Color-Based Object Trackıng For Unmanned Aerıal Vehıcles

Anıl SEZGİN[1]
Sırma YAVUZ[2]

Abstract

Quadrotor has a huge advantage to classical unmanned aerial vehicles (UAV) since it performs vertical take-off and landing (VTOL) with high maneuverability, which is why it has been worked on by many researchers in recent years. Thanks to VTOL ability, it can be used in rugged and limited terrain without the need of long runways for take-off and landing. In this study, the structure and dynamics of UAVs will be examined and a software will be developed to control a Quadrotor. With this control station, the UAV will be remote controlled, the autonomous missions will be tracked and the vehicles status when it is airborne or out of sight will be monitored. Thus the UAV technology developed for military intentions will be able to be used in the search and rescue field.

Keywords: *drone, control, tracking, uav, autonomous*

1 Introductıon

In recent years, Unmanned Aerial Vehicles (UAV) has formed an important part of military and space technology research. These vehicles offer a chance to complete tasks without putting human life at risk. Compared to manned systems, these vehicles have huge advantages such as having no human casualty risks in military areas where the threat is intense and the performance of the vehicle not depending on human weakness. Therefore, many countries have research & development work in this area. The UAV definition on NATO sources is as follows: Vehicles, which have external operators, are remote controlled or have a motorized boost force to steer themselves autonomously, can attach or detach guns or other useful loads to their main bodies, can return and land after missions or self destruct at target as a weapon.

Easy VTOL on all terrains (VTOL, vertical take-off and landing) and the maneuverability are the reasons for us to prefer quadrotors as UAVs. With these 2 features they can be used indoor and outdoor and achieve a great advantage on spying and investigation missions. However maneuverability makes it harder to control these vehicles [1].

Quadrotor has a huge advantage to classical UAVs since it performs vertical take-off and landing (VTOL) with high maneuverability, which is why it has been worked on by many researchers in recent years. Thanks to VTOL ability, it can be used in rugged and limited terrain without the need of long runways for take-off and landing. Furthermore, quadrotor has a simpler mechanical structure compared to other VTOL unmanned aerial vehicles, but despite their advantages

[1] *Istanbul Esenyurt University, Department of Electrical & Electronics Engineering*
[2] *Yildiz Technical University, Department of Computer Engineering*

these vehicles have an unstable structure and a highly nonlinear and interrelated dynamics. For this reason, controlling a Quadrotor brings along its difficulties. In this study, the structure and dynamics of unmanned aerial vehicles will be examined and a ground control station software will be developed to control a quadrotor.

1.1 Related Work

The study by Nicoud and Zefferey analysis the wing, propeller and motor characteristics and propose a methodology to optimiz motor/gear/propeller system, the [2]. The study by Mejias et al. Addresses the task in which UAV is performing an inspection on a set of power lines and an emergency situation occurs requiring UAV to avoid the lines and then find a safe landing area [3]. The study by Zingg et al. presents an approach for wall collision avoidance using a depth map based on optical flow from on board camera images. An omnidirectional fisheye camera is used as a primary sensor, while IMU data is needed for compensating rotational effects of the optical flow [4]. Tournier et al.present the vision-based estimation and control of a quadrotor vehicle using a single camera relative to a novel target that incorporates the use of moire patterns. A target contains markers to determine its relative orientation and locate two sets of orthogonal moire patterns at two different frequencies [5].

2. Drone Structure

Quadrotor is an air vehicle with 4 motors. 2 of these motors rotate clockwise, while the other 2 rotate counter-clockwise in order to balance the torque created by the impellers. It is possible to maneuver in the space by the rotation of the motors on different speed settings. The impeller helixes of the neighboring motors must be different, so that the neighboring impellers can rotate in opposite directions to create a perpendicular force against gravity.

