Computer Vision In Remotely Piloted Aircraft To Avoid Collisions During Flight

Ashwini Kumar[1], Remya Sivan[2], Pallavi N[3]
[1, 2, 3] Dept of Computer Science and Engineering
[1, 2, 3] Atria Institute of Technology

Abstract- *This paper is about the usage of a calculation dependent on PC vision strategies that permit a Remotely Piloted Aircraft (RPA), or robot, to distinguish obstructions before it and rapidly choose to move so as to keep away from them, without utilizing any top of the line complex programming framework/programming utilizing just an essential camera and applying computer vision techniques on images ,consequently maintain a strategic distance from impact , thereby avoiding the collision. We have implemented and test it in a simulated environment, although it can be applied to RPAin day to day life as well.*

Keywords- Camera , Computer Vision , Drones , Obstacles Avoidance , RPA.

I. INTRODUCTION

Machine vision is the technology and methods used to provide imaging-based automatic inspection and analysis for such applications as automatic inspection, process control, and robot guidance, usually in industry. Machine vision is a term encompassing a large number of technologies, software and hardware products, integrated systems, actions, methods and expertise. Machine vision as a systems engineering discipline can be considered distinct from computer vision, a form of computer science. It attempts to integrate existing technologies in new ways and apply them to solve real world problems[1].Computer vision is the science that aims to give a similar, if not better, capability to a machine or computer. Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding[2].Processing images in real time is critical for decision making during the flight and can be done with a simple camera.

Another technology that is gaining great popularity known as drones.Since the innovation to manufacture them today is reasonable and surely known, they are now being utilized in numerous examines and in numerous applications. Among the proposed applications in the common territory are fire checking and quenching, assessment of extensions and structures, crop tidying and even hunt and salvage of survivors after a fiasco. In the military territory the applications are reconnaissance, protection and even air strikes.Low-cost arrangements, utilizing rambles, to these sorts of issues are exceptionally intriguing to be utilized in some vast scale worldwide tasks, influencing the innovation to turn out to be famous . For instance, there is an application called Dronestagram [3] that in Walk 2017, in association with "National Geographic", advanced the Automaton Photography Grant, the primary global challenge committed to this kind of ethereal photography. Ahead of all comers was Indonesian narrative movie producer Dendi Pratama, 29, who recorded a nearby of a falcon flying over Bali Barat Park in his home country[4].

There are numerous approaches to influence a drone or robot to see snags in their way and stay away from them, for example, utilizing sonars, GPS (global position system), recently known courses, and so forth. In this work we utilize just computer vision through the treatment of pictures gathered progressively from a trifling installed camera. The upside of this arrangement is to be basic and to have a minimal effort, turned out to be to be effective in the tests performed of reenacted flight

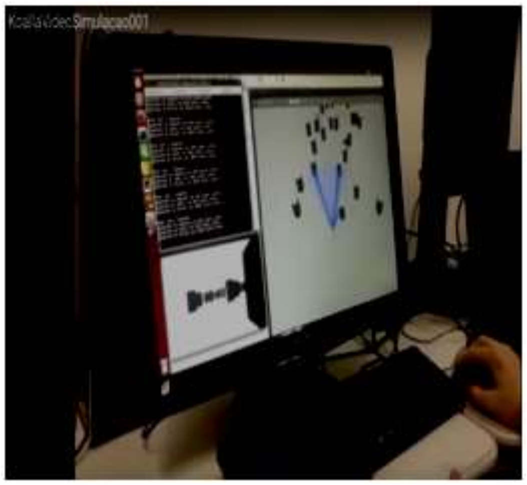

Fig.1. The Simulation

Miranda[6] and Sbeghen[5] presented this idea, us-ing monovision to manage a robot to maintain a strategic distance from deterrents. Sbeghen composed a C++ code in Borland C[7] in the Windows OS to discover the region in a photograph with less impediments, utilizing just twelve known jpg pictures documents recently spared in the hard plate. A key distinction in our work is that the pictures are caught progressively amid the automaton flight, giving a prepared to-run arrangement.

