Autonomous wind turbine inspection using a quadrotor

Autonomous Wind Turbine Inspection using a Quadrotor

Weibin Gu[1], Dewen Hu[3], Liang Cheng[3], Yabing Cao[3], Alessandro Rizzo[2], Kimon P. Valavanis[1]

***Abstract*— There has been explosive growth of wind farm installations in recent years due to the fact that wind energy is gaining worldwide popularity. However, the maintenance of these offshore or onshore wind turbines, especially in remote areas, remains a challenging task. In this work, vision-based autonomous wind turbine inspection using a quadrotor is designed based on realistic assumptions. Both simulation and Hardware-In-the-Loop (HIL) testing results have shown the effectiveness of the proposed approach.**

I. INTRODUCTION

Wind energy, as a clean and renewable alternative to the burning of fossil fuels, is gaining worldwide popularity. Consequently, there has been explosive growth of wind farm installations all over the world in recent years. However, the maintenance of these offshore or onshore wind turbines, especially in remote areas, remains a challenging task. The traditional ways for wind turbine inspection normally require a professional team who will use rope climbing or ground equipment to perform inspection manually. Nonetheless, it can be extremely risky for the maintenance crews to do rope-based inspection and it is often inefficient to use telephotography since small cracks and damages are sometimes undetectable to the naked eye. In addition, traditional inspection also requires high operation costs and long downtime.

To solve these issues of manned inspection, approaches that involves mechanism and robots have been explored such as rope-assisted repair robot [1] and magnetic climbing robot [2]. Despite the promising experimental results, direct contact with the blades or other mechanical components of wind turbine is unavoidable during inspection, which might give rise to additional undesired damages. Unmanned Aerial Vehicles (UAVs), armed with a high-resolution camera, infrared camera or other sensors, are good alternatives that prevent direct contact.

Recent advances in both hardware and software have made possible the rapid development of UAVs for civil applications such as precision agriculture, infrastructure inspection, search and rescue, and surveillance [3]. Thanks to the high mobility, easy deployment and low maintenance cost, UAV-based inspection has great potential for the maintenance of wind turbines. With the use of quadrotors, the safety of the operators can be guaranteed and less downtime for inspection is needed. Specifically, inspection time can be shortened from 3 or 4 hours by manned inspection to only about 30 minutes by using UAV-based solutions. Moreover, for those wind turbines settled in the harsh environment, regular inspection can be performed by acquiring high resolution images and/or videos through onboard cameras for accurate crack and damage diagnosis. Hence, it is straightforward to see that UAV-based inspection is an efficient and cost-effective approach that overwhelms the traditional inspection methods.

There have been a few studies focusing on UAV-based autonomous inspection for wind turbine at standstill in the last decade such as [4]–[7]. The main idea is to navigate the UAV along a series of fixed inspection waypoints generated by vision-based line detection algorithms like Hough transform based on the simplified blade geometry (i.e., blades are regarded as line segments). However, blades are usually blended in reality due to plasticity. When inspecting at a close distance with respect to the blade (e.g., 10m), ignoring such nonlinear factor would lose the blade in sight (especially near the tip) if following a fixed inspection path. Hence, it is important to take into account the curvature of the blade from practical perspectives. To achieve better detection performance, deep learning techniques have also been used such as Convolutional Neural Network (CNN) [8] and Mask R-CNN [9]. Nonetheless, there does not exist a work specifically treating the problem of determining the front side of target wind turbine, which is essential for the inspection where UAV is required to fly along the blade because the inspection normally starts at the front side under such circumstance.

In this work, we aim to solve the stated issues by focusing on the design of the high-level control of a quadrotor to perform autonomous wind turbine inspection. As such, it is assumed that the low-level flight control is well handled by a pre-designed controller (e.g., $\mathcal{L}_1$ adaptive controller as presented in [10]). Our main contributions are as follows:

1) We design and elaborate a comprehensive inspection procedure based on a series of realistic settings of wind turbine at standstill which include: (i) Blended shape close to the tip of the blade is considered; (ii) Both leading and trailing edge of the blade are required to be inspected; (iii) Blade pitch angle (see Fig. 2) is equal to $90\,\mathrm{deg}$ (i.e., blades are perpendicular to the plane of blade rotation) when the wind turbine is idle.
2) We propose a vision-based detection system to determine the front side of target wind turbine as well as achieving more robust and accurate performance

[1]Weibin Gu and Kimon P. Valavanis are with the Department of Electrical and Computer Engineering, University of Denver, Colorado, USA {Weibin.Gu, Kimon.Valavanis}@du.edu

[2]Alessandro Rizzo is with the Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy alessandro.rizzo@polito.it

