Computer Vision Based Solutions for MAV Target Detection and Flight Control

M. F. S. Neto[1,2*], G. S. Eduardo[1,2*], E. C. Silva[1,2*] and W. Caarls[1*]
[1]Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil
[2]AeroRio, Pontifical Catholic University of Rio de Janeiro, PUC-Rio, Rio de Janeiro, Brazil

ABSTRACT

Computer vision applied to localisation and target detection has been a field of study in the literature for some years. Conventionally, MAVs used to rely solely on distance sensors and ran on simple and modest embedded devices. The huge increase of computational power made possible the use of more complex computer vision algorithms for real-time embedded applications. This paper focuses on evaluating different types of computationally demanding algorithms, such as accumulator-based image transforms (Hough, Radon), mathematical morphology and Monte Carlo approaches, to process data fed by a single camera in order to aid a MAV to navigate through an obstacle course.

1 INTRODUCTION

In recent years, there has been a marked increase in research related to multicopters and UAVs (Unmanned Aerial Vehicles). Recent market research showed that the global market revenue of drones was worth US$ 6.0 billion in 2017 and is expected to grow up to US$ 11.2 billion by 2020 [1]. The development of UAVs has been driven by recent advances in computational technology, software development, lightweight materials, global navigation systems, advanced data links, sophisticated sensors and component miniaturisation.

Some UAVs may navigate autonomously by continuously monitoring data from IMUs and a GPS. However, in order to perform complex tasks in confined areas, small autonomous drones will need more complex levels of control autonomy and extra sensors in order to identify features of its surroundings and perform safe and stable trajectories. Computer vision is an often used method of sensing for small UAVs due to its reduced mass and energy consumption compared to other methods, such as LIDARs and sonars [2].

This work aims to develop computer vision-oriented control and decision-making algorithms in order to allow a MAV (Micro Air Vehicle) to perform an obstacle course completely autonomously. More specifically, the hoops element of the IMAV2018 course, that consists in making the MAV fly through a sequence of five hoops, with ellipsoidal geometries, with five different sizes. The smaller the hoops, the higher the score, but it increases the complexity of the task as well.

The objective is to properly identify a hoop and infer its position referential to the MAV, using image processing techniques. Studies will be carried out on classical methods of image processing for object detection [3] while evaluating different algorithms by their accuracy, processing time, noise and deviation, to verify their performance in the accomplishment of the tasks of interest.

2 METHODOLOGY

The developed MAV is based on the Emlid Navio2 flight controller paired with a Raspberry Pi 3 microcomputer. The system will receive environment inputs via a single Raspberry Pi Camera, pointed forward.

For the hoop detection (a hollow ellipse), which in the present work is considered to be almost a circle, the goal is to estimate the coordinates of the centre of the shape, x_o and y_o, and the radius, ρ. In order to achieve that, a mix of different circle detection approaches (Accumulator-based and Monte Carlo) with different methods of image pre-processing (mathematical morphology, clustering and fuzzy segmentation) were used. In total, 7 different combinations were evaluated, they are: Random Sampling Consensus (RANSAC) using a binary image; Randomised Circle Detection (RCD) using a binary image; RCD using a binary image obtained by clustering; Radon Transform over a fuzzy-segmented image; Hough Transform using a binary image obtained by clustering; Hough Transform using a fuzzy-segmented mask; Hough Transform using a greyscale image obtained from the Lab colour space. All the methods were implemented using the Python language and the OpenCV library.

2.1 Accumulator-based approaches

Accumulator-based approaches use a "voting array" with N dimensions, each corresponding to a parameter of the shape to be detected. In case of a hoop, the three necessary parameters results in a 3-dimensional array. The two most common methods for accumulating the array are the Hough transform and the Radon transform.

The conventional way of implementing the Hough method involves determining, for every segmented pixel in a binary image, all possible circles that might contain said

*Email address(es): manoelcfeliciano@gmail.com, guiseduardo@gmail.com, edusilva@ele.puc-rio.br, wouter@ele.puc-rio.br

pixel, and then mapping the parameters of these circles to the accumulator array. Each element of the accumulator array represents a number of votes for the possible circles present in the image. The presence of the local maximums indicate strong evidence of circles with parameters described by their coordinates in the accumulator array [4]. Building a 3D accumulator array is very computationally demanding. In case of large images, the controller performing this kind of algorithm could experience memory issues, since the large amount of possible parameters generates an even larger accumulator array [4]. Because of this, the OpenCV implementation of the Hough transform for circles uses a technique called the Hough Gradient Method that uses just a 2D accumulator. This method receives, as function parameters, a radius threshold, limiting the radius of found circles to a minimum of 10 pixels and maximum of 250 pixels, high threshold for the Canny edge detection of 50, the low being twice smaller, and the accumulator threshold for circle centres of 35. These parameters were empirically chosen by experimenting with a variety of images and footage.

