Vision-based Autonomous Navigation for Wind Turbine Inspection using an Unmanned Aerial Vehicle

R. Parlange*[1], J. Martinez-Carranza[1,2], L.E. Sucar[1], and B. Ren†[3]

[1]Instituto Nacional de Astrofísica, Óptica y Electrónica , Luis Enrique Erro 1, Sta. María Tonantzintla, CP 72840, Puebla
[2]University of Bristol, Bristol, UK, BS8 1UB
[3]Texas Tech University, 2500 Broadway, Lubbock, TX 79409

ABSTRACT

Wind turbines require periodic inspection and maintenance to ensure good performance and a prolonged lifetime. Traditionally, inspection involves the risk of a person falling while climbing down from the platform. Trying to eliminate this risk, Unmanned Aerial Vehicles (UAVs) have been controlled by operators to inspect the structure while taking pictures and video. In contrast, we propose an autonomous UAV system that is able to locate itself and build a map of its environment using visual SLAM. Perception of static rotor blades is based on a single observation, where the Hough transform for lines is used to detect the position of the hub and the angle of the blades, allowing the path planner to make a backwards projection from the 2D image plane to the 3D scene, establishing a set of waypoints to inspect the surface from a safe distance. Experiments were carried out in a simulated environment and a real setting.

1 INTRODUCTION

Wind turbine inspection has been traditionally performed through simple visual inspection from the ground with a telephoto camera lens or by a person who climbs down from the platform. The former method is time-consuming and the latter puts a human life at risk; while both are restricted by the mobility and field of view of the operator.

Trying to overcome these challenges, a recent approach using Unmanned Aerial Vehicles (UAVs) is being adopted in the industry. In contrast to traditional methods, an UAV offers an increased mobility and a close-up view of the surface of the blades. However, this task requires expert pilots and causes them to experience fatigue quickly. Alternatively, autonomous UAVs are not subject to human tiredness and can follow trajectories in a repeatable manner.

*Department of Computer Science at Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).
Email addresses: {parlange, carranza, esucar}@inaoep.mx
†Department of Mechanical Engineering at Texas Tech University.
Email address: beibei.ren@ttu.edu

Figure 1: Left column: simulated inspection using Gazebo, with the bottom image showing the view of the frontal camera; right column: likewise, inspection carried out on an emulated turbine. https://youtu.be/XZZm345rCRY

Autonomous inspection of wind turbines poses a series of challenges. The first one is the localization of the UAV within its environment. This was approached with a computer vision technique named Simultaneous Localization and Mapping (SLAM), to estimate the pose of the UAV, by extracting features from the monocular input of its camera and anchoring them into a map it builds of its surroundings.

To perform the inspection, the UAV must perceive the wind turbine and determine its position relative to the frame of the UAV. Based on an arbitrary takeoff position, the UAV must detect the hub of the wind turbine and the angle of its blades using line features. Once detected, the path planner establishes an inspection trajectory for the UAV to follow while it takes pictures and video of the surface from a safe distance. This method aims at the automatic acquisition of high quality image data for further evaluation with human expertise.

2 Theoretical Framework

2.1 Localization

SLAM with a monocular camera can only build a map up to a scale factor. Autonomous navigation requires a metric scale for path planning, that takes into account the dimensions of the object of interest and the scene. To address this problem, the RGB-D version of ORB-SLAM2 [1] was used, where the RGB frames are coupled with a synthetic depth map [2]. Figure 2 shows the generation of the metric map using the above mentioned procedure. The detection, planning and autonomous navigation modules are described below.

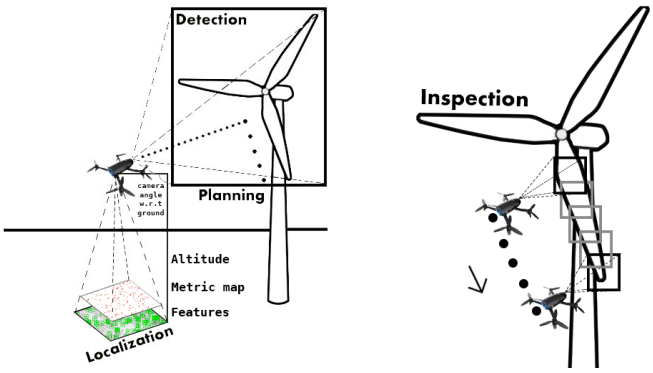


Figure 2: Inspection diagram with three main modules: localization, detection, planning and autonomous inspection.