The important contrasts between the two methodologies are:

Items	Sbeghen's work	This work
Apply to	Robots	Drone
Scenario	Ground	Sky
Simulation	Manipulated	Gazebo
Simulation	Inflexible	Flexible
Images	Only 12	Unlimited
Get images	Files/Hard disk	Capture/Real time
Images	Known	Unknown
View	Static	Dynamic
Interface	No	Raspberry pi
Ready to real case	No	Yes
Adaptability	Hard	Easy
Movement	Discreet	Linear
OS	WINDOWS	LINUX

This paper centers around the calculation made and its joining with the recreated environment[9][10] (fig 1). The stage considered is a quad rotor smaller scale ethereal vehicle (MAV) which is like a helicopter, yet has four rotors[11]. The quad rotor stage is engaging a result of its mechanical effortlessness, spryness, and surely knew dynamics[12]. The remainder of the paper is organized as pursues. In Section II, we present the assets, equipment, programming and we detail the method utilized. Segment III subtleties the reproduction of the model and the instruments utilized in the tests. Area IV appears and examines the test results got. Area V introduces the finish of the work and an assessment to its immaterialness.

II. MATERIALS AND METHODS

We consolidate equipment and programming to gain pictures from a camera, treat and translate them and choose which way is obvious from obstructions, this progressively amid execution. The improvement of the calculation was made in two stages. In the first place, the code was created and tried in a scratch pad with a web camera, in which the choices of hindrance evasion were yielded through expressions, for example, "move to right side", "move to left side", "move ahead" or "quit", contingent upon the situation of the impediment experienced. To make snags, we begun by placing objects before the camera and after that individuals. We likewise strolled with the journal and web camera through the college passageways, accepting from the calculation guidelines on the best way to dodge understudies who came to us or were come to by us amid the walk. In the second step we incorporated the code in a simulated environment, and tried with realistic drone dynamics.

A. HARDWARE

Amid improvement we utilized a note pad with intel (R) pentium (R) CPU N3700 @ 1.60GHz 1.84GHz and 4.00GB RAM and a 1.3mp PISC webcam. The recreation was performed on a work area with intel Core i7-4770 CPU 3.40GHzx8, illustrations Gallium 0.4 on NVE4.

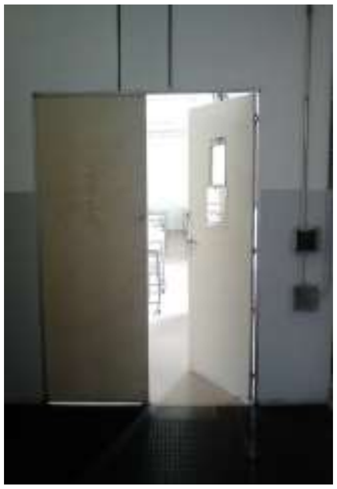
Fig. 2. Primitive Photo

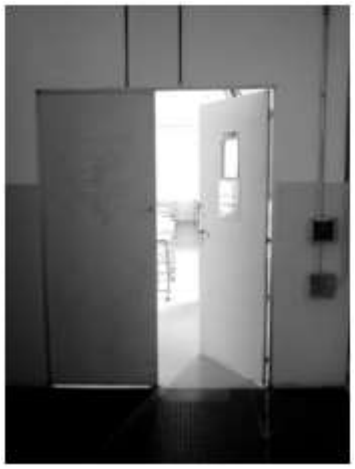
Fig. 3. Primitive photo converted into gray scales photo

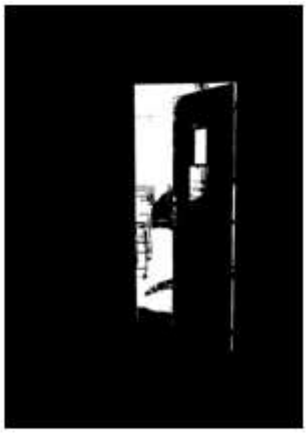

Fig.4. Thresholding

B. SOFTWARE

For improvement we utilized Linux working framework (OS) Ubuntu Desktop 14.04 LTS. The code was written in C++ language utilizing OpenCV 2.8 open source graphics library. The recreation was performed on a similar OS, with ROS Indigo (Robot Operation System) firmware and Gazebo test system 8.0.0.

