Sensor node detection from mid-air-images

A. Renken[1], N. Najeeb[2], T. Craig[3]

Index Terms— Hough Transform, Color filter, UAV.

Abstract - This paper describes techniques developed to detect a sensor node in a picture taken from a drone in mid air. Optimal operating conditions where the algorithm succeeds were identified. Operating conditions include lighting characteristics, node occlusion and tilt. Robotic flight was implemented to maintain accurate navigation and altitude.

1 INTRODUCTION

When deploying a sensor network one major issue is keeping the network alive, or prolonging the age of a network. Powering the individual nodes is sometimes time consuming and a dangerous task if the sensors are located at hard to reach places as seen in Fig. 1. Deploying a robot to do the job of charging a sensor is both faster and safer. The problem of actually locating the sensor is a challenge. Being able to identify the sensor node from images will prove very helpful in assisting the robot to reach its destination and achieve its task. Identifying the node based on its robust features and unique qualities compared to its surroundings is the approach we chose to investigate in our work. Our goal is to identify the node Fig. 2 and determining our level of accuracy, and also be able to identify the node within a range of lighting conditions. The node may also be partially occluded, but it should still be identifiable within reasonable conditions, for example a 10% occlusion should not prevent us from identifying the node.We investigated identifying the node even when it is not normal to the camera lens, and determine the maximum angle off normal where the node was still identifiable by the algorithm.

Fig. 1. A sensor node under a bridge being maintained.

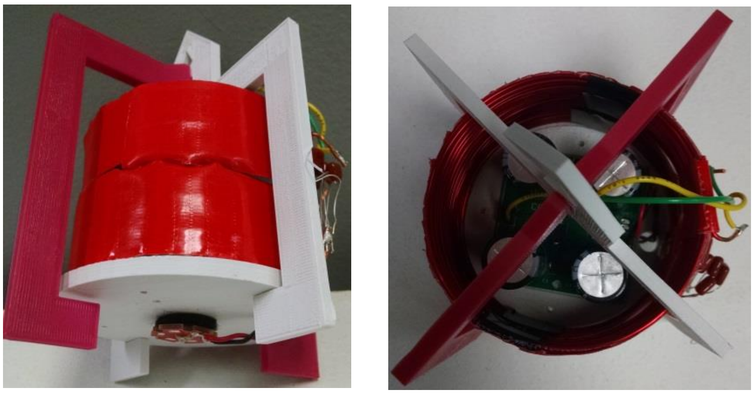

Fig. 2. Sensor node used in our project side and top view.

1. *Department of Mechanical and Materials Engineering, University of Nebraska-Lincoln E-mail: arenken@huskers.unl.edu.*
2. *PhD student, Department of Computer Science and Engineering, University of Nebraska-Lincoln E-mail: najeeb@cse.unl.edu.*
3. *Department of Mechanical and Materials Engineering, University of Nebraska-Lincoln E-mail: lincolnkite@yahoo.com.*

2 Related Work

Our problem is recognizing a specific object in an image. Since we already know the shape of our node, and all our nodes share a similar general shape, we could apply any of the object recognition algorithms covered in class [1]. We used Hough transform to look for circles along with a color filter to help isolate the color band applicable to the node.

The Hough transform takes a grid of parameter values and then each point votes for a set of parameters, incrementing those values across the grid [1]. The maximum and local maxima are then found in the grid. This concept is usually used to find lines or features, but can be extended to finding circles. The parameters that define a circle are the center and the radius. The equation of a circle is:

$$(x - X_0)^2 + (y - Y_0)^2 - r^2 = 0 \quad (1)$$

where X_0 is the X value of the center coordinate and Y_0 is the Y value of the center coordinate. The radius of the circle is defined by r. Knowing the equation of the circle allows the program to scan through the points in the image and identify circles using the process explained for finding lines or features.

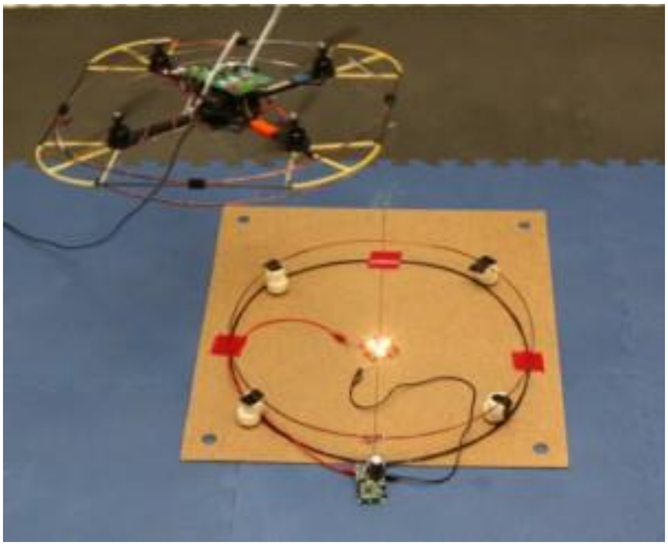

Fig. 3. Demonstration of top down wireless power transfer.