2.2 Detection

Computer vision tasks such as object recognition require an efficient representation, reducing the amount of data in the image and preserving its visual characteristics and structural information. In order to detect straight lines with the Hough transform, edges must be detected beforehand. Edge detection reduces the amount of data to be processed while conserving the outline of the wind turbine. However, some scenes require further processing to remove image noise. The HSV color model was used to segment the image by separating hue, saturation and value channels.

A Gaussian filter was used to blur and remove noise from the image. The Canny edge detector [3] calculates the directional derivatives with the Sobel operator, which uses two 3x3 kernels, one for horizontal and one for vertical differentiation. These kernels are convoluted with the original image, to obtain the magnitude and direction of the gradient. The output is a binary image with the edges, suitable for straight line detection.

The Hough transform, uses an angle-radius parametrization [4], instead of the original slope-intercept parameters [5] to detect the most prominent straight lines [6] from its camera input, as shown in Figure 3. The position of the hub can be found by looking for the intersection of the blade lines.

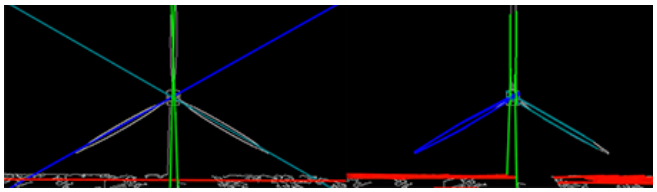


Figure 3: Standard (left) and probabilistic (right) Hough transform for lines, segmented by angle.

2.3 Autonomous navigation

To describe the mathematical relationship between the coordinates of one point in three-dimensional space and its projection onto the image plane, we use the pinhole camera model. With this ideal model, we can also obtain the 3D coordinates from a 2D image if the depth of the image and the intrinsic parameters of the camera are known.

A backward projection is done to recover 3D coordinates from the detection pixel in the 2D image plane, given a known depth [1]. The path planner uses the coordinates from the detection of the hub, along with the angles of the blades to create a flight plan.

Once the path planner has established a set of inspection trajectory points, the UAV must follow these waypoints to approach the blades of the wind turbine and capture photos/video of its surface. In order to accomplish this task, we must find the error between the current pose of the UAV and the target 3D coordinate or setpoint in the control loop.

The UAV approaches the hub of wind turbine using a longitudinal approach. Then it switches to lateral movements to keep the camera oriented perpendicular to the surface of the blades. This process is implemented as a finite state machine with three states: altitude, rotation, and translation.

Altitude. The first state is a proportional (P) controller for the altitude, that minimizes the error between the current z-axis position and the reference, established by the waypoint. Once the error is less than the threshold, the state machine transitions to the rotation state.

Rotation. In this state, the current rotation is compared to a reference rotation matrix that maintains the orientation towards the surface of the blades. A proportional integral (PI) controller is used to reduce the error between the current rotation and the setpoint.

Translation. The last state is a translation controller. The rotation state keeps the orientation of the UAV fixed towards the blade, while a proportional integral (PI) control loop reaches the desired y-axis position.

[1] Depth corresponds to the arbitrary distance from the takeoff position to the hub of the wind turbine.

The controller iterates between these states until the control signals minimize the error on all three axes within a certain threshold. Once this condition is met, the UAV proceeds to follow the next waypoint. Figure 4 shows this navigation controller as part of a state machine of the autonomous inspection procedure described beforehand.

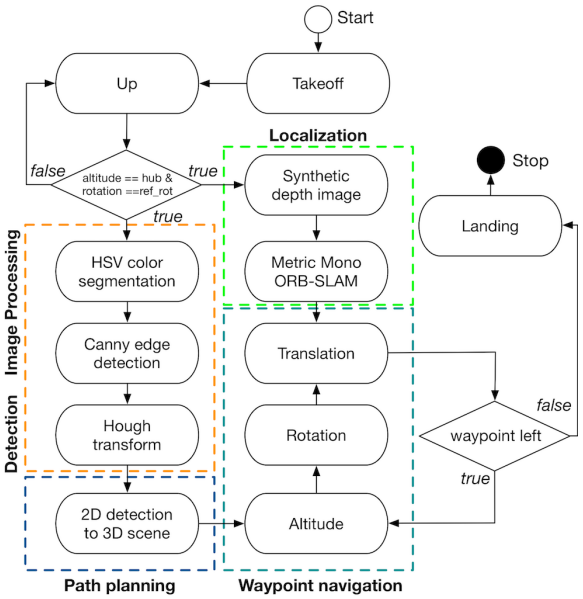


Figure 4: Flowchart of the finite state machine.