C. RPA PLATFORM

The proposed calculation is planned to be kept running on a 450mm quadrotor utilizing a Pixhawk[13] as the principle controller board. The Pixhawk is an open-source flight controller load up in charge of the low-level control of the quadrotor. It is outfitted with whirligig, accelerometers, magnetometer and indicator, and can likewise be associated with an outside GPS module, and has a ground-breaking implanted programming that executes the essential control capacities. A valuable component of the Pixhawk is the capacity to speak with different gadgets through a convention called MAVLink[14], which was grown explicitly for RPA applications. This can be utilized to accomplish self-sufficient control of the RPA, by running the control calculations in a little versatile PC, for example, a Raspberry Pi, which is conveyed by the RPA and sends directions to the Pixhawk by MAVLink messages. The RPA is additionally furnished with a web camera, mounted on its front, to catch the pictures of the way it is moving into.

D. TECHNIQUE

The technique consists of (i) Capturing the picture of the front camera, outline by edge, (ii) Applying dark scale to the caught picture, (iii) Thresholding, (iv) Vertically partitioning the photograph into 03 regions of a similar tallness and width, called left, centre and right, (v) Counting the measure of dark pixels of every region, (vi) Suggesting that the UAV moves to the region with minimal measure of dark pixels, (vii) Discarding the handled picture and supplanting it by another edge, presently with the new situating of the UAV, rehashing the means from (i) until (vii) once more.

i)Capturing outline by casing the picture of the front camera: The UAV has an inserted camera situated before it that is actuated by the control program to catch a picture of the earth before it. Fig. 2 demonstrates a case of an untreated crude photograph (RGB) (red, green, blue) with a goals of 360x640 pixels,

ii)Apply Gray Scale to the caught picture: The principal scientific treatment that the crude picture gets is to have its number of hues diminished to a smaller band, restricted to tones or degrees or sizes of dim. This strategy treats the separation (D) between RGB hues in space, given by equation 1. The littler the separation between two hues, the more comparative they are[15]. In the RGB shading design, Color1: R1, G1, B1 and Color2: R2, G2, B2.

$$D= \quad \text{square root}((R1-R2)2+(tt1-tt2)2+(B1-B2)^2) \quad (1)$$

The degree of luminosity (L) of each pixel is also considered, which indicates how much the human eye can perceive this color[15].

$$L=R*0.2126+\mathrm{G}*0.7152+B*0.072(2)$$

Algorithm 1 shows the pseudo-code of theconversion[15].

```
for Each pixel do
Calculates its luminosity(L);
Paint the pixel with the brightness found;
End
```

Algorithm 1

We can likewise change over any shading to its rough dim dimension, acquire its red, green, and blue natives (from the RGB scale). 30% of red in addition to 59% of green in addition to 11% of blue is included, paying little heed to the scale utilized (0.0 to 1.0, 0 to 255, 0% to 100%.) The subsequent dimension is the ideal dark esteem. Such rates are identified with the genuine visual affect-ability of the customary human eye for the essential colors[16].

Algorithm 2 demonstrates the pseudo-code to this task.

```
for Each pixel do

PixelRGB[0]= 1.30*Pixel RGB[0] //Red;
Pixel RGB[1]= 1.59*Pixel RGB[1] //Green; Pixel RGB[2]=
1.11*Pixel RGB[2] //Blue;
```

End

Algorithm 2

We utilized in our code the capacity "cvtColor" of OpenCV open source realistic library to change over the crude photograph (Fig. 2) to a dark scale photograph (Fig. 3). In the dim scale, the hues R, G and B have a similar incentive in every pixel. We utilized this OpenCV work in our work on the grounds that later on we mean to offer our code to the store of this library and it would not bode well to make a second capacity to play out a similar picture handling effectively existing in that library.