[3]Dewen Hu, Liang Cheng, and Yabing Cao are with the Department of Research and Development, Shanghai FOIA Co. LTD, Shanghai, China {dewen.hu, liang.cheng, yabing.cao}@foiadrone.com

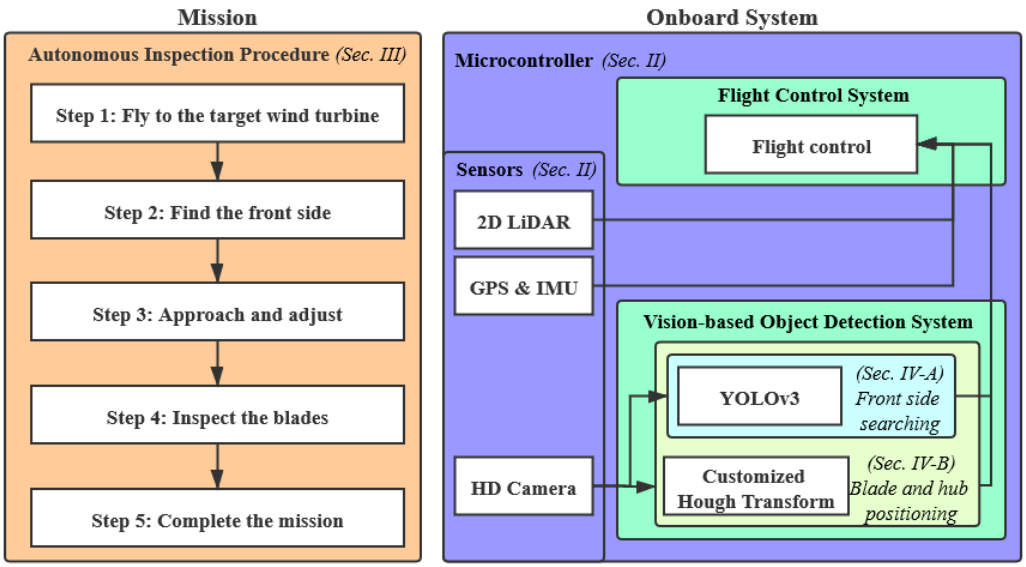

Fig. 1. Overall structure: Mission procedure, onboard hardware, and control system are indicated in orange, blue, and cyan, respectively. The section number in italic type specifies where they are detailed in this paper.

by integrating deep learning technique and traditional computer vision method.

The remainder of this paper is organized as follows. Section II describes the employed UAV platform for autonomous wind turbine inspection. Section III elaborates the thorough inspection procedure designed with the considerations of realistic settings. Vision-based system for wind turbine object detection is detailed in Section IV. Simulation and Hardware-In-the-Loop (HIL) testing results are demonstrated in Section V to show the effectiveness of our approach. Finally, conclusions and future work are given in Section VI. The overall paper structure is depicted in Fig. 1.

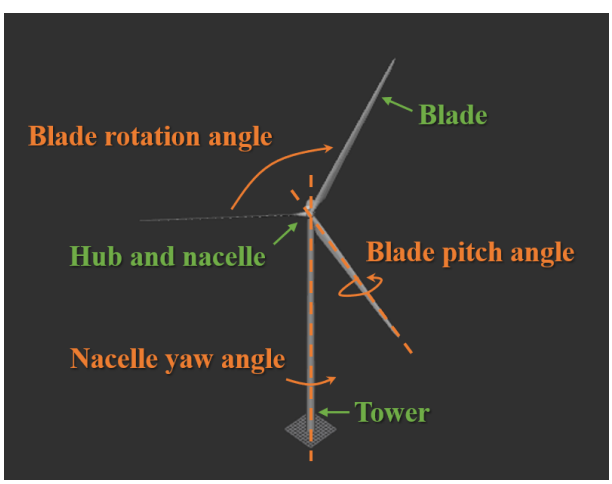

Fig. 2. Model of an offshore horizontal axis wind turbine studied in this work: (i) Main mechanical components: tower, hub, nacelle, and blades (indicated in green). (ii) 3 Degree-of-Freedom (DoF): nacelle yaw angle, blade rotation angle, and blade pitch angle (indicated in orange).

II. UAV PLATFORM

A quadrotor, being equipped with a quad-core ARM processor and a 256-core NVIDIA GPU, is used for wind turbine inspection in this work. Thanks to their exceptional speed and power efficiency, complex control and vision algorithms can be implemented in real time. As common sensory modules, GPS and IMU are included onboard to provide accurate position and attitude information through data fusion algorithm.