3 EXPERIMENTAL FRAMEWORK

3.1 System Overview

The robotic platform used to carry out the experiments was a Parrot Bebop 2 for the physical trials and a Parrot AR.Drone 2.0 in simulation. Even though the UAVs have different Software Development Kits (SDKs), *bebop_autonomy* and *ardrone_autonomy* share topic names, types and coordinate frame conventions for core piloting tasks. These shared characteristics allow the development of algorithms in a simulated environment before testing the effects of the same program in a real setting. As opposed to the AR.Drone 2.0 simulated UAV, the Bebop platform only has a frontal monocular camera. This camera is used for localization, so an additional GoPro camera was attached to the back of the UAV to record the inspection.

The Kinetic distribution of Robot Operating System (ROS) runs on top of an Ubuntu 16.04 Linux operating system. Both *bebop_autonomy* and *ardrone_autonomy* are available as ROS drivers. In order to test and tune our visual navigation system we use Gazebo, a ROS-based robotics simulator. In particular, we use the *tum_simulator*, a implementation of the Gazebo simulator with a model of the AR.Drone 2.0, developed by Hongrong Huang and Jürgen Sturm from the Computer Vision group at the Technical University of Munich (TUM).

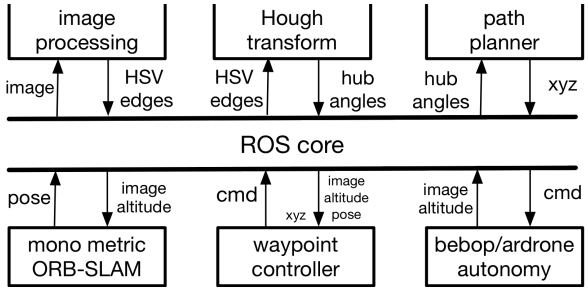


Figure 5: The publisher/subscriber ROS architecture is the same for the simulated environment and real setting scenes.

4 RESULTS

Experiments were carried out in simulation using a model of a scale wind turbine with a height of 10 m to the hub, and in a real setting we used a 3 m tall, emulated scale wind turbine.

4.1 Simulated environment

The waypoint controller follows the inspection trajectory, which is generated by the path planner based on the detection of the hub and the angles of the blades. This control loop uses the continuous localization from the metric SLAM system to estimate the pose of the quadrotor and navigate around the blades of the wind turbine, taking pictures of its surface. Figure 6 shows a 3D plot of the pose estimation from ORB-SLAM against the ground truth from the Gazebo simulator. As can be observed, the inspection trajectories match in all three axes, proving the capacity of the localization system to retrieve an absolute metric scale from its monocular input. The x-axis or depth corresponds to the arbitrary distance to the hub which permits the backwards projection with the pinhole camera model.

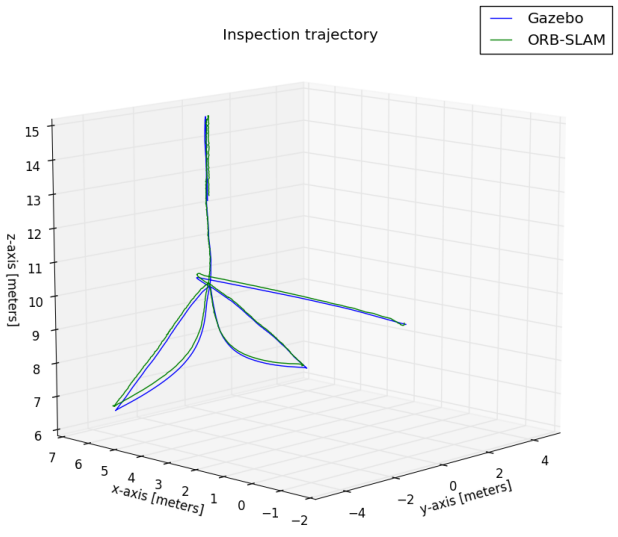


Figure 6: Gazebo ground truth vs. SLAM pose estimation. Root mean squared error, RMSE = 0.344.