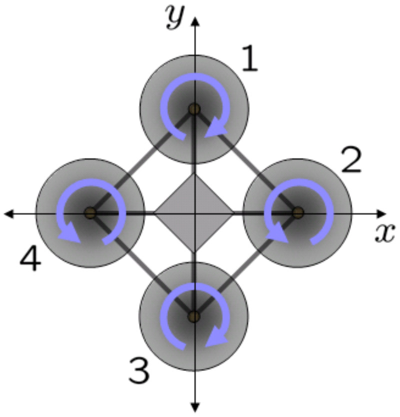

Figure 1: *Rotation direction of the motors*

Although these air vehicles seem to have 6 axis sensors (6 Degreed of Freedom), they have 4. The reson for this is, the advance on the y axis while rotating around the x axis and the advance on the x axis while rotating on the y axis are not independent from each other. In this case, the movements are rotation around x, y, z axis and ascension on the z axis.

Drones are unmanned vehicles which can be autonomously or remote controlled. There are different models for land, air and sea. UAVs can be designed to accomplish various tasks. Quadrotor is an example of small UAVs, these vehicles are mostly used for air photography.

2.1 Fixed-Wing vs Rotary Wing

UAVs generally come in 2 forms: Fixed-Wing and Rotary Wing. The Fixed-Wings can travel much more distance and make much more speed. However, these kinds of vehicles require a rather long track for take-off and landing. Rotary Wing UAVs, on the other hand, can vertically take-off and land and have a higher level of maneuverability. Therefore a drone must be chosen according to the task at hand.

One of the most important features of the Rotary Wing vehicles is they can stay remain stable. Thus, the target could be approached easily and photographs could be taken. These vehicles can maintain their audio and video recording peripheral devices or sensors running even after landing and killing the engines. This process is called perch-and-stare [6]. While carrying a cargo with these vehicles, the cargo can be easily delivered to the desired place. Such things are impossible with Fixed-Wings.

The popularity of these kinds of vehicles has encouraged new developments on various fields such as battery, wireless communication and solid state equipment. These low-budget UAVs have become a part of many military and civilian practices. Low-budget UAVs have begun to be used on various fields such as data gathering, natural event tracking and surveillance for security. Working on the design of UAVs, researchers are constantly in the search for solutions to problems such as the weight of the flight platforms and the energy usage of the components.

2.2 Drone Features

The features that should be taken into consideration while choosing a drone are as follows: Weight, durability, distance, altitude, wing load and motor type.

2.2.1 Weight

UAVs have a broad scale of weight. Vehicles that weigh 400 grams exist as well as RQ-4 Global Hawk, which weighs 10 tons.

2.2.2 Durability and Distance

It is important to categorize UAVs in durability and distance as well as according to various tasks. While calculating drone distance, recharge and bunkering frequency are taken into consideration. Most drones require landing to recharge or bunkering which affects operation time. Bigger drones such a Global Hawk can bunker without landing, however even this affects operation time. While having a lesser level of durability, the AR drone can stay airborne for approx. 20 minutes. This duration is enough for short-distance tasks. Vehicles with higher durability levels can stay airborne for more than 24 hours and can travel approx. 22.000 kilometers.

2.2.3 Altitude

The reason this feature is important for military drone choices is because the vehicle would be hard to sight while on a high altitude. This prevents the vehicle from being noticed and destroyed. Furthermore altitude is vital for photography - the higher the vehicle is the bigger the photo of the land. Vehicles with low altitudes such as the AR drone can go up to a height of 100 meters. Drones with higher altitude ability such as Darkstar and Predator B can go up to a height of 45.000 ft.

2.2.4 Wing Load

Wing load is defined as the vehicle weight/wing area ratio. With this ratio, how much weight the UAV could carry and the velocity it needs to carry its load are calculated. The higher the velocity of the UAV, the more drag on the wing unit area. This way, small UAVs can carry weights almost as much as their own in high velocity. However, overweight constrains the take-off and landing in high velocity and reduces maneuverability [7].

2.2.5 Motor Type

Unmanned aerial vehicles can also be categorized according to their motor types. The weight of the vehicle is determined by the motor type - the bigger the vehicle the more force it requires for take-off. Small UAVs use mostly electric motors, whereas industrial vehicles use piston motors. Furthermore, motor types affect distance and durability of the drones.