To perform vision-based inspection, the quadrotor carries a high-resolution camera mounted in gimbal as payload. This camera is used not only for generating control commands by the vision-based detection system, but also for recording the blades. High quality photos and/or videos acquired by this camera will be stored on a MicroSD card.

We also use a 2D LiDAR based on Time of Flight (ToF) mounted in gimbal. This is to provide obtacle avoidance capability when GPS measurement is unreliable, for instance, when flying closely under the blade. More importantly, it enables the UAV to navigate in a constant inspection distance with respect to the blade considering the non-negligible curvature, particularly closing to the tip of the blade.

III. AUTONOMOUS INSPECTION PROCEDURE

The design of autonomous inspection is based on the following assumptions: (i) Wind turbine is at standstill with fixed nacelle yaw angle and blade rotation angle, and blade pitch angle of $90\,\mathrm{deg}$. (ii) Twisted blades as in real world are taken into account. (iii) Wind turbine specifications are known, including blade length and tower height. (iv) Approximate GPS information of hub position of wind turbine is known. In the sequel, we elaborate the inspection procedure that will be autonomously conducted by the UAV platform as

described in Section II. The inspection trajectory is depicted as shown in Fig. 3.

Step 1: Fly to the target wind turbine

This step comprises take-off and flying to the target wind turbine. First, UAV will be commanded to ascend to the height of the hub (which is known a-priori from our assumptions). Then, it will head to the target wind turbine horizontally and stop at a safe distance to the hub. The safe distance (d) is determined by the blade length (l) and the Angle Of View (AOV) of payload camera as formulated as follows:

$$\tan\left(\frac{1}{2}\text{AOV}\right) = \frac{l}{d}, \tag{1}$$

which leads to the lookup diagram as illustrated in Fig. 5. If the distance between quadrotor and wind turbine is less than this safe distance, collisions might occur in Step 2 and the performance of vision-based system might be degraded due to the incomplete blades appeared in the camera image (see explanation in Fig. 4). Note that external disturbances are not considered in Eq. (1) as it only aims to provide an approximated value for implementation.

Step 2: Find the front side

Previous studies have provided methods to calculate the yaw difference between UAV and wind turbine by using pinhole camera model. However, such methods rely heavily on the detection of a pre-determined feature, for instance, the gap between the hub center and tower [5], which can be barely obtained at a high accuracy in practice. More importantly, no studies have considered how to distinguish the front and back side of wind turbine, which is indispensable because both of which have situations where yaw difference is zero.

To solve these issues, we designed a control logic that commands the UAV to do horizontal circular motion with respect to the hub. The core idea is to compare the relative position of hub and nacelle detected by the vision-based detection system. That is, if the center point of the bounding box of detected hub is on the left side of that of the detected nacelle, we can judge that the shortest way that UAV gets to the front side of the wind turbine along the circular path is in the clockwise direction. Only when the hub is detected and not the nacelle will UAV stop the circular motion and proceed with the next step as it considers the front side is found.

Step 3: Approach and adjust

After finding the front side of wind turbine, vision-based detection system will retrieve the blade rotation angle from the detection results of the blades and estimate the hub position coordinates in the camera image by calculating the intersection point of detected blades. Then, UAV will approach the wind turbine while keeping the hub at the center of the image by adjusting its position based on the real-time hub detection. It will eventually stop at an inspection distance (in our case, set to 8m to obtain high quality pictures) to the hub measured by the 2D LiDAR.

Step 4: Inspect the blades

To perform a complete inspection of each blade, UAV will navigate along the blade twice, i.e., one at its front side while the other at its back side, to fully inspect both leading and trailing edge. Since all the blades have a pitch angle of $90\,\text{deg}$, the inspection trajectories will be designed on their sides rather than at their front as suggested in the previous studies. For each blade, the inspection can be divided into the following three steps:

1) Pan the blade: In order to take high quality photos to facilitate crack and damage analysis, the camera should be directed to the blade with a $45\,\text{deg}$ angle when panning the blade, i.e., $\angle ABC = 45\,\text{deg}$ in Fig. 6. To this end, we need to calculate the expression of line of sight vector $\mathbf{p_c}$ of the camera in the fixed body frame $\mathbf{R_{fb}}$. Note that $\mathbf{R_{fb}}$ is defined from UAV body frame with constraints on x-axis always perpendicular to the blade rotation plane. With blade rotation angle denoted by θ (clockwise positive viewed from the front side as indicated in Fig. 6), $\mathbf{p_c}$ can be expressed as follows:

