Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection

Martin Stokkeland, Kristian Klausen, Tor A. Johansen

Abstract— **An autonomous machine vision module for Unmanned Aerial Vehicle (UAV) navigation for inspection of wind turbines is presented. The system estimates the relative position and distance between the UAV and the wind turbine, as well as the position of its blades, in order to support the initial phase of autonomous inspection before the UAV start to move along the blades to acquire pictures. The key algorithms used are Hough transform for detection of the wind turbine tower, hub and blades, as well as the Kalman filter for tracking. Experimental data acquired by a UAV at a wind park is used to evaluate the accuracy and robustness of recognition and navigation. It is found that under the tested conditions, the method gives sufficient accuracy for the task and can execute in real time on a single board computer in the UAV.**

I. Introduction

The commonly used approaches for wind turbine inspection are mainly inspection through telescopic lenses, by lift or climbing (including maintenance and repair), or recently by using remotely piloted UAVs. Modern wind turbines have dimensions of 100 meters or more, so an autonomous or remotely controlled UAV could be able to approach the inspection target closely with high accuracy compared to telescopic photography. Another factor is the cost, which is expected to be significantly smaller with UAVs than manual inspection by climbing. For offshore wind turbines these arguments are even much stronger. For a similar case, inspection of power lines, [1] described several reasons why using a small UAV is preferable.

We focus on multi-rotor UAVs, which can hover and are typically simpler to operate and less influenced by vibrations than helicopters. Their primary limitation for wind turbine inspection is endurance. Such a UAV is typically set up to navigate with an Inertial Measuring Unit (IMU), magnetic compass, and a Global Navigation Satellite System (GNSS), and can carry cameras to be used for navigation and inspection data acquisition. This paper will focus on a machine vision system to autonomously recognize the wind turbine's relative yaw angle and position of blades, which cannot be determined without imaging sensors, manual operations, or a data interface to the wind turbine control system. The latter may be impractical, error-prone, or simply unavailable (e.g. for an independent inspection service company).

Visual navigation is extensively studied for multi-rotors, in particular for navigation in GNSS-denied environments and collision avoidance [2], [3], automatic landing [4], [5], [6], [7], and visual servoing for tracking of features [8], [9], [10], [11], [12], [13], [14], [15], but is not much studied for wind turbine inspection application. Related applications are power line inspection using UAVs where Hough transform is used to detect the power lines, [16], and bridge inspection using visual servoing [17].

The contribution of this paper lies in elaborating on machine vision methods for which the wind turbine can be recognized, and how its features can provide orientation estimates. The paper focuses on the key visual features of the wind turbine, and the use of key algorithms including the Hough-transform and Kalman-filtering. Videos of wind turbines were recorded from a multi-copter UAV at Bessakerfjellet wind park for testing and validation. Algorithms available in the OpenCV library are utilized. Emphasis is placed on keeping computational demand to a level which can be handled by the Single-Board Computer (SBC) which is operating in real time on the UAV. Further details are found in [18].

Center for Autonomous Marine Operations and Systems (AMOS), Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.

Corresponding author: tor.arne.johansen@itk.ntnu.no

The scope of this research is limited to the initial positioning and recognition phase of an inspection. That is, to approach from an initial point where the UAV has arrived in adjacency of the wind turbine using GNSS navigation or manual piloting, to the inspection starting point which is chosen to be directly in front of the hub of the wind turbine. We point to [18], [19] for results on camera-based tracking of position, velocity and attitude relative to the wind-turbine blades during the actual inspection phase.

II. Autonomous navigation

Wind turbines have typically three degrees of freedom for rotation. Their yaw angle about the tower's vertical axis can change to optimize attitude relative to wind directions. Second, their blades rotate about an axis that is aligned with the shaft that transfer power into the hub. Finally, the blade pitch angles may be possible to vary in order to optimize power generation.