iii)Thresholding: The second and last treatment that the picture caught by the camera gets is to have every one of its pixels turned on (255) or killed (0). We embraced a sift old for monochromatic gadgets, which could be for instance the normal pixel glow of the crude image[15]. For our situation, by experimentation, we picked edge 127. We dole out to every one of the 3 RGB shades of every pixel of the picture in dark scales, the qualities 0 (thoroughly dark) or 255 (absolutely white), if the estimation of RGB shading is individually not exactly or equivalent to the limit or more prominent than the edge. This is proportionate to stating that we painted in dark the darkest dim and the most brilliant white pixels. After this treatment we acquire the outrageous photograph (Fig. 4) that has just the hues dark (0) or white (255). The pseudo-code for this treatment is appeared in **Algorithm3(Generate extreme photo)---**

```
for Each pixel do
if luminosity(L) >= threshold then
Turn on the Pixel - set RGB(255,255,255);
    else
Turn off the Pixel - set RGB(0,0,0);
    end
    end
```

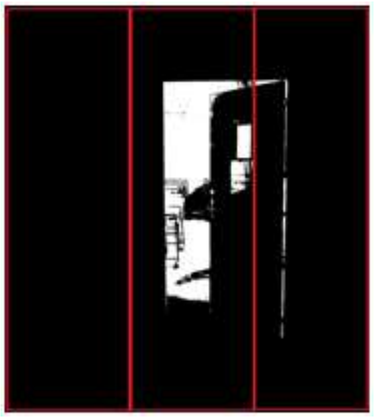
Fig 5. photo divide into 3 areas

iv)Divide the photograph vertically into 03 zones of a similar stature and width, called left focus and appropriate: After this we get the Extreme Photo, we isolate it vertically into three regions of a similar size. The width (t) of every zone is 33% of the complete width of the outrageous photograph.

$$t=div(width/3) \qquad (3)$$

Div is the whole part of the division. In our case, t = div (640/3) = div (213,333 ..) = 213.

In this way, we define the three areas (Fig. 5) as:

Area(0): 360 x (000 to 213) pixels (left side)
Area(1):360x(214to426)pixels(thecenter)
Area(2):360x(427to640)pixels(rightside)

IF	ACTION
A(n)> $p*Tn/100$	stop

v)Count the measure of dark pixels in every territory: After characterizing the zones, we tally the quantity of "off" focuses in every zone, spoken to by A, where n is the section and 0 <= n <= 2.

The pseudo-code for this include is spoken to in **Algorithm4(Count "OFF" pixels)---**

```
for each Area n do
Count the number of "OFF" pixels (An);
end
```

vi) Suggest that the RPA moves to the region with minimal measure of dark pixels (absolutely 0): We decipher that the most proper route for the RPA to progress is the region A with littler measure of dark spots. We think about the dark pixels as obstructions and white pixels as open territory.

The decision table is:

IF		AND	Action
A(0)>	A(1)	A(0)>A(2)	turn left
A(1)>	A(0)	A(1)>A(2)	go center
A(2)>	A(0)	A(2)>A(1)	turn right

If two or more areas have the same amount of pixels RGB(0,0,0), we apply the following decision table:

IF	Action
A(n) = A(1)	go center
A(0) = A(1)	turn right

If the number of black pixels of the suggested area (An) is greater than p% (p percent) of the total area's pixels (Tn), we assume that the UAV is larger than the space available to pass,soitsactionwillstop.

The rate measure p is a parameter and will rely upon the components of the RPA that is executing the calculation.
When it is resolved that the RPA gets an opportunity to progress to one side, left or forward, and that there is adequate measure of free space for this development, it ought to be evaluated whether the free space is orchestrated so as to enable the RPA to go through the picked portion. In the event that the white pixels are dispersed their entirety might be more prominent than the elements of the RPA, yet their plan in space may anticipate the RPA development.

Now other Artificial Intelligence methodologies could be utilized to take further choices, yet that falls outside the extent of our work. Our thought is to offer a snappy fundamental choice on the heading the RPA ought to pursue.

The move choice (forward right, left forward or stop) is deciphered in flying directions and transmitted to the RPA that will execute it.

vii) Discard the prepared picture and supplant it by another casing, presently with the new situating of the RPA: Once the flight guidance is passed to the RPA, another picture is caught at once t + 1, and all handling is rehashed.