$$\mathbf{p_c} = \begin{cases} (1, \sin(\frac{\pi}{2}+\theta_0), -\cos(\frac{\pi}{2}+\theta_0))^{\mathrm{T}}, & \text{when UAV is} \\ \quad \text{in the counterclockwise direction of the blade;} \\ (1, -\sin(\frac{\pi}{2}+\theta_0), \cos(\frac{\pi}{2}+\theta_0))^{\mathrm{T}}, & \text{when UAV is} \\ \quad \text{in the clockwise direction of the blade;} \end{cases} \tag{2}$$

where θ_0 is equal to θ when the quadrotor is at the front side of the wind turbine; $2\pi - \theta$, otherwise. To avoid landing legs appearing in the image, we decompose $\mathbf{p_c}$ into two separate motion, i.e., quadrotor yaw angle and camera gimbal pitch angle with respect to $\mathbf{R_{fb}}$. Hence, the final pitch and yaw angle can be derived from (2) through $\mathrm{atan2}$ function. Since the purpose of formulation of $\mathbf{R_{fb}}$ is to calculate angles, we do not normalize the vector and its magnitude is not important. It is worth mentioning that these angles are only associated with θ. Hence, we can also control pitch and yaw angle by (2) given techniques to estimate θ in real time for slightly rotating wind turbine.

Throughout the navigation along the blade, 2D LiDAR mounted in the gimbal is used to keep constant inspection distance to the blades while following their curvature. To avoid disturbance (specifically, the point clouds of quadrotor itself), a threshold is set for LiDAR range based on the physical size of the quadrotor. As such, all the point clouds that are within this threshold will be regarded as the components of quadrotor, and hence, be neglected in data processing.

2) Fly around the tip: After panning one side of the blade, the quadrotor will fly to the other side. Moreover, it will fly around the tip to reach the other edge of the blade as shown in Fig. 3 before starting panning the second side. By doing this, it can be ensured that a blade is under a complete inspection in terms of two sides (front and back) and two edges (leading and trailing).

3) Transition to the next blade: When a single blade has been inspected, the quadrotor will transit to the next blade. To prevent the risk of collision with the tower (especially at the back side of wind turbine), we design a safe-

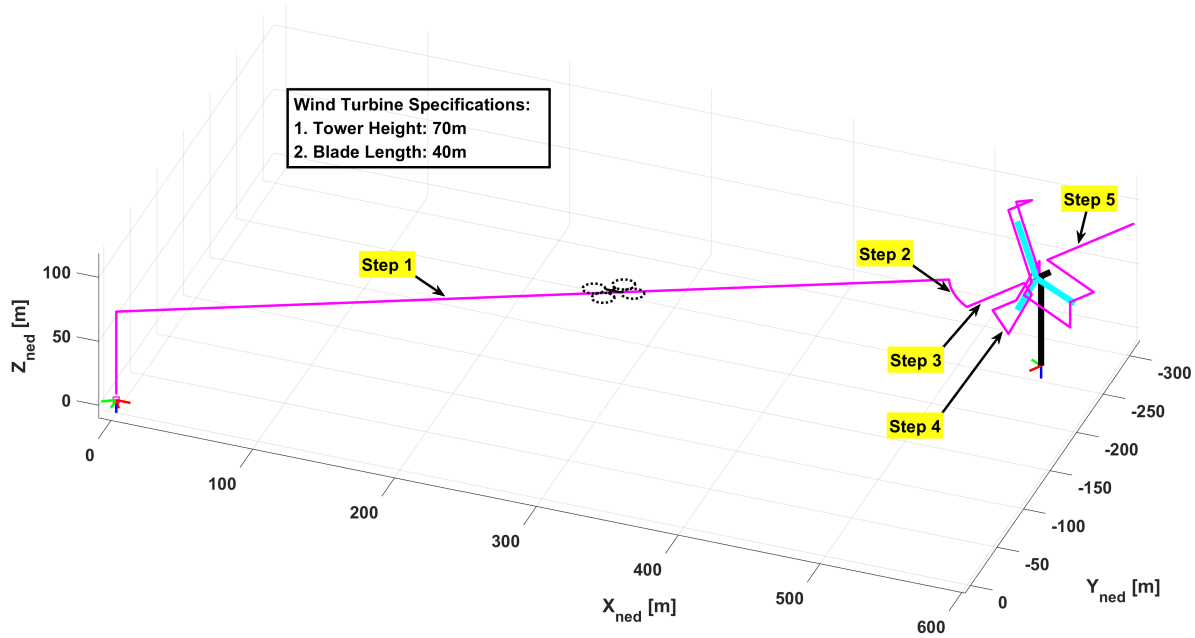

Fig. 3. Autonomous wind turbine inspection. The simplified 3D model of wind turbine has a tower of 70m and blades of 40m. The trajectory of the complete inspection procedure is shown in the local North-East-Down (NED) reference frame in magenta.