2.3 Drone Choice

By taking a look at the aforementioned features, the drone for this study should be lightweight, should be able to stay airborne for 20 minutes to be able to travel the required distance, should be able to go up to a height of 20 meters, should be able to do VTOL and should have cameras for video recording. When these features are taken into considertain, the drone to be used is the AR Drone by the Parrot Company.

2.3.1 AR.Drone

The Parrot AR Drone is a micro UAV developed for the video gaming industry. Released on 2010,

the drone is used both on military and civilian tasks. The drone is also used for academic tasks since it is lightweight, low-cost and maneuverable. Top 3 reasons for this drone to be preferred for this study are: Tenacity, possibility to modify and being low-cost compared to its peers. AR Drone is the first 4 impeller helicopter to be remote controlled with various mobile devices or devices with a Wi-Fi interface. There are sensors, front and bottom camera and an ultrasonic-type height meter on top of the drone. This drone is operated with the commands sent from a Wi-Fi connection
.

Figure 2: *AR.Drone 2.0*

AR Drone is a vehicle developed by Parrot SA. It has an interior casing improved with Expanded Polypropylene (EPP) in order to protect the impellers. This makes it a durable, lightweight and recyclable piece. The impellers are operated by 4 brushless motors (28500 RPM and 14,5W). The vehicle has a 1500mAh Li-po battery, which enables the vehicle an approx. 10 minutes of flight time.

The AR Drone is equipped with a computer with a 1GHz 32bit ARM Cortex A8 processor and 1GB DDR2 Ram. This computer uses the Linux operating system. It is possible to include new improvements to the system such as changes to the software and new hardware like GPS sensor thanks to the USB connections on the device. The AR Drone hosts 6 axis inertial sensors (6-DOF, degrees of freedom), thus the control of the vehicle is maintained by pitch, yaw and roll. The measurement system consists of 3 axis accelerometer, 2 axis roll and pitch gyrometer and 1 axis yaw gyrometer. As an accelerometer, a BMA150 developed by Bosh Sensortech is used.

This air vehicle is made of carbon fiber and strong PA66 plastic. With the help of MEMS (Micro-Electro-Mechanical System) and video rendering, heuristic piloting of a remote controlled object is made possible. By video transmission to all devices with a Wi-Fi interface and using image processing software, the device is operated as desired. The impellers of this vehicle are custom made and the vehicle has a carbon fiber tube body.

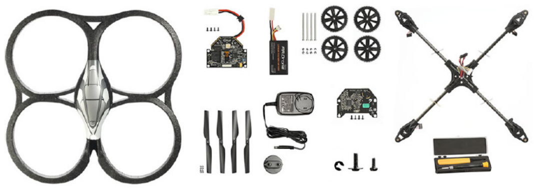

Figure 2: *Parts of AR.Drone 2.0*

2.3.2 Motor

AR.Drone uses brushless engines that are controlled by a microcontroller to power the rotors. Drone can detect whether all four engines are working and if any have stopped. Here is intended to prevent repeated shocks to the engine if the propeller encounters and obstruction while rotating [8].

The motor has a power of 15 Watts, completes 28,000 RPM when hovering and corresponds to 3,300 RPM on the propeller. The motors' ranges begin at 10,350 RPM and goes up to 41,400 RPM. The motor is attached to its electronic controller which has been specially designed for AR.Drone 2.0. An 8-bit low power microcontroller and a 10-bit ADC control the speed of the motor [8].

2.3.3 Battery

AR.Drone 2.0 uses a charged of 1500mAh, 11.1V Li-Po battery to fly. Battery's charge is determined when at 12.5 and low at 9V full (100% if battery is full, 0% if battery is low). Drone monitors the voltage of the battery and displays it to user as a percentage so that flight decisions can be made accordingly. When drone detects low battery voltage, it first sends a warning message to the user, then automatically lands. If the voltage reaches a critical level, the whole system is shut down to prevent any unexpected behavior [8].