It is assumed that for inspection, the wind turbine is at standstill with arbitrary angles of tower yaw angle, blade rotation and pitch. The navigation requirements for wind turbine inspection is divided into the following phases:

1) **Initial positioning.** It is assumed that GNSS and altimeter is used for initial positioning of the UAV, leading to a pre-defined absolute position that is at a given distance to the tower, but with unknown relative yaw-angle to the tower and blades. This distance must be larger than the blade radius to avoid collision risk.
2) **Initial relative positioning and recognition.** Using machine vision, the next step is to position the UAV directly in front of the hub, more specifically that it is located at a given distance near a line that is orthogonal to the disc where the wind turbine blades are, as illustrated in figure 1. In this way, the inspection starts at a known relative position to the hub and disc.
3) **Positioning and tracking for inspection**. In this phase, the UAV moves along the wind turbine hub, blades and tower in a pre-defined motion pattern in order to perform inspection, which typically consist of acquisition of pictures, navigation data and other measurements that will be processed later. During this phase the guidance, navigation and control needs to keep the inspection object at the correct relative distance, attitude and velocity for successful inspection.
4) **Relative terminal positioning.** Moving away from the last inspection position to a given (safe) distance from the wind turbine hub and blades.

In this paper we focus on the 2nd phase, relative initial positioning, that provides an initial configuration for the 3rd and 4th phases. In addition, there might be emergency operating modes that are not considered here.

A. Initial relative positioning and recognition

This phase contains three separate tasks (T1, T2 and T3). Each task is given a priority from T1 at highest to T3 at lowest. Only when the conditions for one task is fulfilled the program moves to the next. Furthermore, if during one task the conditions for a higher prioritized task is invalidated the method reverts back to complete the higher prioritized task. A summary of the procedure is given by the pseudo-code:

while not reached desire distance **do**
  if visual target not in boundary **then** ▷ T1
    if visual target above/below **then**
      move vertically
    if visual target too far left/right **then**
      rotate about the yaw axis
  else if |relative yaw| $>$ tolerance **then** ▷ T2
    move laterally
  else ▷ T3
    move forward

The first task, T1, makes sure that the UAV is facing the target position i.e. the hub of the wind turbine. If the visual target is inside the boundary, the conditions for this task are fulfilled and the next task is initiated.

For the second task, T2, the objective is to achieve the desired relative yaw angle of the wind turbine by flying laterally such that the relative angle is reduced until a certain tolerance is reached. Naturally, by moving laterally one will also experience the wind turbine moving horizontally in the camera frame, albeit slowly compared to the effect of UAV yaw rotation. Consequently, once

the target moves outside the rectangle described in the first task, the first task becomes prioritized and adjusts the attitude to bring the target back inside the boundary. The result of the combined alteration of T1 and T2 is thus a circular motion around the wind turbine, as illustrated in figure 1 by the segments marked T1 and T2. As the relative yaw is reduced past a certain threshold indicated by the green triangle, the relative yaw is deemed as sufficiently small.

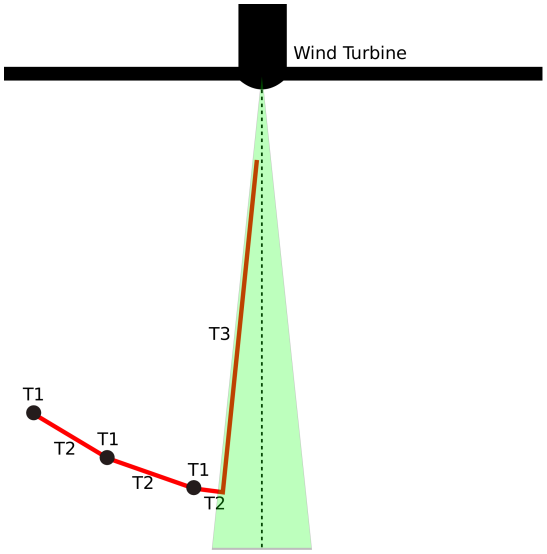


Fig. 1: Top-down perspective showing an example scenario of the navigation path (red line). Indications show when each task is run. The green triangle illustrates the tolerated range for relative yaw.