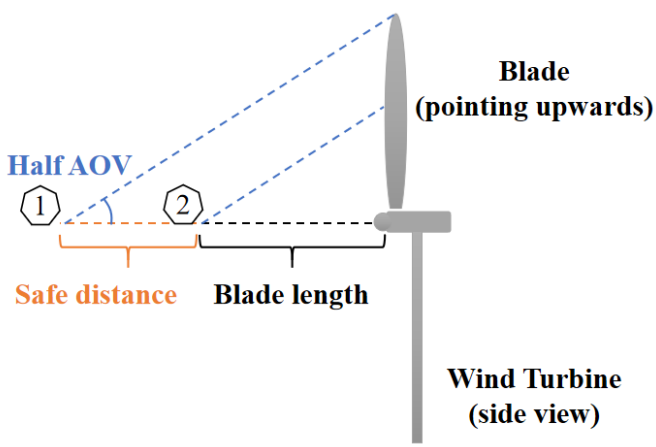

Fig. 4. Illustration of Eq. (1). Safe distance is designed to guarantee the complete view of blade in the vision. For instance, camera is able to see the full blade when UAV is at position 1. While at position 2, only portion of the blade is shown in the image, which might affect the performance of blade detection by the vision-based detection system later.

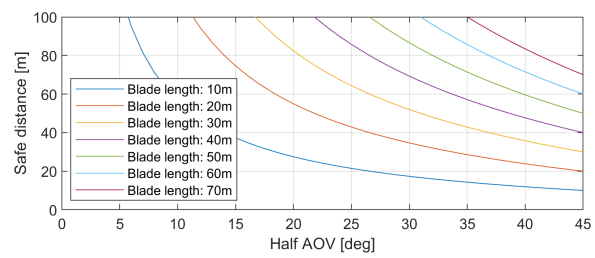

Fig. 5. Lookup diagram of safe distance for different blade lengths and camera AOVs

oriented inspection sequence depending on the blade position as illustrated in Fig. 7. By vision-based detection system (customized Hough transform in Section IV-B), if a blade is found to be located in the region of rotation angle $[\frac{2\pi}{3}, \pi]$ (or $[\pi, \frac{4\pi}{3}]$), then the quadrotor will inspect the blades in a counterclockwise (or clockwise) order. Specifically, the inspection sequence follows $1f \rightarrow tp \rightarrow 1b \rightarrow tr \rightarrow 2b \rightarrow tp \rightarrow 2f \rightarrow tr \rightarrow 3f \rightarrow tp \rightarrow 3b$, where Xf, Xb $(X = 1, 2, 3)$ represents the front or back side of blade number X, tp represents flying around the tip, and tr represents transition to the next blade. It should be noted that the transition from the second blade to the last ($2f$ to $3f$) is different from the transition from the first blade to the second ($1b$ to $2b$) as shown in Fig. 7. After panning the front of the second blade, the quadrotor will transit to the side of the third blade that is closer to the tower, rather than the side that is closer to the second blade. This is to guarantee the safety of both quadrotor and wind turbine by keeping the inspection of the last blade at the back side of wind turbine always away from the tower.

Step 5: Complete the mission

Finally, the quadrotor will return to the take-off position or move on to inspect the next wind turbine based on requirements and battery condition.

IV. VISION-BASED OBJECT DETECTION

The functionalities of vision-based detection system proposed in this work are two-folded: (i) It distinguishes the front side of the wind turbine during Step 2 to generate flight control commands for quadrotor to reach the front of the wind turbine. (ii) It retrieves wind turbine geometry information from camera image to generate inspection trajectory and

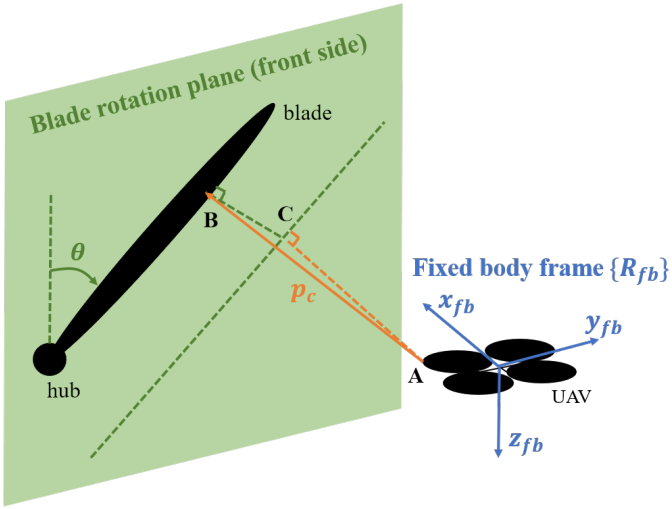

Fig. 6. Derivation of quadrotor yaw angle and camera gimbal pitch angle for a 45 deg blade inspection angle. $\mathbf{p_c}$ represents the vector directing from the camera to the nearest point on the blade. Dashed green line where point C resides represents the desired path of the quadrotor projected on the blade rotation plane. If the desired path is in the position illustrated in the figure with respect to the blade, we say that the quadrotor is in the clockwise direction of the blade (as stated in Eq. (2)). Otherwise, we say that the quadrotor is in the counterclockwise direction of the blade.