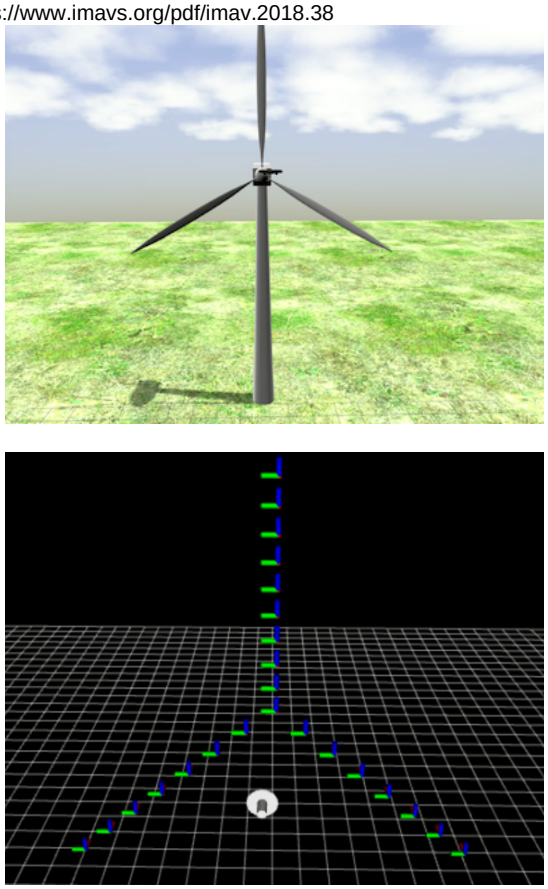


Figure 7: (a) Gazebo simulation. (b) Inspection plan in rviz.

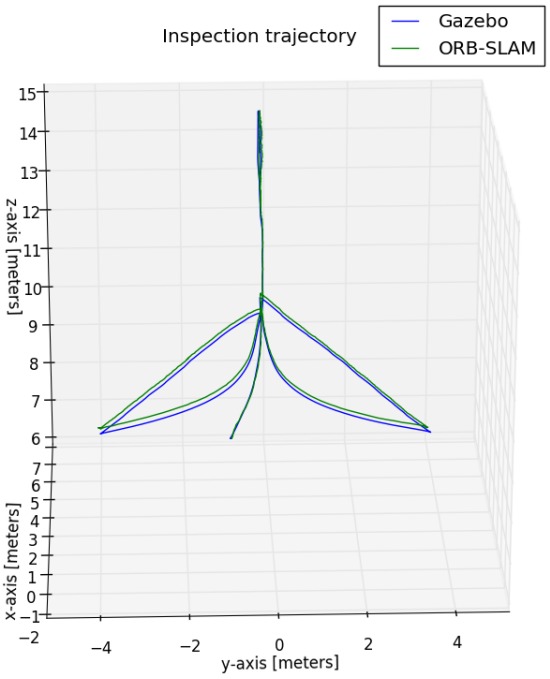
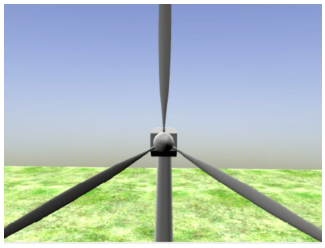
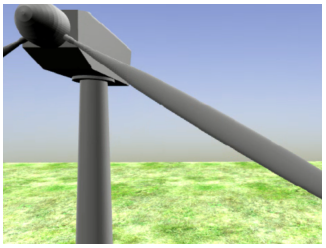


Figure 8: Frontal view of the inspection trajectory.

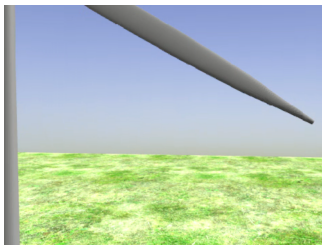
Figure 7(a) shows a Gazebo simulation with the model of the AR.Drone 2.0 inspecting a scale wind turbine with a height of 10 m to the hub or rotor. Figure 7(b) displays the waypoints in the ROS visualization package, rviz. The white arrow depicts the actual pose of the quadrotor estimated by the metric SLAM system. Figure 8 is a frontal view of the trajectory for comparison with the inspection plan above.