2.3.4 Sensor

AR.Drone 2.0 has many motion sensors which are located below the centrall hull. First version of AR.Drone features a 6-DOF, MEMs-based, miniaturized inertial measurement unit that provides the software with pitch, roll and yaw measurements. Inertial measurements are used for automatic pitch, roll and yaw stabilization and assisted tilting control. Ultrasound telemeter provides with altitude measure for automatic altitude stabilization and assisted vertical speed control. Camera aiming towards the ground provides with ground speed measure for automatic hovering and trimming [8]. Measurement unit contains 3-axis accelerometer, a 2-axis roll and pitch gyrometer and a single yaw gyrometer [9]. AR.Drone has 2 cameras and a main board which holds two processors. One of these processors is used to gather data from I/Os. The other one is used to run all the algorithms that drone needs to maintain flight stability and process images [10].

2.3.5 Main Board

The main board includes a 1GHz ARM Cortex A8 central CPU with 8GHz video DSP, which runs Linux operating system. It includes 1Gb 200MHz DDR2 RAM and has a Wi-Fi chipset from Atheros and USB port for direct flashing. Main board is equipped with a pressure sensor and vertical on board camera operating at 60 FPS. The pressure sensor and vertical camera are used in combination with the feedback from the navigation board to provide absolute stability [10].

2.3.6 Camera

There are 2 cameras on AR.Drone, the frontal camera is a CMOS sensor with a 90 degrees angle lens. AR.Drone automatically encodes and streams the incoming images to the host device. AR.Drone 1.0 uses QCIF as bottom camera image resolution and QVGA as frontal camera image resolution. Video stream frame rate is set to 15 FPS. AR.Drone 2.0 uses 640x360 or 1280x720 image resolutions for both cameras, the video stream frame rate can be adjusted between 15 and 30 FPS [8].

2.3.7 Embedded Software

AR.Drone is based on Linux kernel, the operating system simultaneously manages threads pertaining to: Wi-Fi communications, video data sampling, video compression for wireless transmission, image processing, sensors acquisition, state estimation and closed-loop control [10].

3. Communication With Drone

The software which is developed on this study, based on application programming interface (API). The AR.Drone has an open source API that is used as a research standard for developing applications. API includes a software development kit (SDK) which has been written in C and runs on Linux, iOS and Android platforms.

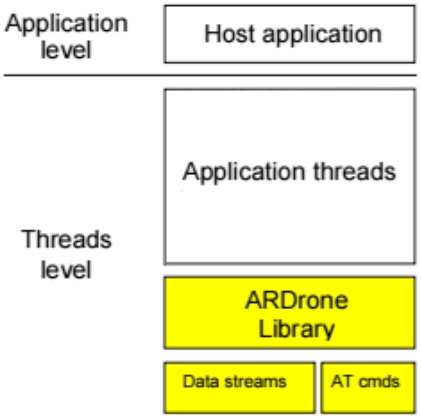

Figure 3: *Layered architecture of AR.Drone*

3.1 AT Command Protocol

AT command protocol is used to control the AR.Drone 2 over wifi. To be able to send AT commands to the AR.Drone 2 you have to connect to the drone. AT commands are encoded as 8-bit ASCII characters with a carriage return "<CR>" as a newline delimeter. All AT commands start with "AT" followed by a command name, sequence number and optionally a list of comma-separate arguments for the command.

- AT*REF (input) - Takeoff/Landing/ Emergency Stop command
- AT*PCMD (flag, roll, pitch, gaz, yaw) - Move the drone
- AT*FTRIM - Sets the reference for the horizontal

plane (The drone must be on the ground)

- AT*CONFIG (key, value) - Configuration of drone

3.2 AR.Drone Autonomy

Ardrone_autonomy is a ROS driver for AR.Drone Quadrotor. This driver is based on official AR.Drone SDK. Information received from the drone is published to the ardrone/navdata topic. Message contains the following information [11]:

- header: ROS message header
- batteryPercent: The remaining charge of drone battery
- state: Drone’s current state

1. Inited
2. Landed
3. Flying
4. Hovering
5. Test
6. Taking off
7. Landing

- rotX: Rotation about X axis (Left/right tilt in degrees)
- rotY: Rotation about Y axis (Forward/backward tilt in degrees)
- rotZ: Rotation about Z axis
- magX, magY, magZ: Magnetomter readings
- pressure: Pressure sensed by Drone’s barometer
- temp: Temperature sensed by Drone’s sensor
- wind_speed: Estimated wind speed
- wind_angle: Estimated wind angle
- altd: Estimated altitude (mm)
- vx, vy, vz: Linear velocity (mm/s)

3.2.1 Terminal Control with AR.Drone Autonomy

After installing AR.Drone drivers successfully, following commands are used to control AR.Drone through Linux OS terminal [12].