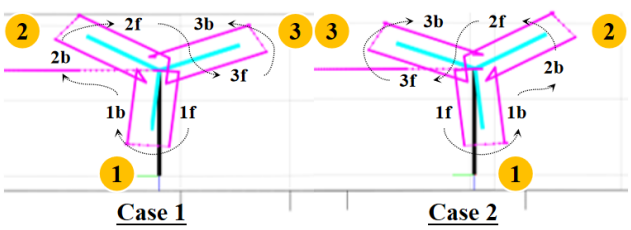

Fig. 7. Safe-oriented blade inspection sequence viewed from the front side of wind turbine. As indicated in numerals, case 1 requires an inspection in clockwise, while case 2 requires an inspection in counterclockwise (both following the sequence starting from $1f$ to $3b$).

locates the hub for adjusting the quadrotor's position in Step 3. In the following, we discuss the two aspects, respectively.

A. Searching for the front side of wind turbine

According to the control logic explained in Section III, the performance of hub and nacelle detection is critical to the application. Considering the complexity of the environment as well as different blade geometry that may appear when quadrotor is flying around the hub, deep learning technique is used to take advantage of its strong learning capability. To achieve real-time detection, YOLOv3, a variation of YOLO-based CNN family of models for object detection, is adopted in our system. On a 256-core NVIDIA GPU, it can processes images of size 416×416 at approximately 2 FPS. Compared to other detection networks such as Fast R-CNN, it can better suffice the real-time requirement.

A similar network architecture of YOLOv3 is employed as presented in [11]. To speed up the inference while guaranteeing good performance, we choose the input size to $416 \times 416 \times 3$ for our network. The training set contains nearly 700 images in which tower, hub, nacelle, and blades are all labelled. As shown in Fig. 8, high-precision detection results are obtained on the test set after 1-day training with CUDA installed. Meanwhile, the detector also shows good robustness with regard to different environment conditions (i.e., light condition and background).

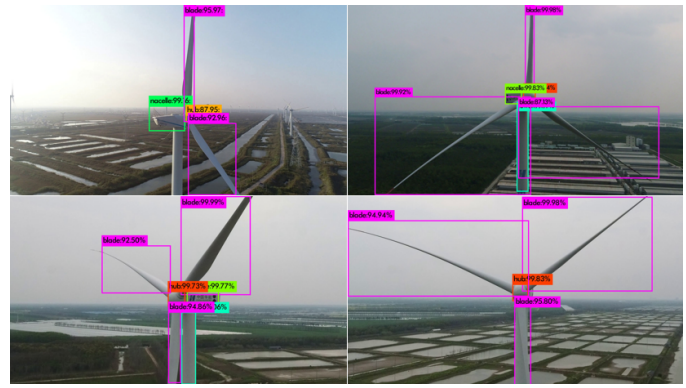

Fig. 8. Detection results by YOLOv3 network of wind turbine with different blade rotation angle and environment conditions: nacelle (green bounding box), hub (orange bounding box), blades (magenta bounding boxes), and tower (cyan bounding box).

B. Blade rotation angle calculation and hub positioning

An integration of deep learning technique and traditional computer vision method is adopted for blade detection and hub positioning. The reason to include traditional method is that blade geometry information can now be exploited as UAV is at the front of the wind turbine and traditional methods do not behave as black boxes as deep learning techniques do. However, limited to the dependency of blade detection, deep learning technique is used for hub positioning later on when blades are out of the image during approaching. The brief working principle of such integration is as follows. For starters, blades are detected by a customized Hough transform algorithm specifically designed for wind turbine inspection. Blade rotation angles are retrieved from the line segments of the detected blades to generate inspection trajectory. Next, the hub coordinate in the image is calculated as the intersection point of the blades. Using this as a-priori information, YOLOv3 network is then in charge for hub detection while UAV is approaching the wind turbine.

1) Customized Hough transform: As a well-known approach of line segment detection, Hough transform cannot always provide satisfactory results considering its computational cost as well as specific requirements of applications. Hence, in this work, we present a customized algorithm with a novel rating system concerning the geometry constraints based on probabilitic Hough transform [12]. The overall logic is shown in Fig. 9. For better understanding, readers are referred to [5] as the explanation of basic and common parts will be omitted here due to the limited space.