Our work was done in conjunction with NIMBUS (Nebraska Intelligent MoBile Unmanned Systems) Lab at the University of Nebraska - Lincoln. The NIMBUS Lab is an exciting place where the latest research and technology in software and systems engineering, robotics, and sensor networks converges to develop more capable and dependable UAVs (Unmanned Aerial Vehicles).[2] . In their wireless power transfer project, the NIMBUS Lab uses a sensor node seen in Fig. 2 as the receiver of wireless power transfer [4]. This has been shown to work experimentally with other drones equipped with an inductance coil. These experiments can be seen in Fig. 3 and Fig. 4 GPS (Global Positioning System) is used

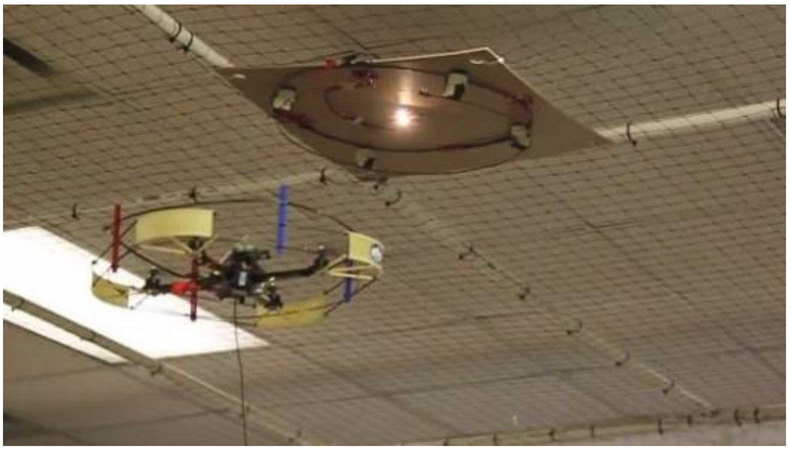

Fig. 4. Demonstration of bottom up wireless power transfer.

to assist the UAV in identifying the location of the node. This becomes problematic when more precise alignment is needed or when the node drifts for example, due to natural environment like soil erosion, wind, or animals.

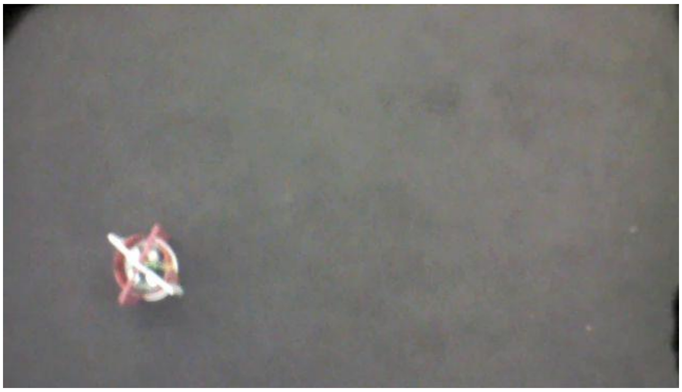

Fig. 5. Node in normal lighting in no cluter scene.

3 Problem Definition

A UAV or drone will be flying close to the sensor node (node) and capturing images. The drone used to capture the images for this project is an AR Parrot Drone taking pictures at a resolution of 640 by 360 pixels with a frame rate of 28 frames per second. Our program attempts to identify the node in the image. Fig. 5 and Fig. 6 shows two different node images a UAV captured from mid-air, with the node located in different environments for each image. The images of the nodes include some blurring due to flight vibration. The nodes are 11.6 centimeters by 11.6 centimeters by 10.5 centimeters. They have different colors, but both have portions in the red spectrum. They also have an inductance coil with a diameter of 8.2 centimeters. The nodes overall shape from a top view and the induction coil provides two circles of similar diameter to detect. These can be seen in Fig. 7.

Fig. 6. Node in cluttered environment.