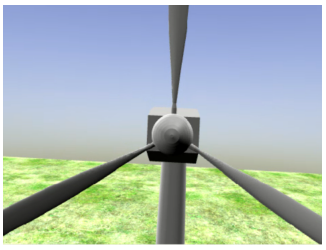
(a) Simulation - Hub.



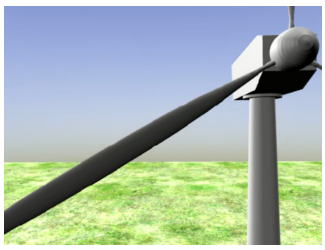
(b) Simulation - Right blade.



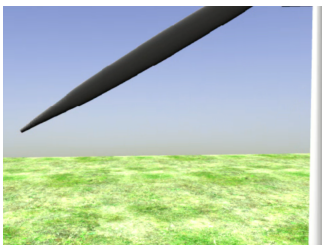
(c) Simulation - Tip of right blade.



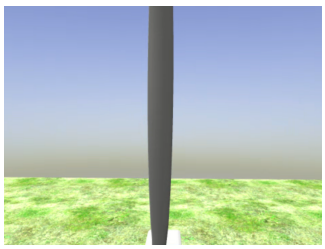
(d) Simulation - Hub.



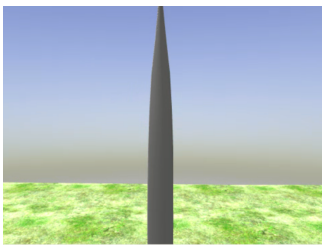
(e) Simulation - Left blade.



(f) Simulation - Tip of left blade.



(g) Simulation - Top blade.



(h) Simulation - Tip of top blade.

Figure 9: Wind turbine inspection in simulation.

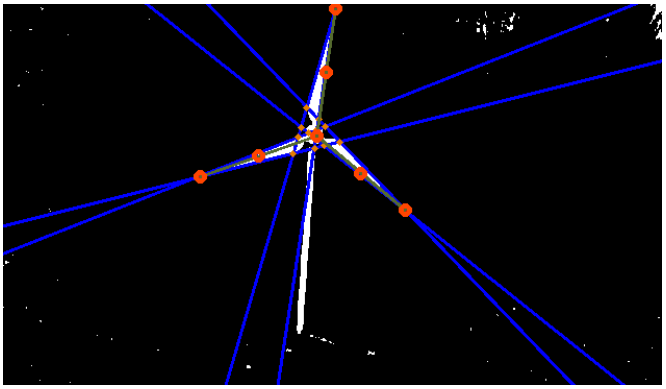
The sequence of pictures in Figure 13 shows captures taken at the generated waypoints. The purpose of this image data is to provide useful information for an inspection expert, to detect damage on the surface of the blades and schedule maintenance for the wind turbine, to sustain its performance.

4.2 Real setting

Due to illumination changes and background image noise found in the proposed scene, alternative color segmentation and detection methods were used.

The CIE L*a*b color space was used to segment the wind turbine from the background. It expresses color in three numerical values, *L* for lightness, *a* for green-red and *b* for blue-yellow components, with a perceptually uniform distribution with respect to human color vision. In this particular scene, the color model allowed the distinction of the white wind turbine from the glow of the sky and the reflection of the light on the leaves. Thus, removing noise and enhancing the detection of the blade lines.

LSD: A Line Segment Detector [7] had better line detection results as it requires no parameter tuning, making it suitable for detection at different lighting conditions. Figure 10(a) shows the six prominent finite line segments, corresponding to the blade lines, extended to infinite lines. Subsequently, the hub is detected by obtaining the centroid of the intersections of these lines. This coordinate is then used by the path planner, which makes the backwards projection to generate the inspection points along the blade lines, as depicted in Figure 10(b).



