry pi which can interact between the high(application) and low(physical) layers. All sensors which connected to arrduino via i2c bus communicatio with each other in one enclosed loop (Fig. 5).

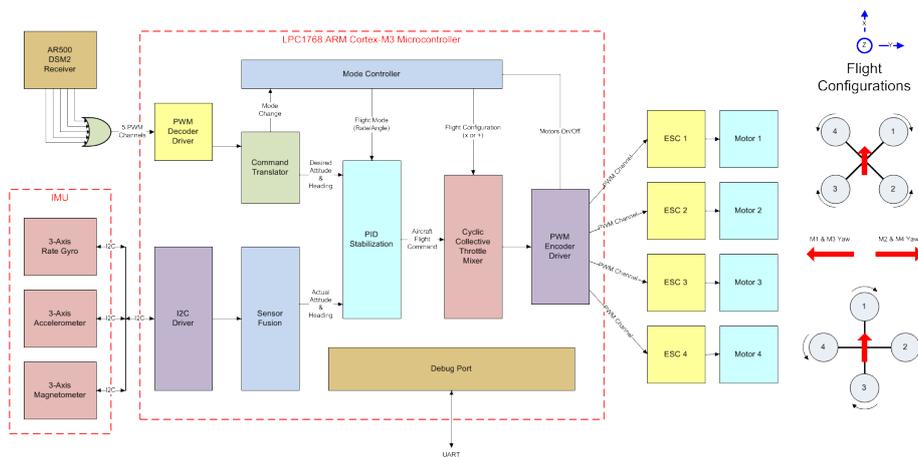


Figure 5. Loop

Peripheral device like a camera module is connected to raspberry pi 3 over the SPI interface. All of the electrical hardware components previously mentioned provide the necessary ground work for the software subsystems to allow the copter to function as desired. The software plays one of the most important roles controlling and facilitating the features of the device discussed in this paper. All of the inputs from the various sensors gyroscope, accelerometer and barometric sensor are to be taken and evaluated by the microcontroller to facilitate flight. The approach is divided into two main modules: One for Face detection module and Sending data to the ground station over the network WiFi (Fig. 6).

With radio control taking off the quadcopter, and try to get it stable as much as it possible, or activate position hold (using barometr and gps). Start the pipeline on the raspberry pi side by the following command using ssh:

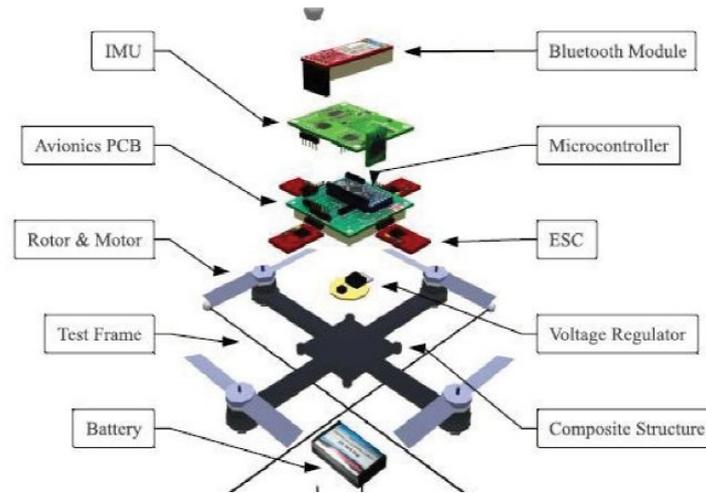


Figure 6. Quadcopter architecture

```
raspivid -n -w 640 -h 480 -t 0 -o - | gst-launch-1.0 -v fdsrc ! h264parse !
rtph264pay config-interval=10 pt=96 ! udpsink host=192.168.1.3 port=9000
```

Recieve the stream to ground station (Fig. 7):

```
gst-launch-1.0 -v udpsrc port=9000 caps='application/x-rtp, media=(string)video,
clock-rate=(int)90000, encoding-name=(string)H264' ! rtph264depay ! video/x-h264,
width=640,height=480,framerate=30/1 ! h264parse ! avdec_h264 ! videoconvert !
autovideosink sync=false
```