The third task, T3, commands the UAV to move forward towards the target until the desired distance has been reached. A correct course is ensured and maintained by the former tasks. Should a disturbance cause the UAV to lose its heading or get a too big yaw angle, it reverts to T1 or T2, respectively, to correct the error.

It is assumed that the UAV altitude is kept constant at a suitable level near the top of the tower.

B. Estimation of Distance to Tower

The navigation approach relies on being able to detect when the UAV has reached the desired distance. This can be achieved by applying the pinhole camera model (e.g. [20]). By rearranging this model with respect to a given object's distance to the camera, x^c, one obtains

$$x^c = \frac{f}{y^i} y^c, \quad x^c = \frac{f}{z^i} z^c \tag{1}$$

where f is the focal length, (y^i, z^i) are image coordinates in the image plane, and (x^c, y^c, z^c) are local coordinates in the coordinate frame aligned with the camera mounted on the UAV. Now, let the measured width and height of the given object be given by Δy^i and Δz^i in the image plane. If the actual width or height of the same object is known and given by Δy^c and Δz^c, the distance $d = x^c$ can be estimated by either of the following equations:

$$d = \frac{f}{\Delta y^i} \Delta y^c = \frac{f}{\Delta z^i} \Delta z^c \tag{2}$$

A suitable target object would be the tower having a known width. The tower has easily distinguishable edges that provides a reliable basis for this approach. The tower is usually wider at its base, therefore a specific area of the tower should be targeted. The uppermost area was found to be the best choice because this area persists inside the field of view at all distances.

C. Estimation of Tower Yaw Angle

One can observe from Figure 2 that as the yaw angle increases, two points which are behind each other when viewed from straight ahead, tend to drift apart. In particular, the center of the hub and the top of the wind turbine tower, which are among the easiest points to determine, form the distance δ in the image plane.

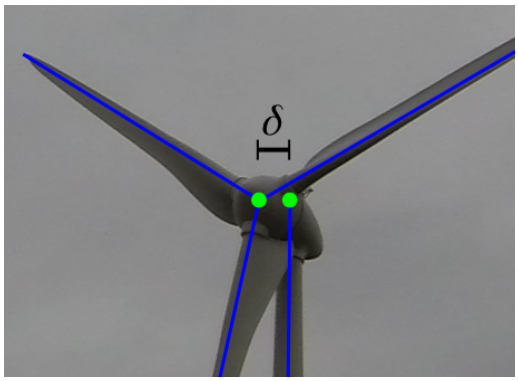


Fig. 2: The gap between the hub center and tower increases as the yaw angle increases.

Consider the pinhole camera model:

$$\delta = \frac{f}{d}|\Delta p^w| \sin\psi \tag{3}$$

where Δp^w is the actual distance between the two points (assume to be a known constant from the geometry of the wind turbine), and ψ is the difference in yaw angle between the wind turbine and camera coordinate frames. Hence,

$$\psi = \arcsin\left(\frac{\delta d}{f|\Delta p^w|}\right) \tag{4}$$

D. Estimation of blade orientation

For a wind turbine with three blades it is known that there is a 120 degree angle beween them. For the initial phase of the positioning in order to start tracking blades, it is necessary to estimate the blades orientation relative to the fixed vertical tower as considered in Section III-D.

III. Machine vision

The camera-based autonomous navigation algorithm runs in a loop where a new iteration is initiated at the arrival of a new image frame, see Figure 3.