The ability to identify the node in different lighting condition, partial occlutiom, and when it is at an angle are also problems that our algorithm attempts to address. The algorithm takes still images as input and reports the location of the node in the image by giving the virtual center of the node, and the radius. Having a UAV that could locate the node using visual input alone enables prolonging the life of a sensor network even when GPS cannot provide an accurate node location. This is critical to the prolonged life of a network because civilian GPS is only accurate within a ten foot radius at best. Natural and man-made obstructions can obscure the GPS satellites from view, adversely affecting the accuracy of GPS. These obstructions can be anything from trees to bridge overhangs. Visual identification also enables a sensor network to have some level of

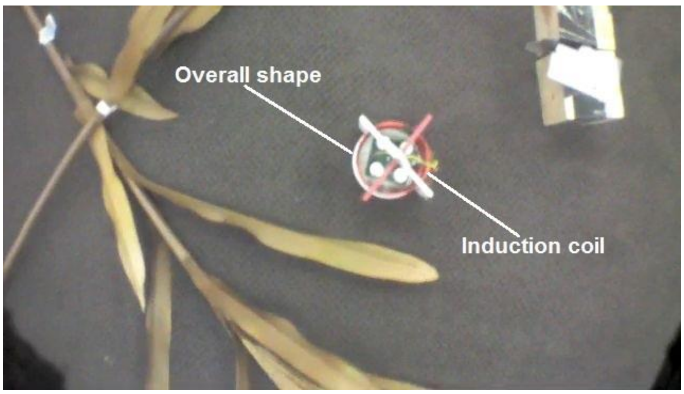

Fig. 7. Node overall shape and diameters.

mobility while still being tracked and identified. Due to time limitations we reduced the amount of possible noise compared to a real world deployment, and also be limited to two different nodes. Our first attempts used a sensor node in a setting by itself, then we started adding other objects to the scene. We also made sure the node is always in the captured image. This could be guaranteed in a real world situation by have the UAV directed to the vicinity of the node via GPS. We will also start with a fixed known altitude the UAV is hovering over the node. The altitude and location of the drone used was tracked using a VICON camera system. Knowing the altitude in outdoor environments is a good assumption as the drone has an ultrasonic sensor on the bottom of it, allowing it to track its altitude.

Fig. 8. Node in darker lighting conditions.

Several restrictions were placed on the images processed by the algorithm. This includes one, and only one, node being in each image. Other restrictions include the node being within 30 degrees of normal to the drone and obscured less than 50%. These maximum values were determined through experimentation.

Fig. 9. Paritally Occluded node in simulated environment.

4 OBJECTIVES

We had three objectives:

a- Finding the node under different settings.

b- Still being able to identify a partially occluded node.

c- Still being able to identify a tilted node.

Lighting throughout the year will change as the Sun's relative position changes. Lighting also changes with time of day and cloud coverage. Fig. 8 shows the node in a dimmer light environment. This will make detection in different lighting crucial to finding the node as lighting cannot be controlled outdoors. Changing light will make the color of the node appear slightly differently. The Sun will also slowly fade the color of the node. The node could be the only thing in a surrounding environment, such as under a bridge. The node may also be surrounded by other objects, such as under a tree. This means that the algorithm must be able to detect the node in many different types of lighting and with differing amounts of clutter. Detection

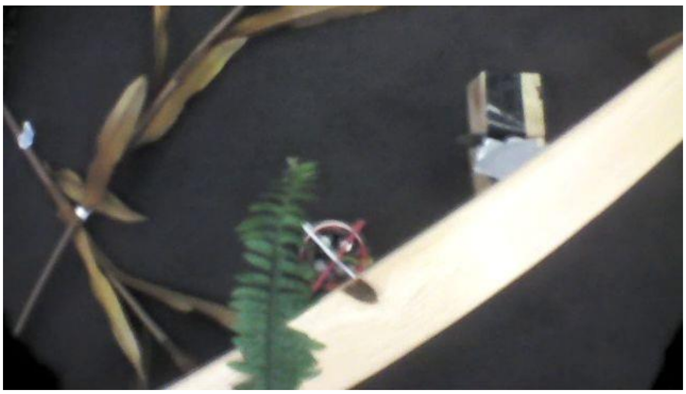

Fig. 10. Partially occluded node man-made obsticles.

with clutter is vital, as the algorithm must be able to distinguish between the node and objects with similar properties. In situations where there is clutter around the node, the clutter may also partially occlude the node. The node may not look like a circle if it is partially occluded as seen in Fig. 9 and Fig. 10. This presents a unique challenge of finding an arc with an unknown angle. This will probably occur when the node is deployed outdoors due to the environment. In nature, leaves blowing around, branches growing, and other natural events may partially cover the node.