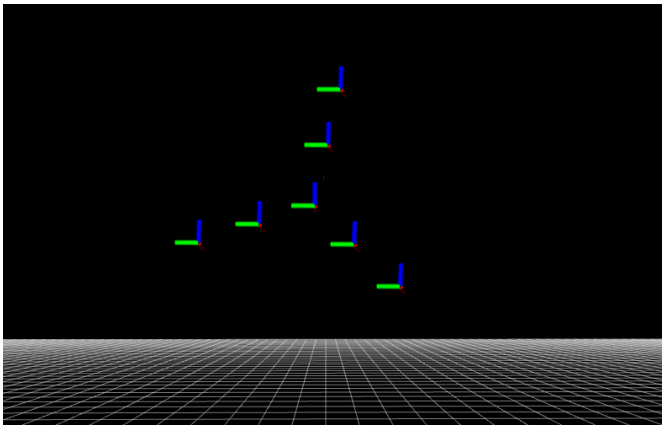


Figure 10: (a) Hub and blade detection over the CIE L*a*b color space. (b) Inspection plan in rviz.

The UAV carried out the autonomous inspection of a 3 meters tall scale wind turbine located at INAOE. It had a GoPro camera attached to its back to record video of the inspection, operating with an inverted orientation.

Figure 11: Frontal camera.

Figure 12: GoPro on the back.

The following photographs show the detection of the wind turbine, proceeded by the approach to the hub and inspection of the blades. The flight plan considers the coordinate of the hub first, an inspection point in the middle of the blade and another one at the end, returning to the hub after each blade. After the execution of the trajectory is completed, the UAV lands.

(a) Real setting - Hub.

(b) Real setting - Left blade.

(c) Real setting - Tip of left blade.

(d) Real setting - Right blade.

(e) Real setting - Tip of Right blade.

(f) Real setting - Top blade.

Figure 13: Wind turbine inspection in a real setting.

5 Conclusions

The increased capacity and expanding installation of wind parks require an inspection method capable of producing high quality and readily available data for inspectors. An UAV offers a close-up view of the surface of the rotor blades, increased safety and better mobility than traditional methods.

Motivated by the above advantages, we have presented a method that does not depend on GPS localization, aiming at carrying out the inspection task in an fully autonomous manner.

Our perception system is capable of detecting the blades and hub in simulation using the Hough transform over the HSV color space, and in a real setting with the Line Segment Detector (LSD) over CIE L*a*b.

This automatic detection generates a plan of waypoints, which are then followed autonomously by the flight controller, thanks to its capacity to locate itself with our metric monocular SLAM system.

Furthermore, we carried out several flight tests with two emulated wind turbines of different sizes, facing strong changes in outdoor illumination and background noise that affected the detection. However, we managed to find the right line detectors, coupled with segmentation over suitable color spaces, for each scene.

As future work, we will develop a detection method that is more robust to illumination changes and can handle different scenarios with minimal parameter tuning. This could offer faster setup times for autonomous wind turbine inspection. We will also carry out tests with full-size wind turbines.

Acknowledgements

This research project has been funded by CONACYT grant 291137 and CONACYT-INEGI project 268528. Special thanks to Oyuki Rojas and Yafeng Wang for their contributions to the experimental design.

References

[1] Raul Mur-Artal and Juan D. Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 10 2017.

[2] L. Oyuki Rojas-Perez and Jose Martinez-Carranza. Metric monocular SLAM and colour segmentation for multiple obstacle avoidance in autonomous flight. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 234–239. IEEE, oct 2017.

[3] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 11 1986.

[4] Richard O. Duda and Peter E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1 1972.

[5] P.V.C. Hough. Method and means for recognizing complex patterns, 3 1960.

[6] Martin Stokkeland. A Computer Vision Approach for Autonomous Wind Turbine Inspection using a Multicopter. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, page 106. Norwegian University of Science and Technology, June 2014.

[7] R.G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, apr 2010.

[8] J Matas, C Galambos, J Kittler, and Y Cvssp. Progressive Probabilistic Hough Transform.

[9] Martin Stokkeland, Kristian Klausen, and Tor A. Johansen. Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection. Master's thesis, 6 2015.

[10] Sondre Høglund. Autonomous Inspection of Wind Turbines and Buildings using an UAV. Master's thesis, Norwegian University of Science and Technology, 2014.

[11] Bjorn E. Schafer, Davide Picchi, Thomas Engelhardt, and Dirk Abel. Multicopter unmanned aerial vehicle for automated inspection of wind turbines. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 244–249. IEEE, jun 2016.

[12] Huiyi Zhang. A feasibility study of a computer-based wind turbine blades surface flaws inspection method.