Commands for basic movements:

- Takeoff

```
rostopic pub -1 /ardrone/takeoff std_msgs/Empty
```

- Land

```
rostopic pub -1 /ardrone/land std_msgs/Empty
```

- Switch Camera

```
rosservice call /ardrone/togglecam
```

- Forward

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: 1.0, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}’
```

- Backward

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: -1.0, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}’
```

- Left

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: 0.0, y: 1.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}’
```

- Right

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: 0.0, y: -1.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}’
```

- Up

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: 0.0, y: 0.0, z: 1.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}’
```

- Down

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: 0.0, y: 0.0, z: -1.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}’
```

- Clockwise Rotation

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: -1.0}}’
```

- Counterclockwise Rotation

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 1.0}}’
```

- Stop

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist ‘{linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}’
```

- Camera

```
rosrun image_view image_view image:=/ardrone/image_raw
```

- Front Camera

```
rosrun image_view image_view image:=/ardrone/front/image_raw
```

- Bottom Camera

```
rosrun image_view image_view image:=/ardrone/bottom/image_raw
```

- Height Sensor

```
rostopic echo /sonar_height
```

- Navigation Info

```
rostopic echo /ardrone/navdata
```

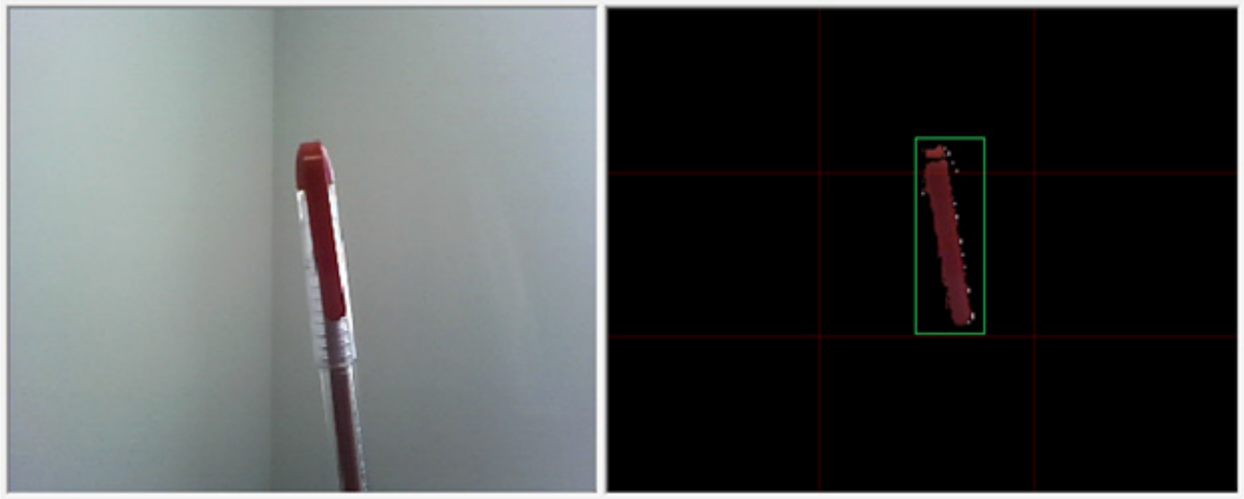

Figure 4: *RGB to HSV*

4 Software

In this study, a script is developed to control AR.Drone with a keyboard and autonomous control with color detection and tracking. When script runs drone takes off and both camera images are also checked. If there is no object to track on both camera images, then drone starts to look for the object to be followed by turning around itself. If desired, this autonomous operation can be terminated, and user can control drone via keyboard.

4.1 Color-Based Object Tracking Software