As for the Radon method, the density of each element of the accumulator array corresponds to the match between the image and a template generated using the parameters given by the coordinates of the given element [5]. This match is obtained by calculating the inner product between a segmented image emphasising the hoop (e.g. using a fuzzy membership function) and the projection of the template into a blank image.

2.2 *Monte Carlo approaches*

Monte Carlo approaches form a class of algorithms that relies on the repeated random sampling of the process inputs. They come as an alternative when numerical or analytic strategies are not practical or possible for the solution. Furthermore, they benefit from the fact that they do not need an accumulator. There are two common methods for detecting circles which use this kind of approach: the Random Sample Consensus (RANSAC) and the Randomized Circle Detection (RCD) algorithms.

The RANSAC algorithm proposed by Fischler and Bolles [6] is a classic implementation of this approach in the computer vision field. Its working principle is based on the robust estimation of a number of parameters from a model using a random number of hypothesis. The main difference from other common robust parameter estimation algorithms is the use of the smallest number of observations possible to obtain the initial solution of the problem. This is achieved by first solving for the model using three random samples each time and then verifying the degree of trust of the estimations based on the inliers that follows a predefined threshold. The output of the algorithm is given by the estimation with the highest degree of trust in a voting procedure. Therefore, the main advantages of the algorithm lie not only in its robustness to the presence of outliers but also in its efficiency. It is important to note that, for the RANSAC method in this work, the edges of the hoop are not detected using Canny edge detection, but using the external contour retrieval OpenCV function.

Following a different strategy, the RCD algorithm proposed by Teh-Chuan Chen and Kuo-Liang Chung is based on a voting procedure in the parameter space. It works by first selecting 4 random edge pixels from the image using a distance criterion to determine the existence of a possible circle. Further, it verifies the circle candidate by using an evidence collection process such as the number of edges pixels that lies inside it [7]. This algorithm uses a number of thresholds to achieve the previous strategy with the following empirically determined values for the test environment. The distance threshold, limits the distance between a circle candidate and each edge pixel to 40 pixels. The ratio threshold, limits the ratio between each edge pixel and the number of pixels in the boundary of the circle candidate to 0.5. The minimum distance between two edge pixels of a possible circle is 10 pixels. The circle detection task is stoped if there are less than 10 edge pixels in the set. Finally, the maximum number of attempts in detecting a circle is set to 10.

2.3 *Image pre-processing*

As described in sections 2.1 and 2.2, each method needs to work with a segmented image, that is, an image that emphasises the hoop and/or its features by combining different colour space channels of the capture, so computer vision algorithms such as the Hough Transform can easily detect the correct edges in the image, as the present noise is mitigated. The standard colour space used by cameras is the RGB (Red-Green-Blue). In this work, however, the segmentations are obtained using the HSV (Hue-Saturation-Value) and Lab (Lightness-green/red-blue/yellow) colour spaces, because these setups enhance the contrast between the hoop and the background.

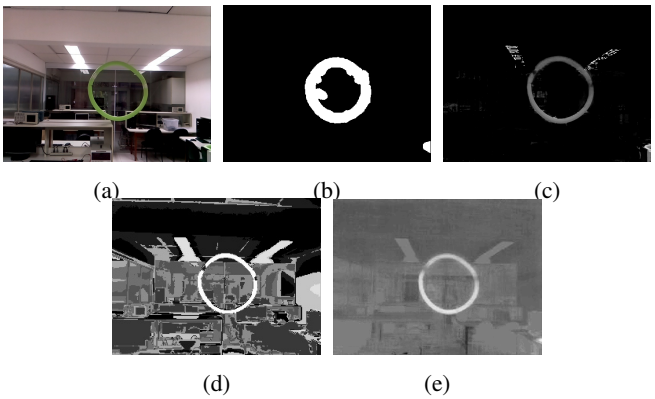

Figure 1: Different methods of image pre-processing. (a) Original frame. (b) Binary segmentation. (c) Fuzzy segmentation. (d) KMeans mask. (e) Lab 'a' channel.