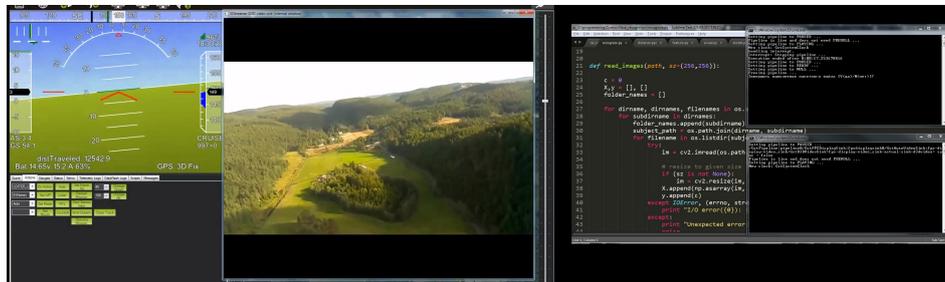


Figure 7. Video screenshot from UAV in the ground station.jpg

In this work has been achieved the following results: construct and modeled the prototype of UAV quadcopter, and implemented the GStreamer framework into raspberry pi board for recieving video data link into ground station. In prospective it gives wide spectre of work, in future it can be able to used for platform based on detection, recognition object, for rescue and area search, monitoring hard to reach places, data gathering and so on.

References

- 1 Nonami K., Kendoul F., Suzuki S., Wang W., Nakazawa D. Autonomous Flying Robots. — Tokyo: Springer, 2010.
- 2 GStreamer framework documentation. — [ER]. Access mode: [https: GStreamer.freedesktop.org](https://GStreamer.freedesktop.org).

- 3 Viola P., Jones M.J. Robust real – time face detection // International Journal of Computer Vision. — 2004. — Vol. 57. — No. 2. — P. 137–154.
- 4 Raspberry pi platform. — [ER]. Access mode: <https://www.raspberrypi.org>.
- 5 Arduino platform. — [ER]. Access mode: <http://arduino.cc>.

Н.А. Есмағамбет

Пилотсыз ұшу аппараты үшін түрді тану жүйесімен қатар GStreamer мультимедиалық фреймворкты қолдану

Мақала пилотсыз ұшу аппаратын (ПҰА) құру және ПҰА Pipeline арнасы бойынша GStreamer фреймворкасы арқылы OpenCV-да кейінгі өңделу үшін жер стансасына ПҰА-дан бет-пішінің анықтау мақсатында деректер базасын беру әдісі зерттелген. Жұмыс барысында «Raspberry Pi» және «Arduino» есептеу аппараттық платформа аясында C++, Python бағдарламау тілінде «ArduQuad, Arducopter, MultiWii» жобаларының Open Source кешенді алгоритмдері қолданылған. Сондай-ақ Wi-Fi желісі арқылы ПҰА-дан жер стансасынан «ішкі» бейнедеректердің «жолы» сараланған.

Н.А.Есмағамбет

Применение мультимедийного фреймворка GStreamer в связке с системой опознавания лиц для беспилотных летательных аппаратов

Статья посвящена исследованию и экспериментальной разработке беспилотного летательного аппарата (БПЛА) и методу, по которому видеопоток с БПЛА передается по каналу Pipeline, обрабатываемому с помощью фреймворка GStreamer для дальнейшей его обработки в OpenCV, который, в свою очередь, принимает на наземную станцию отработанный видеопоток для опознавания лица. В ходе работы был смоделирован беспилотник с полетным контролером на базе аппаратной вычислительной платформы «Raspberry Pi» и «Arduino» с использованием комплексных алгоритмов Open Source проектов «ArduQuad, Arducopter, MultiWii» на языке программирования C++, Python. Разработана «магистраль», по которой передается «сырой» поток видеоданных через сеть Wi-Fi с БПЛА на наземную станцию.