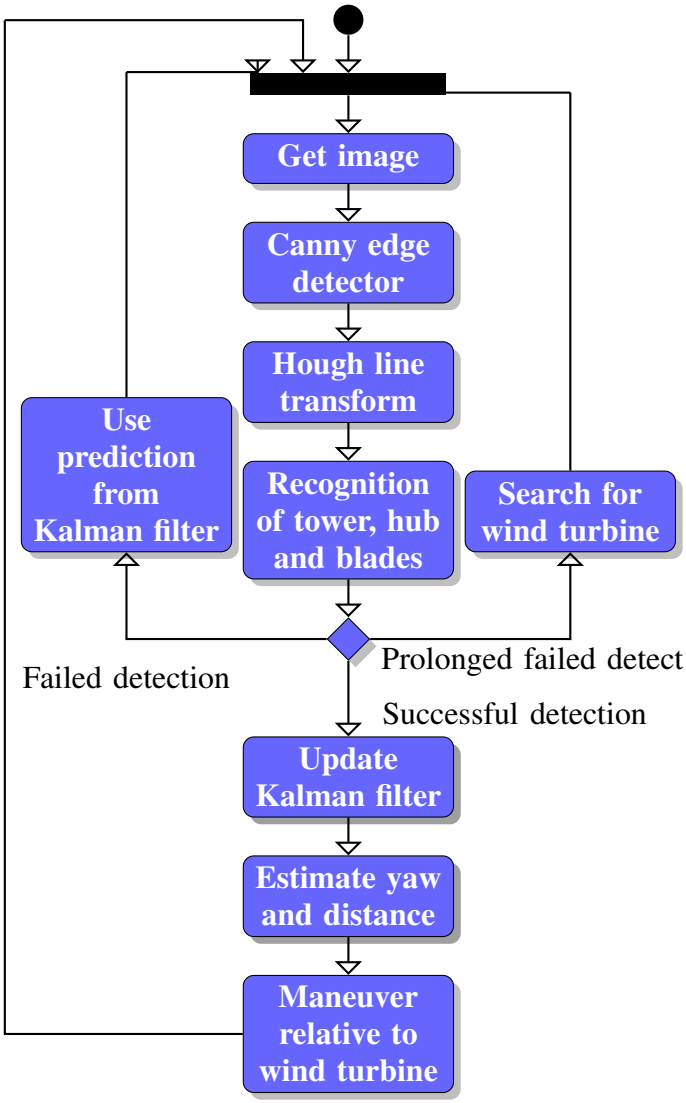


Fig. 3: Activity diagram describing the flow of the machine-vision-based autonomous navigation.

A. Wind turbine visual features

In order to successfully recognize and track the features needed to perform the estimation and navigation functions described above, it is of key importance to identify the characteristics which distinguish the main parts of the wind turbine (tower, blades and hub) from its surroundings.

1) Texture and color: The smooth surface of a wind turbine has virtually no texture, making the wind turbine more distinctively separated from the background. Despite being white in color, a wind turbine can appear in any shade of gray or weak color depending on the lighting conditions and reflections from its environment. However, no particular color tone should be prominent.

2) Geometry: The distinct shape of the wind turbine is of great importance as it possesses many features which can be applied by image analysis algorithms. In particular, the straight edges of the rotor blades and the tower are favorable subjects for edge or line detection algorithms. Furthermore, the radial shape displayed by the tower and the blades provides solid foundation for many approaches. In addition, a similar shape is not expected to appear elsewhere in the environment (except for other wind turbines which might be present).

3) Size: Since GNSS is more practical to use when far away, the UAV is expected to be close to the wind turbine when the algorithm is used. At this range the wind turbine, when in view, should cover a large portion of the image frame and stretch out of the frame when at close distances. Consequently, as long as the UAV is not moving too fast and is facing the wind turbine, it should be clearly visible in the image. The size of the wind turbine as it appears in the image can also serve as an estimate of the distance from the UAV.

4) Movement: It is assumed that the wind turbine is stopped for maintenance so that the blades do not rotate. However, when the UAV moves, the images show a movement of the wind turbine relative to the background.