The simplest form of segmentation is the generation of a binary image by verifying the pertinence of each pixel to a

previously-set threshold. In the case of the HSV channels, applying a threshold to the 'Hue' channel, for example, will emphasise the pixels belonging to a certain range of colours. Figure 1b is an example of a binary segmentation. This kind of image can be further improved by applying morphological operations, such as erosion, to remove noise, and dilation, to close gaps in the image. For the RANSAC algorithm, the image is generated by applying a threshold to the 'Hue', Saturation and Value channels, respectively, at the intervals (30;60), (32;255) and (50;255) and, then, eroded once and dilated 4 times in order to remove noise and close gaps. On the other hand, for the RCD algorithm, the image is generated by applying a threshold only to the 'Hue' channel, at the interval (30;80) and, then, dilated 3 times. These ranges were empirically established by experimenting with a variety of images and footage.

The Fuzzy segmentation works likewise, but instead of generating a binary image, it generates a greyscale image where the value of each pixel represents its membership grade to the specified group of parameters (in the HSV image, the parameters would be colour, saturation and brightness). The obtained image is, then, used as a mask over the original greyscale image, resulting in Figure 1c. In this work, the fuzzy membership function used is just a trapezoidal function applied to the 'Hue' channel, with parameters P = (20;40;60;80).

The KMeans segmentation uses the Scikit-learn clustering function to identify clusters of pixels with the same colours in the HSV space. Then, this information is used to generate a mask that better emphasises the hoop and hides unwanted portions of the image. Figure 1d shows a binary image used as a mask, obtained by the KMeans method.

3 Tests and Results

3.1 Preliminary observations

During the development phase, each hoop detection method was tested a set of pictures of hoops similar to the ones used in the obstacle course. This is to ensure that every method would display reasonable levels of detection rate, accuracy and processing time.

One of the most important parameters for using computer vision in drones is the processing time of each frame. Optimisations such as subsampling the frames down to 160 px width, maintaining the aspect ratio, had to be made in order to improve the real time performance. However, this was not enough for the Radon method which was able to correctly detect the hoop, but needed 37 minutes and 46 seconds to process a single frame in the Raspberry Pi, an impractical amount of time for live applications. The result for this particular algorithm was expected since it is based on a 3D accumulator and does not have any optimisations, other than image subsampling, implemented. Thus, this method was not included in the next phase of testing. It should be noted that better results are expected with a GPU due to the parallelizable nature of this method.

3.2 Live testing

This phase of testing consisted in comparing the performance of each algorithm using the MAV's hardware for varying positions in relation to the hoop, as well as different viewing angles. The MAV was kept at the same position on a still surface at all times with its motors turned off.

In each test, the Pi Camera captured a set of 100 frames after which the Raspberry Pi proceeded to process the frames using the methods chosen, finding the parameters of the hoop in pixels, proportional to the camera resolution. The parameters were then combined with known information, such as the actual average radius of the hoop (the hoop used is ellipse-shaped, with minimum diameter of 780 mm and maximum diameter of 860 mm, an average of 820 mm), camera resolution and field of view (for the Pi Camera, it is 62.2° horizontally and 48.8° vertically), in order to estimate the relative position between the MAV and the hoop. Thereafter, the methods were evaluated by the root mean square error (RMSE), standard deviation and confidence interval (95%) of the measured longitudinal distance to the hoop and the euclidean distance (Y and Z axes) to the centre of alignment, processing time and the ratio of valid detections.

First, the hoop was positioned at the minimum distance of detection from the MAV, 140 cm. Then, in steps of 30 cm, the hoop was brought farther from the MAV until the distance of 410 cm. In this test, not only the perceived radius of the hoop was smaller, but the perceived thickness of the edges was also thinner, affecting the image segmentation. Furthermore, the changes in luminosity in different parts of the room where the tests were conducted might interfere with the hoop detection, requiring each algorithm and image segmentation method to be robust against these changes.