E. IMPLEMENTATION

Our calculation was actualized in C++ and keeps running on the ROS stage. ROS (Robot Operational System) is an open source innovation made to help analysts in creating mechanical applications. ROS gives us numerous apparatuses and offices that were exceptionally valuable in our work.
A ROS application is a system of hubs that speak with one another. Every hub is a free program in the framework and a Master hub likewise exists to deal with the system. Hubs that produce information distribute this data in themes as messages, while hubs that need that information buy in to the relating subjects and get the distributed messages.

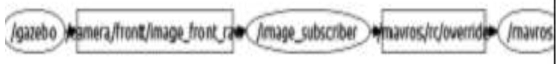

Fig. 6. Diagram generated by the program rqt graph showing the relevant nodes and topics in the application

In our application we made a hub called im-age supporter, which buys in to the point where the picture messages from the camera are being distributed and is in charge of preparing the pictures and choose where the RPA should move. In a genuine execution the picture messages would originate from a hub associated straightforwardly to the camera equipment, however in our usage the Gazebo test system distributes the information from the mimicked camera. Another hub, called Mavros, is in charge of speaking with the Pixhawk. At the point when the picture endorser hub chooses where to move the RPA, it makes a message of sort mavros msgs/OverrideRCIn, which speaks to a direction from a radio controller, fills it with the comparing esteems to cause the ideal development and distributes it to a theme where Mavros is bought in. Mavros makes a MAVLink message containing that data and sends to the Pixhawk through a sequential association. Figure 6 demonstrates a chart produced by a ROS instrument called rqt diagram demonstrating the hubs and the applicable points.

III. SIMULATION

To test and assess our technique we directed recreations of two distinct situations: a straightforward one in which the automaton needs to maintain a strategic distance from just a single obstruction and an increasingly mind boggling one where there were a major number of hindrances. In every situation the automaton needs to push ahead into the deterrents and stay away from the greatest number of as it can. The position information of the automaton was gathered and plotted in 2D graphs utilizing MatLab[17] for a decent representation of the reenactment.

a) One Obstacle

The first and essential recreation comprises in a situation containing the automaton and just a single deterrent before it, appeared in Figure 7. The obstruction is a 1 meter long, 1 meter wide, 2 meters high box, situated at 5 meters from the automaton's beginning position. The automaton is instructed to fly at a steady 1 meter height. The consequences of this test can be found in Figure 8. As we see the automaton had the capacity to evade the single impediment easily, keeping a separation of in any event 1.5 meters.

b) Several Obstacles

In the second test Gazebo was told to create 35 arbitrarily orchestrated snags in a 35x21 meters territory (fig 1). Every hindrance is the equivalent 1x1x2m box of the primary test, anyway since they're situated haphazardly the separation between every one is obscure. The automaton was directed to fly at a steady 1 meter elevation. It was seen that even with the high number of deterrents the automaton was as yet ready to maintain a strategic distance from crashes while exploring through the region.

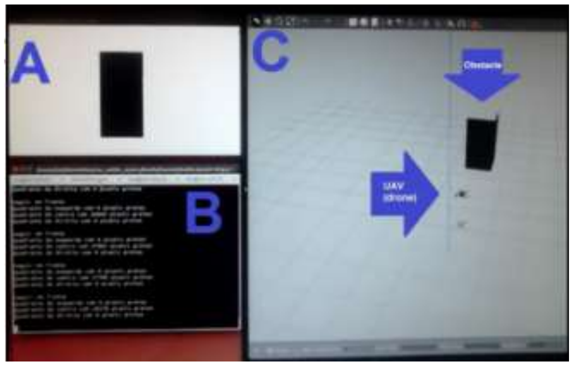

Fig. 7. Simulated Environment

A is the frontal vision of the camera
B is the command and result screen
C is the simulated output