5) Point-of-view: It will generally be assumed that the UAV is positioned in front or behind of the wind turbine such that the rotor blades or tower usually do not overlap. When directly in the front or back the angle between the blades are 120 degrees. A change of perspective skews the image altering both angles and proportions, thus providing cues for estimating the yaw orientation of the wind turbine.

6) Environment: Under clear weather conditions a blue sky serves as excellent contrast to the wind turbine. In such conditions, shadows might cause challenges. Clouds and gloomy weather leads to less contrast, which might also interfere with detection. Depending on the height and the pitch angle of the camera, the horizon, other wind turbines or straight features may appear at varying height in the image. If they appears too close to the hub, they are expected to interfere with blade detection.

B. Hough line transform

The Hough transform, [20], is used to detect straight line features such as the tower and blades of the wind turbine. The progressive probabilistic Hough line transform [21] in OpenCV runs efficiently. These properties make the Hough transform attractive, and it was therefore chosen as the basis for the recognition program.

Considering the detected lines, there are some issues. Sometimes a line which should be detected as a single line is broken into smaller segments. Another issue may appear if there is a significant amount of noise in the image. Small changes can affect the resulting edge map, resulting in frame to frame variability of the Hough transform. This has a notable effect on the start and end points of the detected lines. Those matters aside, the Hough transform was found to give accurate description of the contours of the wind turbine. The direction and location of the lines makes it possible to identify the blades and tower along with their orientation.

C. Pre-processing and Canny edge-detector

The Hough transform operates on a binary representation of the image, and it is necessary with pre-processing that should also make the algorithm robust to background, lighting conditions, UAV motions not compensated for by image stabilization (in camera, gimbal or software), and other sources for inaccuracies.

We use the standard Canny edge detection, where the first part consists of computing directional derivatives using the Sobel operator, [20], based on gray-scale images. It utilizes two 3×3 kernels for vertical and horizontal differentiation. Then the edges are thinned by keeping only the strongest gradients along the edge when tracking the edges. The final step converts the edge map to a binary image and removes weak edges that are not likely to be of significance. This is done through hysteresis thresholding where the thresholds are tuned experimentally.

D. Wind turbine recognition

1) Locating the tower: Among the parts of the wind turbine, the tower is arguably the simplest to correctly identify. Similarly to the blades it tends to exhibit long and distinct edges, but whereas the blades may settle in any orientation, the tower is fixed in its vertical stance. Therefore one could begin by searching for vertical lines, with a few degrees tolerance. An issue arises due to the prevalent roll of the UAV. One solution is to use the roll angle estimated from the IMU to adjust the search angle. Another option is to mount the camera on a gimbal stabilizer to physically prevent rotation of the image frame.

Because the UAV expectedly flies at an altitude at level with the hub, the base of the tower will be below the bounds of the image and consequently the lines of the tower continues beyond the lower edge of the image frame. Based on this knowledge another restriction is imposed; the lines representing the tower must possess an ending point near the lower bound of the image.

2) Locating the blades: If at least one line was identified as part of the tower, the algorithm proceeds to search for the lines of the blades. By identifying the highest point among all the tower lines, a first estimate is provided for the hub location. Thus by searching for lines ending in a

circular area around this point one can expect to find lines belonging to the blades.

An issue with this approach occurs when the horizon appears at a vertical position in the image close to the hub, which can potentially lead to it being incorrectly identified as a blade. One cannot simply ignore horizontal lines, because (as is apparent in Figure 4) a blade may also be horizontally aligned.

The lines which were determined to belong to blades are sorted by angle and grouped with other lines of adjacent angles. Together each group form a single blade object, where their average angle determines the estimated angle of the blade. Similarly, the tower lines form a tower object.