The key block in Fig. 9 is "blade grouping and rating", which first groups the line segments that are likely to belong to one blade and then eliminates all the blade candidates

that violate the geometry constraints. The criteria for blade grouping are: (i) The distance between the endpoints are within certain threshold. (ii) The angle difference is within certain threshold. Blade candidates are then calculated by averaging the coordinates of endpoints of line segments considered to be one blade.

Next, blade rating will evaluate all the candidates based upon: (i) The distance between the top of the detected tower and the endpoint that is close to the top. (ii) Times of violation on blade geometry (i.e., $120\,\mathrm{deg}$ between each pair of blades). A normalized score will be assigned on these two aspects. For example, the score of a blade on (ii) is calculated as the number of violation it made over the maximum times of violation of all the blades. A final score of each blade is then obtained through a complementary filter of these two scores. The first round of rating ends up with keeping the three candidates with the lowest final score. Afterwards, a second rating on violation is conducted on the remaining candidates. This is necessary to eliminate those false positives that have passed the first round of rating by taking advantage of distance. Through our simulation (see Fig. 10), it is found that this second round of rating makes a big difference as it can greatly enhance the robustness of the detector. Issues such as horizon line that passes through or is close to the hub (depicted in pink in Fig. 10) can be neatly resolved using the customized Hough transform.

When the number of eligible blade candidates is equal to 2, ”blade autofilling” block will automatically calculate the rotation angle of the last blade subject to blade geometry constraint.

2) YOLOv3: The reason to use YOLOv3 network is because the detection performance of blades by Hough will be degraded if the quadrotor is within a certain distance to the wind turbine, leading to the situation where only partial of the blades can be seen by the camera. Also recall that this is exactly the reason why we introduce the concept of “safe distance” in Step 1 of Section III. The YOLOv3 network used here is the same one as shown in Section IV-A. Note that images of different size of hub are added into the training set. When the hub coordinate is returned by Hough, YOLOv3 network will compare it with the center of the bounding box of its detected hub. If the distance between them is within certain threshold, then YOLOv3 network will be in charge of hub positioning and this coordinate will be recorded as a-priori information for the next detection during UAV approaching the hub.

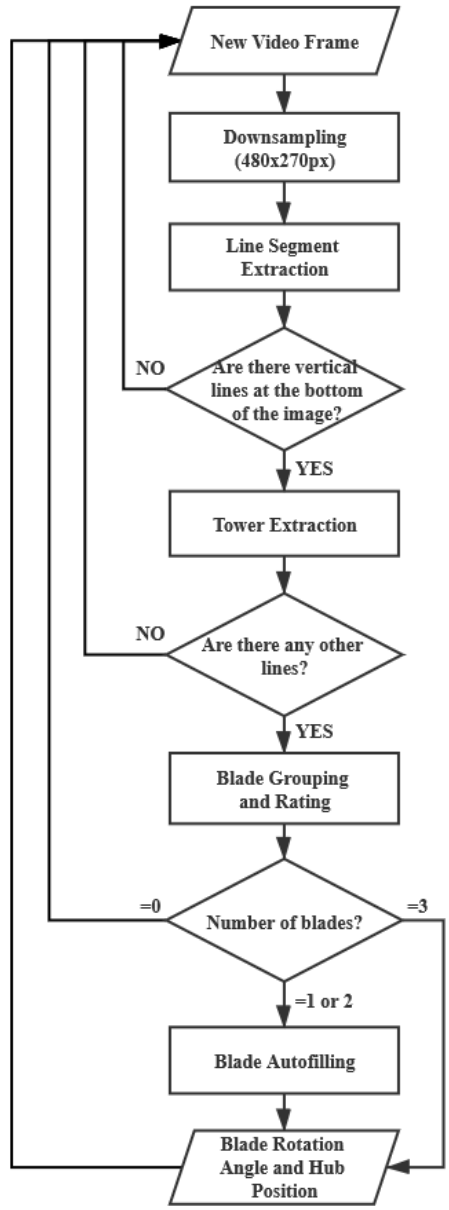

Fig. 9. Flow chart of customized Hough transform

V. SIMULATION

A. MATLAB simulation

To verify the designed strategy of autonomous wind turbine inspection, we first performed simulation of the designed system (vision-based detection system excluded) in MATLAB environment without including any hardware. The UAV trajectory result is shown in Fig. 3 where disturbances like wind gusts and sensor noise are not considered. Note that the simplified 3D model is illustrated only for visualization purposes. Since vision-based detection system is not included in the simulation, all the information needed for the inspection task (e.g., blade rotation angle) is specified explicitly in the program. From MATLAB simulation, we can quickly check the feasibility of the designed inspection path as it provides an insight on the potential collisions with the tower.