Algorithm	**Pre-processing (ms)**	**Detection (ms)**
Hough Fuzzy	16.90 ± 1.12	6.38 ± 5.13
Hough KMeans	438.78 ± 166.69	62.43 ± 14.48
Hough Lab	9.64 ± 0.53	4.28 ± 0.70
RANSAC	44.25 ± 2.60	23.22 ± 6.71
RCD	44.25 ± 2.11	50.71 ± 13.59
RCD KMeans	441.11 ± 177.52	1334.39 ± 564.57

Table 1: Average times ($\pm$ standard deviations) required for each algorithm to process one frame and detect the hoop.

Table 1 shows the average processing time plus standard deviation required for each method to pre-process an image and detect the hoop over all the distances tested. All the methods kept consistency even when the rates of detection seen in Figure 2a dropped to very low values. The Hough Fuzzy, Hough Lab and RANSAC methods displayed processing times fast enough for high-framerate applications. The processing time is noticeably affected by the image

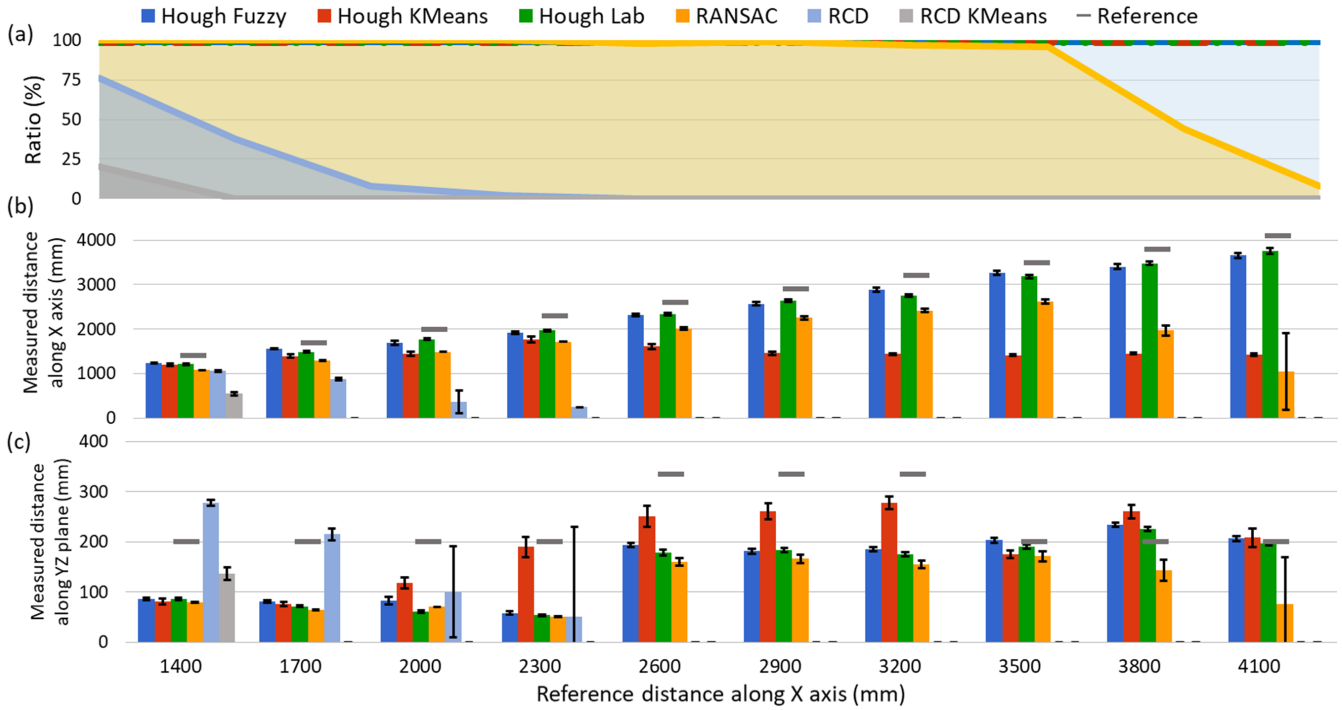

Figure 2: Algorithm performance for different distances between 1400 mm and 4100 mm. (a) Ratio of valid detected frames. Notice how all the Hough methods achieved 100% detection. (b) Estimated distance along longitudinal axis for each algorithm, with reference distances and confidence interval of 95%. (c) Estimated distance over the frontal plane (YZ axes) for each algorithm, with reference distances and confidence interval of 95%. The reference distance along the YZ plane is (0;-200) mm for all reference distances along the X axis, except between 2600 mm and 3200 mm, where it was set to (-300;-150) mm due to placement difficulties in the testing room.

pre-processing method chosen, as the methods that use the KMeans clustering took longer on average to process the segmented frame.