The node may not alway appear as a circle or portion of a circle. If the node angular position is tilted, it will no longer appear as a flat circle but more elliptical. It is beneficial when the algorithm can tolerate a certain amount of angular movement of the sensor. Especially, since the UAV is still be able to transfer power or data to and from the node even when the node is not aligned with UAV camera normal.

5 Approach

The radius of the node is needed to optimize the Hough transform algorithm. This radius can be determined based on the radius height equation:

$$r = -38.562h + 80.466 \tag{2}$$

where h is the altitude of the drone in meters and r is the radius of the of the node in pixels. The equation was derived based on the line of best fit generated by the set of points shown in Fig. 11. To find the different radii, the drone was programmed to fly with the aid of VICON camera tracking system at different altitudes. The camera system allowed for precise altitudes determination. A sample of a low altitude and high altitude image can be seen in Fig. 12 and Fig. 13 respectively. The videos captured during this flight at different heights were then brought into MATLAB,

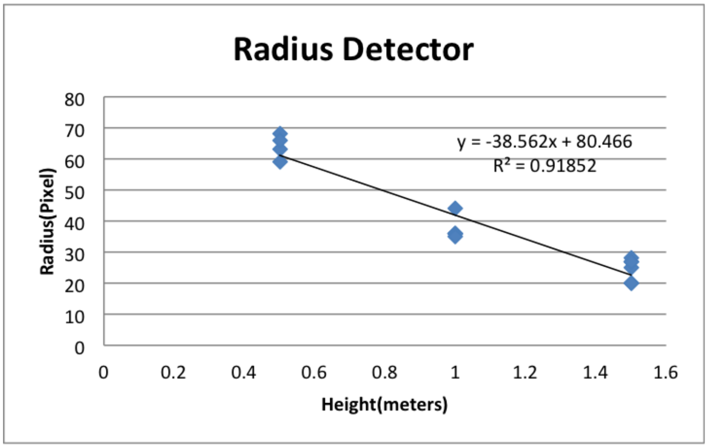

Fig. 11. Experimental results of radius vs height.

Fig. 12. Node as seen from altitude 0.6 meters.

Fig. 13. Node as seen from altitude 1.6 meters.

where the radii were found using the imtool function. This data was then plotted in Excel and a trend line was fit. Extrapolating from Eq. 2, it was found that images taken at a height above two meters would result in the node being too small to detect (4 pixels with our camera).

From this equation, we were able to place a radius value into the Hough circle detector, using a small safety buffer on both sides of the expected radius to account for various noises and variations in the image. This constraint, along with scanning through the images, allows the Hough Transform to find the circles. The object polarity indicates whether the circular objects are brighter or darker than the background. For the Hough transform to be the most effective, two different techniques were implemented by changing the object polarity. The two cases that were tested were circular objects that were brighter than the background and circular objects that are darker than the background, as seen in Fig. 14.

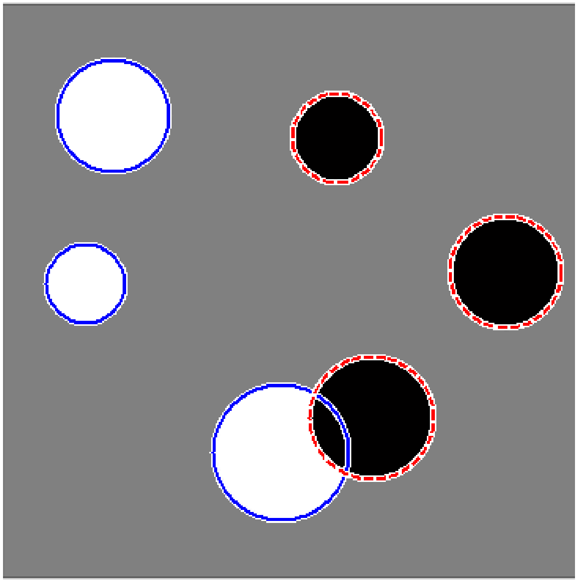

Fig. 14. Hough transforms detecting light on dark and dark on light circles.

There are multiple other features that can be used in the imfindcircles function. For example, the computation method may be changed. The computation method is the technique used to compute the accumulator array for the computational constants. For example, Atherton and Kerbyson's phase coding method is an option [4], along with using a two stage circular Hough transform.