B. Hardware-In-the-Loop (HIL) testing

To perform more realistic simulation, we implemented all the control algorithms (also a low-level controller which is not presented in this paper) in C++ on a microcontroller with a quad-core ARM processor and a 256-core NVIDIA GPU (running Ubuntu 16.04 LTS). Then, we connected it to a quadrotor attached with a high-resolution camera to retrieve simulated onboard sensor data for flight control system and

Fig. 10. Detection results by customized Hough transform with different environment conditions: tower (in green), blades (in cyan), hub (in blue), and false positives (in pink).

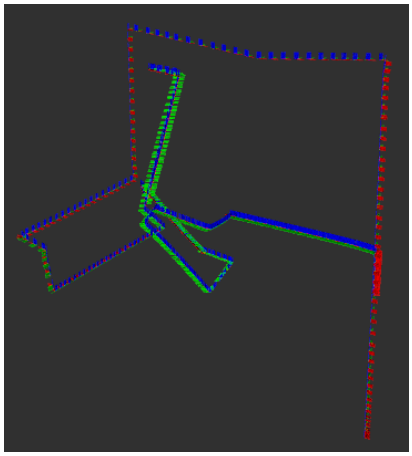

Fig. 11. Simulated inspection trajectory denoted by axes (RBG for x, y, and z axis in UAV body frame, respectively) displayed by *rviz* package.

real-time images of a wind turbine toy model for vision-based object detection system. Different from MATLAB simulation, information such as blade rotation angle can be directly obtained from the vision-based object detection system. The complete inspection trajectory is displayed by *rviz* package offered by ROS as shown in Fig. 11.

VI. CONCLUSIONS

In this work, we have presented a comprehensive design of autonomous wind turbine inspection by using a quadrotor armed with a high-resolution camera and a 2D LiDAR. Specifically, the design is built on top of realistic engineering considerations. Moreover, a vision-based detection system is proposed for wind turbine object detection, which integrates a YOLOv3 network and a customized Hough transform algorithm. Both MATLAB simulation and HIL testing have shown satisfactory results of the design. As future work, we will conduct experimental field tests to validate the effectiveness and robustness of the system. Furthermore, it is of our great interest to investigate autonomous inspection strategy for slightly rotating wind turbine based upon our current studies.

ACKNOWLEDGMENT

This work is partially supported by an NSF Grant, CMMI-DCSD-1728454, Compagnia di San Paolo, and by an Amazon Research Award granted to Dr. A. Rizzo. We also thank Shanghai FOIA, Co. for support.

REFERENCES

[1] S. Hayashi, T. Takei, K. Hamamura, S. Ito, D. Kanawa, E. Imanishi, and Y. Yamauchi, "Moving mechanism for a wind turbine blade inspection and repair robot," *2017 IEEE/SICE International Symposium on System Integration (SII)*, pp. 270–275, 2017.

[2] A. Sahbel, A. Abbas, and T. P. Sattar, "System design and implementation of wall climbing robot for wind turbine blade inspection," *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, pp. 242–247, 2019.

[3] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. M. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[4] S. Høglund, "Autonomous inspection of wind turbines and buildings using an uav," 2014.

[5] M. Stokkeland, K. Klausen, and T. A. Johansen, "Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection," *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 998–1007, 2015.

[6] B. E. Schafer, D. Picchi, T. Engelhardt, and D. Abel, "Multicopter unmanned aerial vehicle for automated inspection of wind turbines," *2016 24th Mediterranean Conference on Control and Automation (MED)*, pp. 244–249, 2016.

[7] R. Parlange, "Vision-based autonomous navigation for wind turbine inspection using an unmanned aerial vehicle," 2019.

[8] O. Moolan-Feroze, K. Karachalios, D. N. Nikolaidis, and A. Calway, "Improving drone localisation around wind turbines using monocular model-based tracking," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7713–7719, 2019.

[9] H. Guo, Q. Cui, J. Wang, X. Fang, W. Yang, and Z.-R. Li, "Detecting and positioning of wind turbine blade tips for uav-based automatic inspection," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 1374–1377, 2019.

[10] R. A. S. Fernández, S. Domínguez, and P. Campoy, "L1 adaptive control for wind gust rejection in quad-rotor uav wind turbine inspection," *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1840–1849, 2017.

[11] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.

[12] J. E. S. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform," *Computer Vision and Image Understanding*, vol. 78, pp. 119–137, 2000.