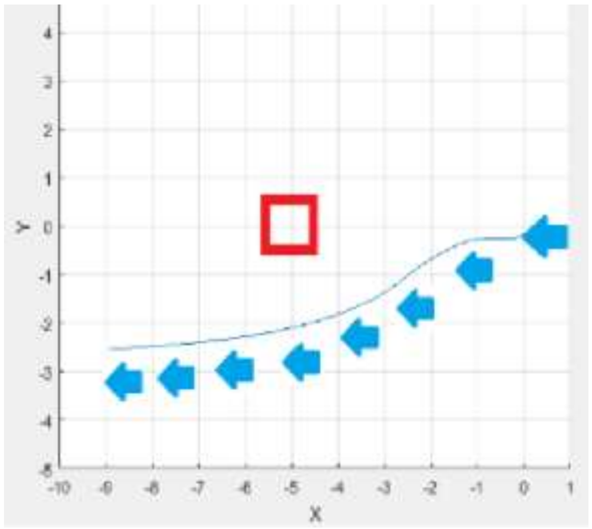

Fig. 8. Drone’s Trajectory (in meters).

Red means obstacle and blue is the drone’s path

IV. RESULTS AND DISCUSSIONS

The tests demonstrate that the calculation is effective and effectively moved the drone far from every one of the deterrents it experienced in its way. Anyway this work constrained itself to maintaining a strategic distance from the snags, however not executing a flight plan. In this specific circumstance, we can say that the objective of the calculation was effectively accomplished. Recovering the drone on its unique course after it evade a hindrance isn't the extent of this calculation and there are different codes to do this. Additionally we do exclude the estimation of the separation between the hindrance located and the automaton in flight. Again the reason is the presence of different algorithms to play out this errand. The underlying center is for open air flight, and we don't execute developments over or under impediments. Just deviations to the sides and to the middle were examined.

V. CONCLUSION

The algorithm was demonstrated productive in driving the drones through obstructions in the reproduced condition without causing collisions. There are anyway a few impediments and conceivable upgrades. The principal confinement is that the algorithm is just equipped for moving the drone sideways. Since quadrotors are equipped for vertical and full 3D development, an improvement is stretch out the calculation to drive the robot under or over hindrances as well. Another confinement is identified with the lights in the environment. It is realized that computer vision arrangements are touchy to changes in lighting, so it would enthusiasm for future attempts to test our algorithm in various conditions a notwithstanding executing and testing it in a real-life situation.

VI. ACKNOWLEDGMENT

I appreciate the guidance of my professors Remya Sivan & Pallavi N and the funding which is provided by our college Atria Institute of Technology.

REFERENCES

[1] https://en.wikipedia.org/wiki/Machine_vision
[2] http://www.bmva.org/visionoverview
[3] Dronestagram http://www.dronestagr.am
[4] Drones popularity http://oglobo.globo.com(in portuguese)
[5] SBEGHEN, Renam Costa, Processamento Digital de Imagem” (in portuguese), Monografia defendida e aprovada na FAJ, 2007

[6] Miranda Neto, Athur de, ”Navegac¸a˜o de roboˆs baseada em monovisa˜o” (in portuguese), Campinas,SP-Brazil, 2017
[7] Borland https://www.microfocus.com/pt-br/borland,
[8] ROS Indigo (Robot Operation System), http://www.ros.org
[9] Gazebo Simulator 8.0.0, http://gazebosim.org
[10] Ascending Technologies, GmbH, http://www.asctec.de
[11] Koushil Sreenath, Taeyoung Lee, and Vijay Kumar. Geometric Control and Differential Flatness of a Quadrotor UAV with a Cable-Suspended Load. In IEEE Conference on Decision and Control (CDC), pages 22692274, Florence, Italy, December 2013.
[12] Pixhawk, https://pixhawk.org/
[13] MavLink http://www.mavlink.org/
[14] Bueno, Andr, Fundamentos da Computao Grfica(in portuguese), Pon- tifcia Universidade Catlica, Brasil: Rio de Janeiro, 2011
[15] L.F.Lima, https://lazarolima.wordpress.com/2010/08/19/processando - imagens-em-grayscale-e-negativo-em-c/ (in portuguese),2010
[16] MatLab, MATrix LABoratory, http://matrixlaboratory.com