The sensitivity factor for the imfindcircles function allows the amounts of viable circles to be defined. Circular arcs begin to be found as the sensitivity of the Hough Transform is increased. This is helpful in cases where the node is obscured or due to blurring from drone movement, causing the circle not to be seen clearly. If the user wanted to have only exact circles, the sensitivity could be lowered.

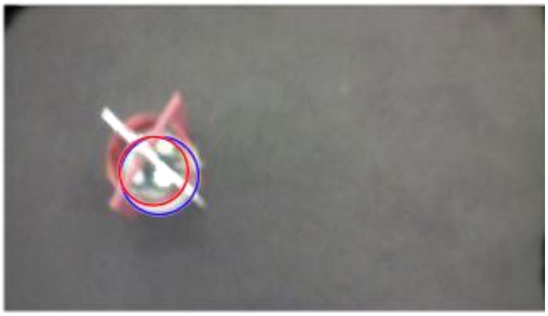

Fig. 15. Hough transforms detecting light on dark and dark on light circles on a node.

The edge threshold parameter for the function allows further tweaking and modification to be performed to optimize the results. The edge gradient threshold determines edge pixels of the circles in the images as shown in Fig 15. By changing this parameter the amount of weak and strong circles are defined. A higher parameter allows more circles to be found which lead to many false positives, as the value is lower the amount of circles found decreases but are more typically actual circles. By changing this parameter, the different lighting conditions can be accounted for.

The outputs of the imfindcircles function are the centers and radii. The virtual center of the node is given by x and y coordinates for where the Hough circle center is located in the image. The radii output gives an estimate radius for the associated circle. The process implemented runs images through the imfindcircles function twice, once with a dark polarity and then with the bright polarity. The radius is also passed with a buffer so that only correctly sized circles are found. The radius is determined using equation 2 by the height identified in the UAV flight control program. By applying a safety buffer to the expected radius the function gathers the centers locations from both functions. This gives the center locations of the viable circles. Although this process works to find some cases, another process needs to be completed to filter out excess noise and false positives. This extra constraint was applying a strict color filter that matched to the color of the image.

A B

Fig. 16. A sensor node passed through the color filter, A is the original image, B is the output of the color filter.

The idea behind color filter was to take the image and break it into the three color bands: red, green, and blue. From here, the color filter takes each pixel and checks to see if it is between a minimum and a maximum for each band. If the pixel is within that range for all the bands, it outputs a value of 1. If it is not within those ranges, it then outputs a zero. Lighting changes may affect the values needed for the upper and lower color bands but this was not found to be a significant source of error while testing and can be adjusted with accordance to light intensity. The values from the success cases are then placed into a new image to be processed. An example of color filter can be seen in Figure 16. The next step is to pad the new image with zeros in preparation to search the image for the max concentration of values.

We run a square box with same dimensions as the largest circle diameter detected via the Hough transform function through the padded image. The values in each box are summed and compared against the last largest value.

If the value is larger than the previous max, then it becomes the new value to check against and it logs that coordinate as the max. The square box continues until the entire image has been scanned and the global max concentration for the image is found. The center of the circle will be found from the box because the global max will be found when the box is all the way around the colored portion of the node.

After we find the center of the circle we use color detection, the center can be compared to the centers found from the imfindcircles. An image is created that is based on the center locations of both color and the centers information from the original image and the radius extending from the center location. A small safety buffer is added to account for the whole circle to be shown. A sample of this image created around the center of the node can be seen in Fig. 17. The blue ring shows the circle found by the algorithm and the red dot represents the center of the the virtual node. Caution is taken not to create an image that is out of the original image frame by checking the locations that the image will be made from and if it is trying to create an image that will accesses data that is out of bounds it sets that location to be the edge of the original image edge. This allows the images to be generated without problems, while allowing us to see if the results are successful.

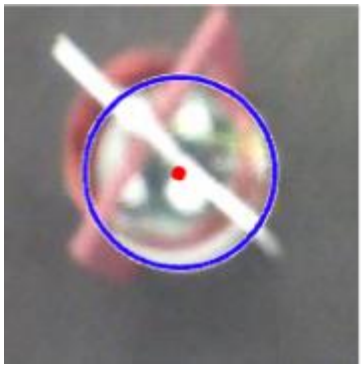

Fig. 17. Virtual node generation.