Object detection and tracking is the most important and challenging fundamental task of computer vision. Tracking objects based on color is one of the quickest methods for tracking an object. The speed of this technique makes it very useful for applications. RGB (Red, Green, Blue) color space is a kind of color spaces which uses red, green and blue to elaborate color model. HSV (Hue, Saturation, Value) is color model that describes colors in terms of their shade (saturation). Hue of a color refers to which pure color it resembles, it is described by a number that specifies the position of the corresponding pure color on the color wheel as a fraction between 0 and 1. Value 0 refers to red, 1/6 refers to yellow, 1/3 refers to green. Saturation of color describes how white the color is. A pure red is fully saturated with a saturation of 1, tints of red have saturations less than 1 and white has a saturation of 0. Value of a color describes how dark the color is, a value of 0 is black [13]. The most common color space is RGB but it is flawed for color tracking because changes in color that look small to the eye effect all three channels significantly. HSV is more useful color space for tracking.

In this study, a red-colored object is used to be detected and tracked by drone. 2 different cameras on drone can be used for detection.

Firstly, we capture video feed from device and convert it to HSV color space to detect red-colored test object. After get the components in this color range, draw a minimum-area bounding rectangle on HSV frame.

4.2 Control Software

Drone will takeoff, land or emergency stop with publishing these messages:
ardrone/takeoff
ardrone/land
ardrone/reset
After takeoff, drone is controlled by publishing geometry_msgs::Twist to cmd_vel topic:

- -linear.x: Move backward
- +linear.x: Move forward
- -linear.y: Move right
- +linear.y: Move left
- -linear.z: Move down
- +linear.z: Move up
- -angular.z: Turn left
- +angular.z: Turn right

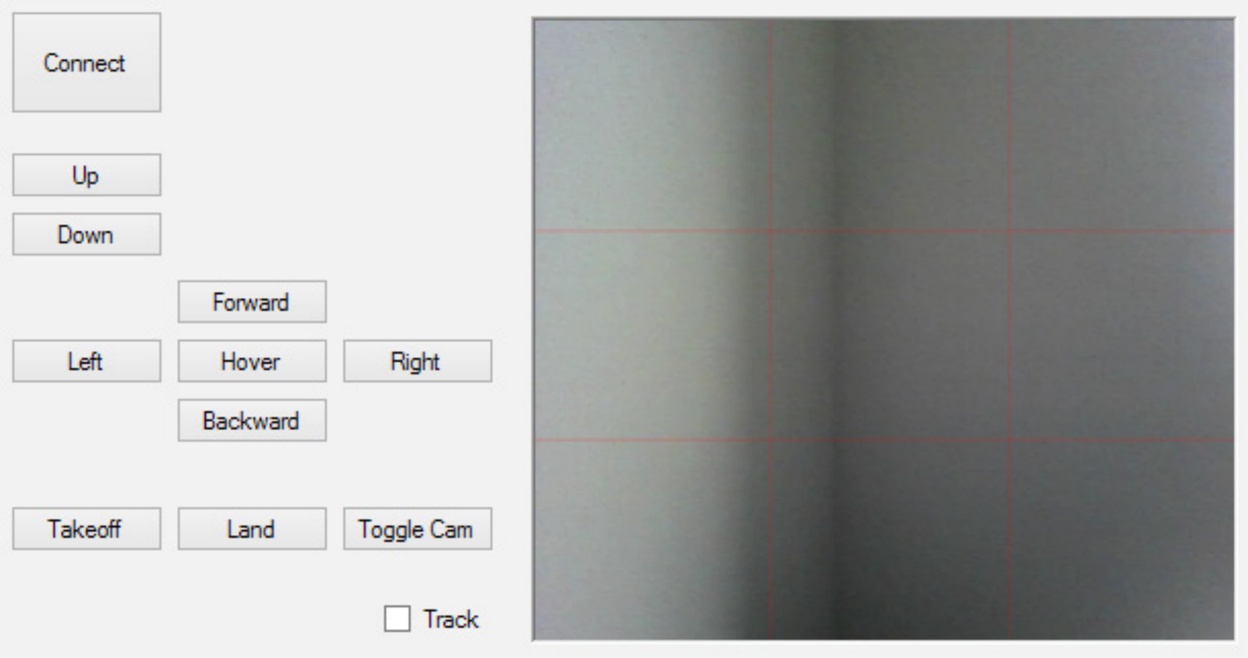

Figure 5: *Control panel*

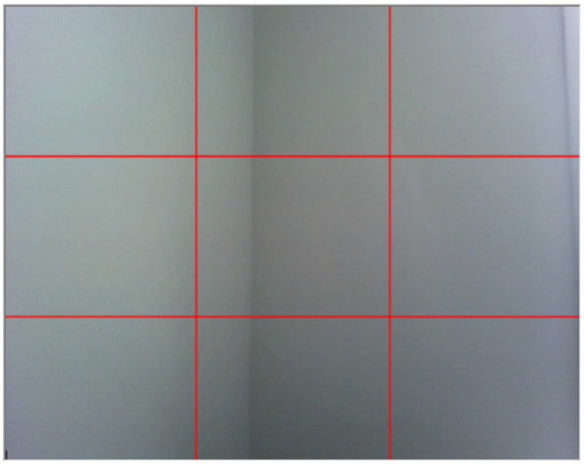

Figure 6: *Grid lines on video feed*