The property of there being a 120 degree angle between the blades is utilized by implementing a voting system. A blade which receives a certain amount of votes is assumed to be a false detection and is removed. The voting is conducted by comparing each detected blade to each other. If the angle between two blades is not close enough to 120 degrees by a certain tolerance, both blades obtain a vote. The tolerance was set to 20 degrees to accommodate for measurement error and skewed angles due to potential yaw angle difference between the UAV and wind turbine. Setting the vote limit to 3, such that three votes or more signifies a false blade, proved to work well. Thus, if for instance, the three actual blades plus a fourth false blade is detected, then the false blade receives three votes in total from the actual blades and is removed. Meanwhile, the true blades receive only one vote from the false blade. Figure 4 shows how the false blade lines are removed when voting is successful.

3) Locating the hub center: The first step is to calculate intersections between the recognized blade lines. Since all the blades beam out from the hub, the hub location can be found by calculating their common intersection point. However, it can be observed in Figure 4 that the detected lines for the blades are sources of disagreement. For instance, by looking at the upper right blade one observes that the curvature on the lower edge causes a steeper angle on the corresponding line compared to the lines from the upper edge. Since lines are detected at both edges of the blades there will be a spatial disagreement as well. At this point it is unclear which lines provide the appropriate result. This was handled by calculating the intersection points between every line pair, except between lines belonging to the same blade, see Figure 5.

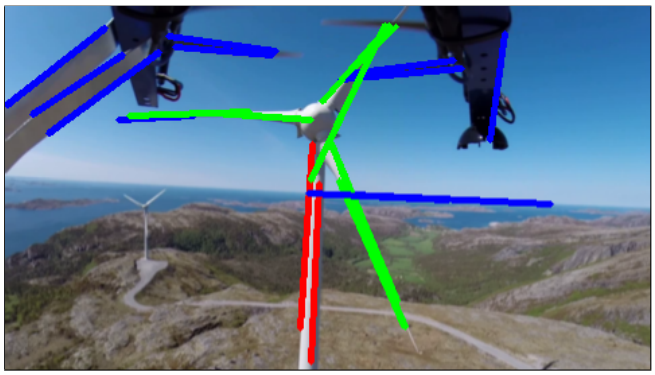


Fig. 4: The false detections have been removed by the voting procedure. Only the green lines corresponding to the actual blades detected.

Fig. 5: The green circles show the points where the blade lines intersect each other.

The second step is to obtain the average intersection point. From the previous step there may appear some stray intersection points causing discrepant results. For instance, if the horizon is incorrectly detected as a blade it can intersect with the other blades at a distance far from the hub. To prevent such scenarios, restrictions are imposed on the points of intersection. Points which have horizontal position far from the line extended by the tower are ignored. Since the actual vertical position of the hub is more uncertain, a less strict restriction is imposed on vertical distance based on the temporary estimated hub center (highest point among the tower lines). After the points caught by the restrictions have been removed, an average is calculated from the remaining points which results in the estimated hub center position.

E. Feature Tracking

Feature tracking refers to tracking how a feature moves or evolves over time in a sequence of images. This requires state information about its position being updated between frames. Our algorithm tracks the position of the hub center in the image frame, so that the UAV can be maneuvered in the fashion described in section II.

Noise and failure to detect the hub is dealt with using the Kalman filter with a simple motion model where the velocity is modeled as a Markov process, [20]. For the case when the hub is not detected one may estimate its position using a prediction from the model of the filter. This is done by only running the prediction step in the Kalman filter, while skipping the measurement update steps. Consequently, the program will still have knowledge of the states of the object, and can act accordingly. However, if the program is not able to locate the hub after an extended period of time, e.g. when the covariance exceeds a given threshold, instead of continuing to blindly rely on the model it should switch to the searching state where movement is stopped until the wind turbine has been identified again by rotating about the yaw axis to search.