6 EVALUATION

The node detection algorithm was tested under several different environments, including obscured, different altitudes, different lighting, cluttered, and by itself. All of these different situations require minute tweaking in the parameters of the function to give optimal results. Most of these different situations could be accounted for with additional sensors that would assists the drone in optimal parameter selection while transitioning into the different environments. In total we gathered 16,000 images of the nodes. Of these, fifteen were processed from the height images, twenty from the occlusion, nine from angle, and five from illumination experiments. In addition to these images, 93 were used to evaluate algorithm robustness.

The drone was flown at different altitudes to get the trend line for the radius of the node as a function of height. Once the trend line was established, these images were processed by the node detection algorithm. The algorithm was very accurate at detecting the node in these images.

Lux	Detection	Color	Hugh	Bright Hugh
5	No	No	No	No
78	Yes	Yes	Yes	Yes
120	Yes	Yes	Yes	Yes
133	Yes	Yes	No	Yes
178	Yes	Yes	Yes	Yes
263	Yes	Yes	No	Yes

Table 1. Hough Transform detection at different Lux level (Luminous flux per unit area)

Next, different lighting variations were tested. The Table 1 shows the results of the experiments. The blue area shows the results from when only specular light used, and the red area where specular and ambient light are used.

As seen in Table 1, the Hough Transform starts to break down at higher lux. This collapse is probably due to the image becoming over exposed, making it hard to distinguish what is darker than the background. This hardship is due to the Hough Transform is expecting the circles to be darker than their background. The experiment where ambient light was also applied was able to achieve higher lux before the Hough Transform was no longer able to detect. This is due to ambient light illuminating the node from different angles, allowing the image to be more evenly exposed.

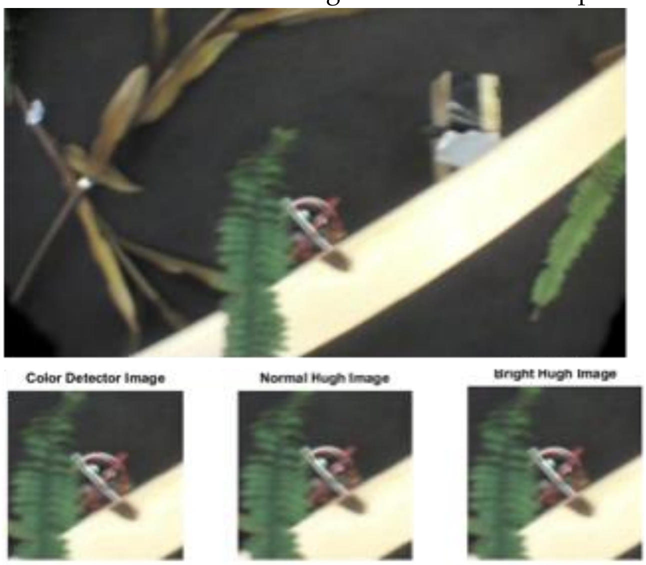

Fig. 18. Original image, color filter, Hough and Bright Hough.

The goal for the obscured images was to be able to detect the no de with 10% occlusion. With the right configuration, we were able to surpass this goal. This allowed the node to be found in much worse cases, sometimes even up to 50% occlusion. The pictures tested included both occlusion by plant-like-materials along with man-made materials, giving both a soft and hard edge between the node and what was occluding it. A hard and soft edge sample can be seen in Fig 18.

The last goal was to find how much a node may be tilted before the circle finding portion of the algorithm would fail to detect it reliably. As seen in Fig. 19 and 20, the algorithm could still detect the node up to 40 degrees of tilt.

Interestingly, one of the circle detectors as shown in Fig. 19 failed to find the node, but found a hand instead. This shows the importance of the arc as a robust feature. In this case, the hand had an arch

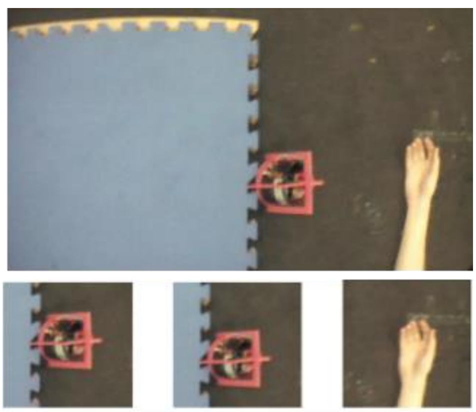

Fig. 19. 35 degree tiled node, color filter, Hough, Bright Hough.