Figure 2a shows how the detection rates of the hoop drops as the distance from the camera increases. All the Hough methods achieved 100% detection rate at all distances, while the other methods displayed a falloff in detection rates past a certain distance. This happens because the hoop becomes very thin at increased distances, becoming faded after morphological operations, impairing detection by the Monte Carlo-based methods. This doesn't affect the Hough-based methods because those rely on Canny edge detection, identifying the presence of the hoop as long as it contrasts with the background.

Figure 2b displays the evolution of the average distance estimation and confidence interval along the longitudinal axis (X axis), calculated based on the ratio between the hoop size in the capture, in pixels, and the actual hoop size, in millimetres. Figure 2c displays the average estimation and confidence interval of the euclidean distance over the YZ plane (vertical and lateral axes) between the hoop and the centre of the capture.

It can be said that the Hough Fuzzy, Hough Lab and RANSAC methods were the best performers in this test by demonstrating high detection rate, low processing time, the least errors compared to other methods and little dispersion of results. It can also be concluded that the perceived radius of the hoop is greater than its actual radius, leading the algorithms to estimate shorter distances to its centre. This effect may be due to the eccentricity of the hoop, causing the detected radius to be its semi-major axis instead of its mean radius, preset in software.

The other methods perform worse due to various reasons. Besides the long processing time, the Hough KMeans method couldn't correctly identify the hoop, as evidenced by the almost constant distance estimate regardless of the actual distance. Furthermore, both RCD methods could only properly detect the hoop within close proximity, making them unreliable for longer distances.

Another round of tests consists on evaluating the performance of each algorithm in detecting a rotated hoop. The motivation for this kind of test is that the MAV might not always be perfectly aligned with the hoop, requiring each algorithm to detect it nonetheless. In addition, each algorithm will perceive the hoop as an ellipse with increasing eccentricity as the angle of rotation increases. The routine for this test is similar