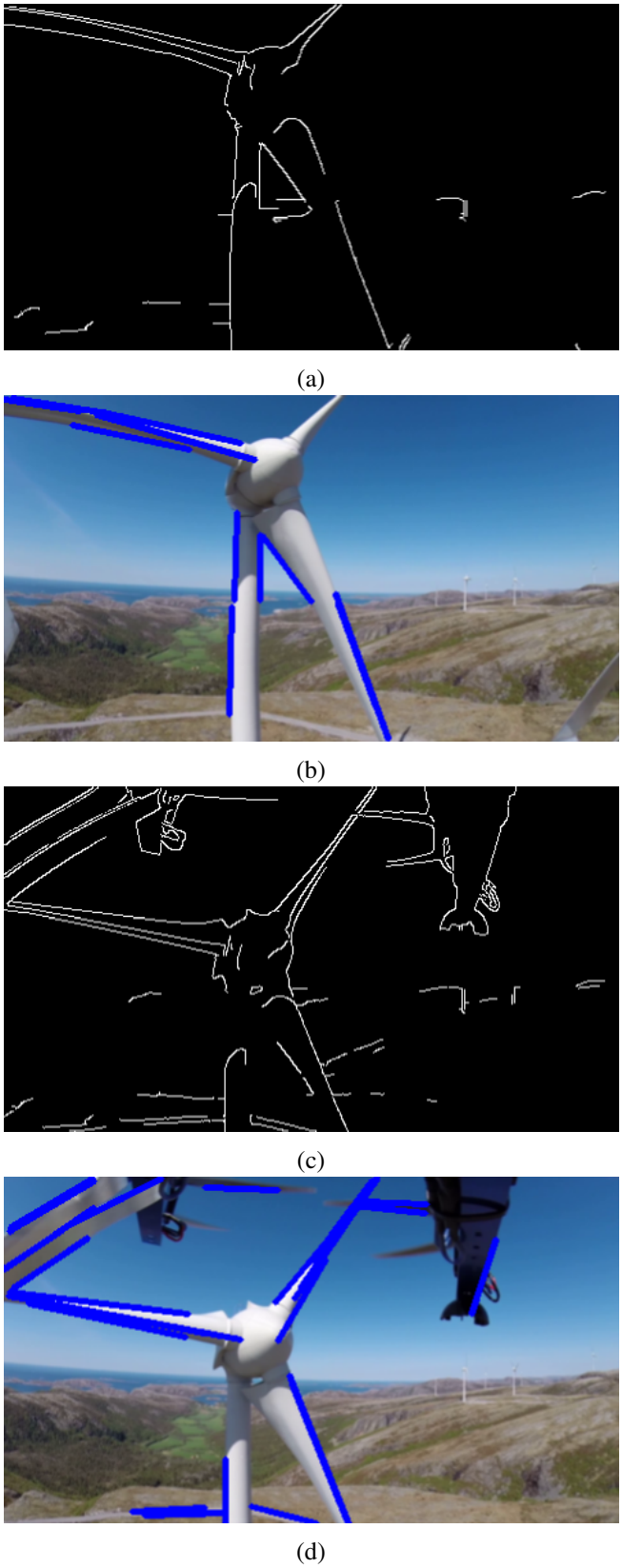
IV. Experimental results

In this section the program will be evaluated by using video sequences as input. The videos were recorded at Bessakerfjellet wind farm using a GoPro camera mounted in a stabilizing gimbal on a hexa-copter. The experiments mimic the maneuvering plan described in Section II-A, to get representative footage at the expected range and angles. The images are down-sampled to 480x270 pixels.

The UAV is controlled using an ArduPilot autopilot. It can receive commands from the pilot via the operator display or remote control unit. It can also receive commends from an on-board PandaBoard SBC where the machine vision, estimation, navigation and high-level control is implemented in real-time.