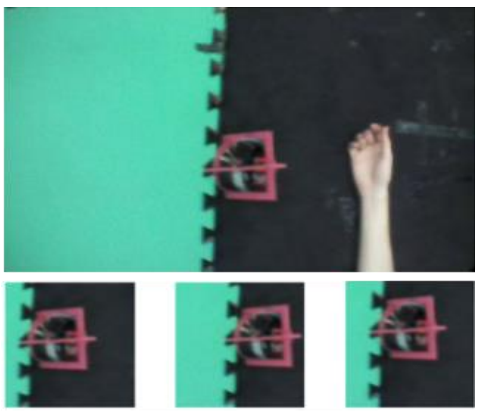

Fig. 20. 40 degree tiled node, color filter, Hough, Bright Hough.

closer to what the algorithm was expecting. The node's arch was too large when looking for an arch that is brighter than the background.

Overall, the algorithm was able to achieve the three main objectives. All three methods were not always able to detect the node, but very often, at least two of the three detect it. Of the 93 images we processed, we had a success rate of 90.32%. A success was defined as two of the three methods detecting the node. The color detection portion of the algorithm was able to find the node in 91 of the 93 test cases. In the two cases were the color detection failed, the other two methods were still able to detect the node.

Fig. 21. Node detected in outdoor environment.

7 Future Work

Bag of features can be implemented to help find the node. This could be implemented on its own, or, in addition to the current system. Bag of features could help find partially occluded nodes better along with being trained to find nodes that are tipped further more reliably.

Flying outdoors was tested a bit. The few images that were taken outside had little clutter, consisting mostly of ground rocks, wood chips, and a few bushes. More testing needs to be done to make sure the algorithm is robust enough to find the nodes in more cluttered, outdoor environments. Additionally, the lighting outdoors can not be easily simulated indoors, such as at dawn or dusk when the light is coming from the sides.

Currently, our code requires exactly one node in the image. This is problematic for when the GPS does not get the drone close enough to the node or if two nodes drift into the same frame. Multiple nodes can be solved by running the image through the process, finding the node, segmenting it out of the image, and processing the image again. Segmenting out the first node would effectively leave one node in the image. Currently, the algorithm will find the center of whatever appears to look the most like a node. If there are no nodes in the image the drone will get mixed results of where the center of the node is. At this point, the drone can ascend to increase its field of view , to guarantee the existence of the node in the image.

Another future modification could be to replace RGB(Red Green Blue) with HSV (Hue Saturation Value) filtering. This will provide more accurate results for variant lighting conditions. HSV is quite effective for a single color, but if the node is drastically faded with time then this method may begins to fail. [11]

A way to help simplify our detection methods would be to remove textures from the image. The textures of the areas where the node is located are known, therefore, with that knowledge, the textures could be removed. Unfortunately, texture would be difficult as it changes with scaling of the image. Using Gabor filters the textures can be grouped together to allow the node with smooth edges extracted from the image.[12].

8 Conclusion

Through this project, we learned that good computer vision algorithms take advantage of several fundamental concepts. One of the main concepts we leveraged was utilizing key features about the node. In our case, we knew that the node was round and was a shade of red. This allowed us to filter out things that did not have enough redness, and then inspect what remained for circles. Additionally, we knew the altitude of the drone, which allowed us to precisely predict the radius of the target node in the image.

We also looked for robust features. The arcs produced by the node while it was slightly tilted were close enough to circles that they could still be picked up by Hough Transform. Finally, we learned that implementing multiple search techniques, and then comparing the results can lead to a more accurate conclusion. This means that if a couple of methods find the node, but one fails, it can still be considered a success as two of the three were able to detect it.

The limitations found through our experiments showed with more light, detection with higher node occlusion is possible. After the node became over 50% occluded, we were unable to reliably detect it. This is because there was not enough of the node exposed to provide an arch larger enough for the Hough Transform or enough color for the color detector to find. Additionally, if the light became too bright, greater than 125 lux with no ambient light, we lose the ability to detect the node reliably. This is because the image would start to be overexposed making it hard to detect color or circles. The area in which we are able to reliably find the node can be defined as the area below the curve in Fig. 22.