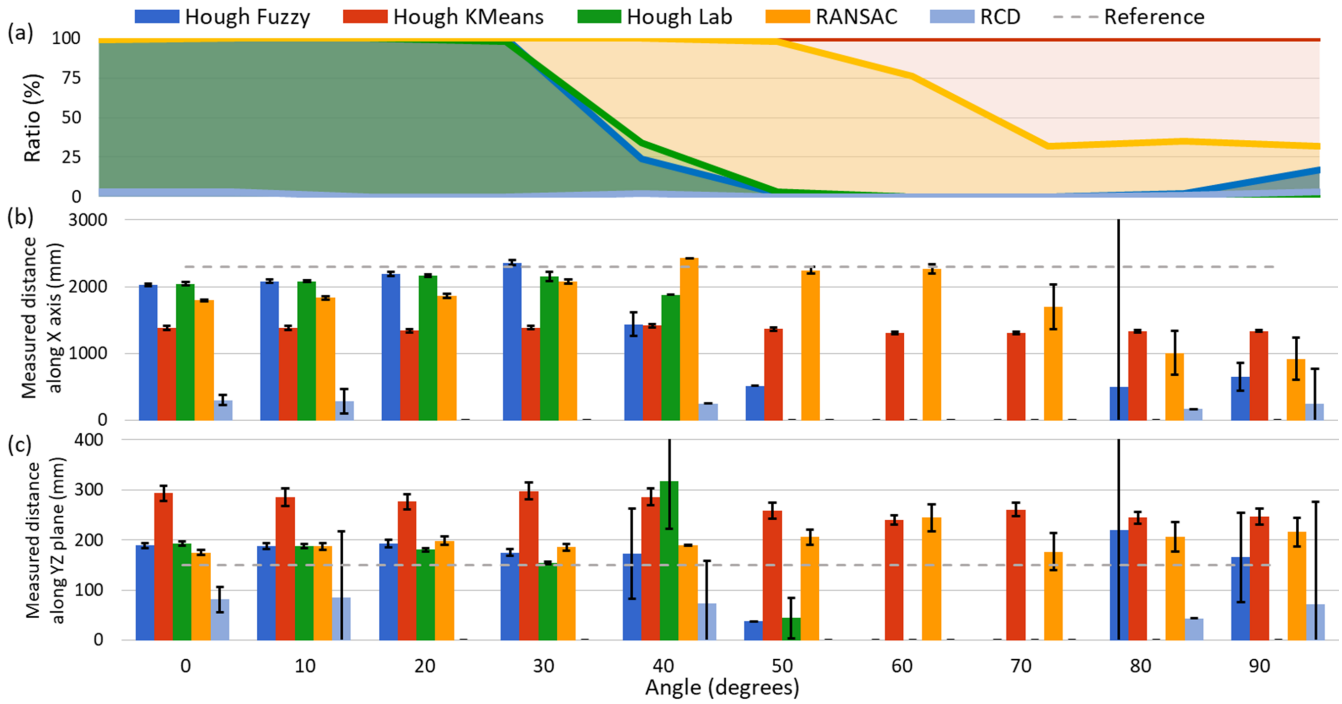

Figure 3: Algorithm performance for different hoop rotation angles at a distance of 2300 mm. (a) Ratio of valid detected frames. At this distance, both RCD methods had very low and 0 detection rates. (b) Estimated distance along longitudinal axis for each algorithm, with reference distances and confidence interval of 95%. (c) Estimated distance over the frontal plane (YZ axes) for each algorithm, with reference distance ((0;-150) mm for every test) and confidence interval of 95%. The three algorithms with best results (Hough Fuzzy, Hough Lab and RANSAC) kept consistency in measurements with good detection ratios.

to the previous one: starting from 0° rotation (with the MAV completely facing the hoop head on), in 10 steps of 10°, up to 90° (hoop sideways in relation to the MAV), each algorithm evaluates a set of 100 frames per step.

This test was performed by placing the hoop at 2300 mm from the camera. This distance was chosen because the hoop occupies a reasonable area of the frame. However, both RCD methods underperform, as shown in Figure 2b. In spite of that, both of them were also subjected to these new tests. The RCD KMeans results were omitted in Figure 3, because its valid detection frames ratio remained null for all analysed cases.

Figure 3a shows the detection rate of valid frames as the angle of the hoop increases. As expected, detection rates drop past certain angles, except for the Hough KMeans method that, as observed in previous tests, might be detecting noise instead of the actual hoop.

Figures 3b and 3c show the distance estimations along, respectively, the longitudinal axis and the YZ plane, as the rotation of the hoop varies. As expected, measurements at angles where detection rates are low are mostly noise, as evidenced by the greater dispersion of results past these angles.

An interesting phenomenon in the measurements was a steady increase in the distance estimation by the Hough Fuzzy, Hough Lab and RANSAC methods while keeping low dispersion. As previously said, the estimated distance is calculated from the ratio between the measured radius from the capture, in pixels, and the actual radius of the hoop, in millimetres. This leads to the conclusion that the estimated radius found by these algorithms becomes smaller with the increase of eccentricity of the hoop seen from the MAV's point of view.

As a final test, each detection method is evaluated on its ability to detect the hoop while it is partially outside the camera field of view. This test is performed at the constant distance mark of 2300 mm by the longitudinal axis and increasing distances by the lateral axis, so the hoop can be partially visible by 25%, 50% and 75% of its total area. The 100% and 0% visibility tests are also included for comparisons.

Figure 4a shows how the detection rate drops as the hoop moves away from the field of view. The obtained results indicate that the Hough Fuzzy, Hough Lab and RANSAC methods still can identify the hoop even in situations where it is not fully contained in the frame.