We receive images from the camera as shown in the figure above and divide them into 9 parts with grid lines. We repeat this process for both camera images which are received from drone. The location of rectangle of red-colored object on the screen which is detected in video feed, is identified. After identifying the location, script checks which part of screen, rectangle is on. With this part number, script decides the movement of drone. There are 2 cameras on drone, by using front camera video feed drone moves vertically (forward, left, right) and horizontally (up, down). By using bottom camera video feed, drone moves vertically (forward, backward, left, right).

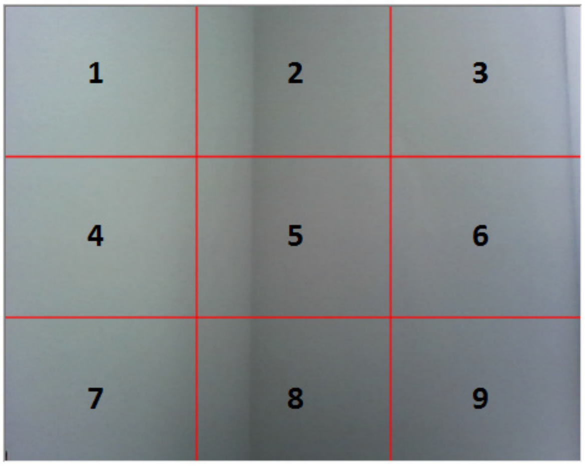

Figure 7: *Part numbers of video feed*

If the rectangle around the detected object which is drawn on HSV color space, is on part 2, software commands the drone to move up, if the rectangle is on part 8, software commands the drone to move down, if the rectangle is on part 4, software commands the drone to move left and if rectangle is on part 6, then software commands the drone to move right. For bigger objects which can occupy almost same area as one part of video feed images, script calculates the area of rectangle by using length of sides, if the area of rectangle in 1 part is greater than the half of the rectangle area then software commands the drone to move to keep the rectangle in the middle of the screen which means part 5.

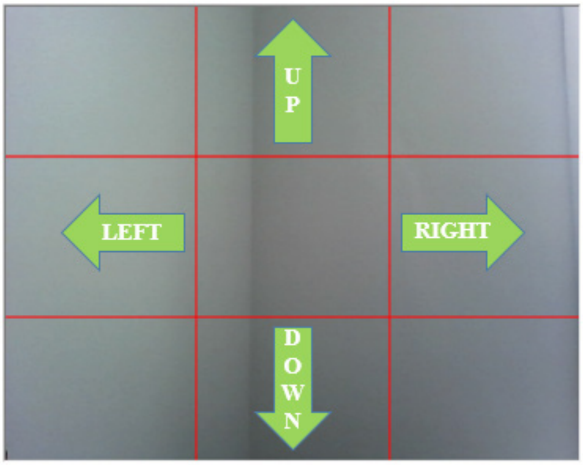

Figure 8: *Directions by using front camera*

On bottom camera video feed the same process is repeated to keep the rectangle in part 5 but this time if greater area is on part 2, software commands the drone to move forward, on part 8 software commands the drone to move backward, on part 4 software commands the drone to move left and on part 6 software commands the drone to move right.

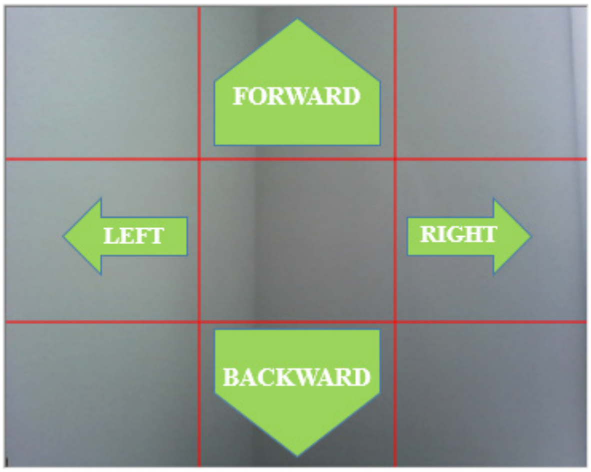

Figure 9: *Directions by using bottom camera*

Conclusion

The system which is developed in this study, performs the autonomous functions of target detection and tracking using the AR.Drone. AR.Drone is the most suitable aerial vehicle to implement this systems. A few challenges were encountered when implementing this system. Drone's components slightly damaged towards the end of testing which caused drone to move unpredictable at times. The system is successfully implemented, AR.Drone is capable of performing color detection and tracking during flight.