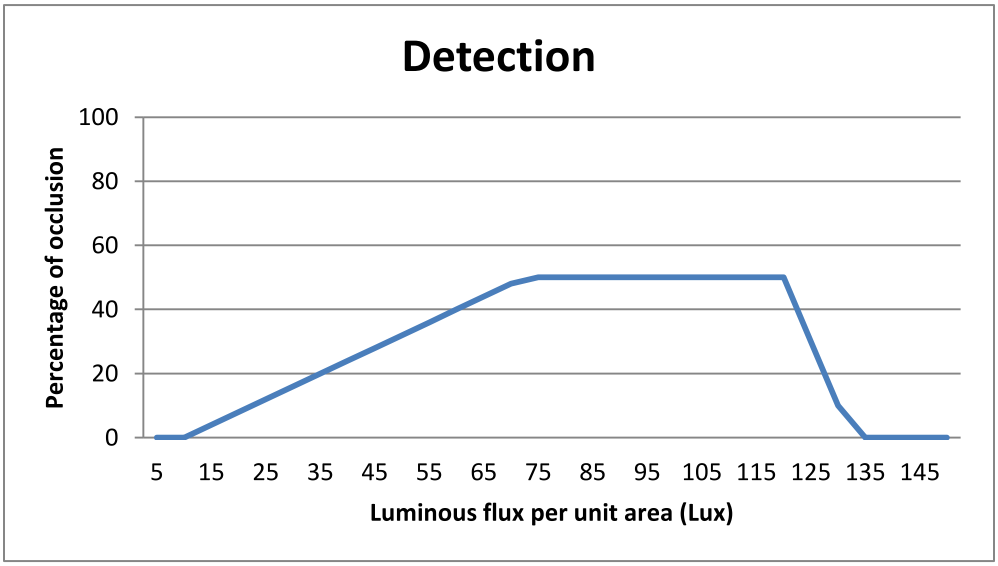

Fig. 22. The area below the curve is the area of optimal node detection (using our current camera).

REFERENCES

[1] A. Samal Dr., "Computer Vision course material" CSCE 473/873, Summer 2015.
[2] A. Mittleider Dr., "Experimental Analysis of a UAV-Based Wireless Power Transfer Localization System", 2014
[3] R. Szeliski, Computer Vision: Algorithms and Applications. Microsoft Research, pp. 221-224, 2010.
[4] T. Atherton, D. Kerbyson, "Size invariant circle detection" Elsevier Science B.V, 1999
[5] R. Szeliski, Computer Vision: Algorithms and Applications. Microsoft Research, pp. 281-283, 2010.
[6] B. Griffin and C. Detweiler, "Resonant Wireless Power Transfer to Ground Sensors from a UAV". Proceedings of IEEE International Conference on Robotics and Automation (ICRA), St. Paul, Minnesota, 2012.
[7] X. Chen, L. Lu, Y. Gao, "A New Concentric Circle Detection Method Based on Hough Transform" The 7th International Conference on Computer Science & Education (ICCSE 2012) July 14-17, 2012.
[8] S. Guo, X. Zhang, F. Zhang. 2006. "Adaptive Randomized Hough Transform For Circle Detection Using Moving Window" Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006.
[9] X. Huang, T. Sasaki, H. Hashimoto, F. Inoue. 2010. "Circle Detection and Fitting Based Positioning System Using Laser Range Finder" SI International 2010.
[10] S. Grigorescu, N. Petkov, P. Kruizinga. "Comparison of Texture Features Based on Gabor Filters" IEEE Transactions on image processing, VOL. 11, NO. 10, October 2002
[11] N. Drakos. "The HSV Colorspace" University of Leeds, March 2, 1998
[12] L. Fei-Fei, R. Fergus, P. Perona. "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories." Workshop on Generative-Model Based Vision. 2004

ACKNOWLEDGMENT

We would like to thank Dr. Samal for guidance and support during this project.

We would like to thank Dr. Detweiler and NIMBUS Lab personnel for assistance and use of equipment.

We would like to thank our colleague Lichao Sun (James) for his assistance and willingness to work with us on future work ideas related to our project.

- We also like to thank **Alexander Renken** for working on basic hough transform algorithm (4 hours), cross detection (2 hours) normal and bright hough (1.5 hours), tilt experiments (1.5 hours), occlusion experiments (3 hours), presentation (12 hours), writing report (16 hours). (Total: 40 hours)
- We also like to thank **Najeeb Najeeb** for working on programming UAV (1.5 hours), gathering input video (1 hour), converting video to images (3 hours), conducting indoor clutter (1 hour), light (2 hour), occlusion (3 hours), tilt (1.5 hour), and outdoor experiments (1 hour), result analysis (code and evaluation) (4 hours) presentation (12 hours), writing report (16 hours). (Total: 48 hours).
- We also like to thank **Trevor Craig** for working on normal and bright hough (1 hour), improving the hough transform to support occlusion (2 hours) and tilt (1.5 hour) and adding color filter (4 hours), output image generation (1.5 hours), light experiments (2 hours), occlusion experiments (3 hours), presentation (12 hours), writing report (16 hours). (Total: 43 hours)