Figures 4b and 4c show how the visible hoop area affect the distances estimated by the algorithms. The results confirm that Hough Fuzzy, Hough Lab and RANSAC algorithms can correctly identifying the hoop position and radius, in spite of

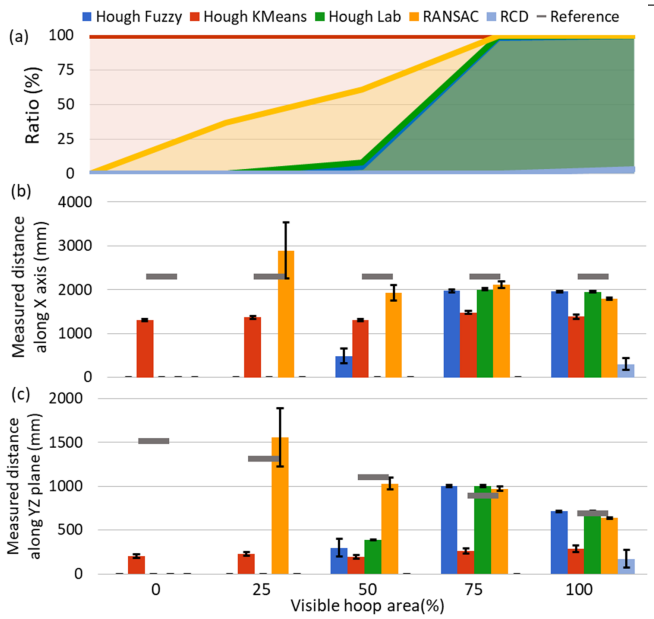

Figure 4: Algorithm performance as the hoop moves outside the camera field of view. (a) Ratio of detected frames. (b) Estimated distance along the longitudinal axis for each algorithm, with reference distance and confidence interval of 95%. (c) Estimated distance over the frontal plane (YZ axes) for each algorithm, with confidence interval of 95% and reference distances of (-1510;-150) mm, with the hoop outside the frame, and the distance along the Y axis increasing in steps of 210 mm until (-670;-150) mm, when the hoop is fully into view.

the partial concealment of the hoop. For visible hoop areas as low as 75%, these three methods measured similar distances during valid results, along the longitudinal axis and the YZ plane. However, regarding this aspect, it is noted that the RANSAC method outperforms the others, since it returned adequate measurements for visible hoop areas as low as 50%, while others start to underperform earlier.

4 Conclusion

There are many different algorithms used for implementing computer vision for different applications, each one with their own advantages and disadvantages. This work evaluated the performance of different combinations between 4 image segmentation methods (binary, fuzzy, Lab grayscale and clustering) and 4 circle detection algorithms (Hough transform, Radon transform, RANSAC and RCD), focusing in identifying and measuring a hoop in order to aid a MAV to traverse it. The best suited algorithms for the task were selected by comparing their better performance and reliability. The Hough Fuzzy, Hough Lab and RANSAC algorithms presented shorter processing time, higher detection rates and better accuracy.

These 3 methods performed well, however they have some distinctions that are relevant to the objective of traversing the 5 hoops of the obstacle course. The RANSAC method works really well for adverse situations, such as detecting a hoop with high eccentricity (or rotated more than 40°) or a hoop that is not fully viewed by the camera (regular detection rate for a visible area as low as 50%). However, the MAV might not have to deal with these kind of situations, making the Hough approaches preferable, because of their faster processing times and better accuracy, especially the Hough Lab method, that presented the best performance in the tests conducted.

4.1 Future work

This work has evaluated 7 computer vision approaches based in different situations with a stationary MAV. In the future, new tests will be conducted with a flying MAV in order to evaluate different algorithms, considering the effects of vibration and motion.

Furthermore, the performance of other methods and their combinations with pre-processing algorithms will be evaluated in future works, including methods based in deep learning (with the appropriate hardware support).

Acknowledgements

We would like to thank the Brazilian funding agencies CNPq, FAPERJ and CAPES for continued support and supplied resources.

References

[1] C. Stamford. Gartner says almost 3 million personal and commercial drones will be shipped in 2017, 2017. https://www.gartner.com/newsroom/id/3602317 (Accessed on 05/05/2018).

[2] D. Floreano and R.J. Wood. Science, technology and the future of small autonomous drones. *Nature International Journal of Science*, 521:460–466, 2015.

[3] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Publishing Company, New York, NY, 2013.

[4] H.K. Yuen, J. Princen, J. lllingworth, and J. Kittler. Comparative study of hough transform methods for circle finding. *Image and Vision Computing*, 8(1):71–77, 1990.

[5] M. van Ginkel, C.L. Luengo Hendriks, and L.J. van Vilet. A short introduction to the radon and hough transforms and how they relate to each other, 2004.

[6] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] T.C. Chen and K.L. Chung. An efficient randomized algorithm for detecting circles. *Computer Vision and Image Understanding*, 83:172–191, 2001.