REFERENCES

[1] J. -D. Nicoud ve J. -C. Zufferey, "Toward indoor flying robots", IEEE/RSJ International Conference on Intelligent Robots and Systems (Volume:1), 2002.

[2] L. Mejias, P. Campoy, K. Usher, J. Roberts ve P. Corke, "Two Seconds to Touchdown - Vision-Based Controlled Forced Landing", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.

[3] S. Zingg, D. Scaramuzza, S. Weiss ve R. Siegwart, "MAV navigation

through indoor corridors using optical flow", IEEE International Conference on Robotics and Automation (ICRA), 2010.

[4] G. P. Tournier, M. Valenti, J. P. How ve E. Feron, "Estimation and Control of a Quadrotor Vehicle Using Monocular Vision and Moire Patterns", Navigation and Control Conference and Exhibit, 2006.

[5] T. W. Danko, A. Kellas ve P. Y. Oh, "Robotic rotorcraft and perch-and-stare: sensing landing zones and handling obscurants", Proceedings of the 12th International Conference on Advanced Robotics., 2005.

[6] S. J. Morris, "Design and flight test results for micro-sized fixed-wing and VTOL aircraft", Proceedings of the First International Conference for Emerging Technologies of Micro Air Vehicles, Atlanta, 1997.

[7] S. Piskorski, N. Brulez, P. Eline ve F. D'Haeyer, "AR Drone Developer Guide SDK 2.0", Parrot, 2012.

[8] N. Dijkshoorn, "Simultaneous Localization and Mapping with AR Drone", Universiteit van Amsterdam, 2012.

[9] P.-J. Bristeau, F. Callou, D. Vissière ve N. Petit, "The navigation and control technology inside the AR.Drone micro UAV", 18th IFAC World Congress, Milano, 2011.

[10] M. Monajjemi, "ARDrone Autonomy", 23 04 2015. Available: http://wiki.ros.org/ardrone_autonomy.

[11] H. Huang ve J. Sturm, "tum_simulator Package", 12 05 2014. Available: http://wiki.ros.org/tum_simulator.

[12] J. W. Shipman, "HSV Color Model", 16 10 2012. Available: http://www.nmt.edu/tcc/help/pubs/colortheory/web/hsv.html.

[13] E. Altuğ, "Vision based control of unmanned aerial vehicles with applications to an autonomous quadrotor, PhD thesis", University of Pennsylvania, 2003.

[14] J. Leishman, "Principles of Helicopter Aeordynamics", Cambridge University Press, 2000.