A. Recognition performance

The Canny edge detector was found to be a critical part in order to perform robustly under lighting variation, and good tuning of its parameters is



(a) (b) (c) (d)

Fig. 6: (a) Shows how the shadow under the hub causes a split left edge of the tower which in (b) results in two separate Hough lines. (c) Shows how the same shadow blends into the background and hides the edge, which leads to no Hough line detected in (d).

important. A change in lighting can affects the intensity distribution in the image, and can thus alter the strength of the edges. The hysteresis threshold values need to be adjusted low enough to accept the most prominent edges, but high enough to filter weaker unwanted edges. It should be emphasized that the transitions from good to bad is gradual, and we have only studied a limited number of videos from a single site under a limited number of weather and lighting conditions. The Hough line transform was found to be inherently robust. Because the shape of the wind turbine always remains the same, a specific set of parameters will produce similar results unless the Canny detector outputs a significantly different edge map, in which case the Canny edge detector needs to be adjusted.

Assuming the minimum requirements for successful detection are met, it will now be discussed how a detection may fail. This can occur when either the tower is not detected or if the sufficient amount of valid blades is not detected. Tower detection fails if a too small part of the tower is visible. If the tower edges are split into smaller lines, it can lead to failed detections of hub positions. Such an issue arose with the appearance of a shadow cast by the hub onto the tower. For example, Figure 6 shows how this shadow disrupts the edge detector and consequently also the Hough transform. Worse than segmenting the lines, it can hide the edges completely in the shadow area if the background is of matching color, such that the highest point among the tower lines (which form the center for the blade search area) is lower than it should be. The effect is stronger when the distance to the wind turbine is shorter, because then the shadow appears larger in the image.

Another scenario is when too many blades are (incorrectly) detected. Then there is much uncertainty as to which detections are true blades, and the voting procedure will discard all detections.

B. *False positives*

A successful detection does not necessarily imply a correct detection, due to the occurrence of false positives, being detected tower lines or blade lines which do not actually correspond to edges of the tower or blades.

On the other hand, false blade detections occurred more frequently. Every detected line which has an end point inside the blade search radius yields a true or false positive. This was most commonly caused by the horizon and the visible parts of the hexa-copter itself, see Figure 7, and to some degree the environment when the blade search radius was big. However, the environment produced very few edges in general. Many of the false edges are removed by the voting procedure (section III-D.2) or are dominated by the usually more numerous true blade edges.

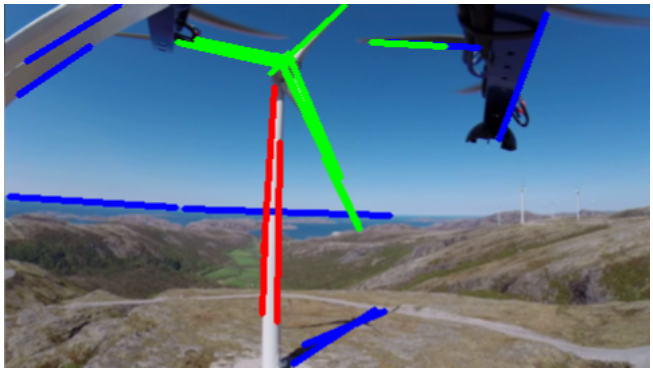


Fig. 7: The interference of the hexacopter is an example of where the edge of another entity is falsely detected as a blade edge, illustrated by the incorrect green line.

C. *Distance estimation*

Distance is estimated using eq. (2) with the width of the tower as known information. This was done by first finding the outermost tower lines, and extending them to a common height and measuring the length between them (in pixels). It was considered to use the highest point among all tower lines as the common height, but since this point can vary significantly between frames the bottom edge of the image was chosen instead. This causes a more stable estimation, but corresponds to a more uncertain actual width of the tower.

The results are shown in figure 8. It is observed that the resolution of the distance estimations is low, which is due to the low display resolution in the image. For instance, a single pixel in difference causes a jump in distance estimation from 35.2 m to 39.6 m. The results show estimations with somewhat larger distance than this in general. For clip # 1 and towards the end of clip # 2, the actual distance is roughly unchanged, but the results showed the estimated distance increasing. This is

because the hexa-copter was flying upwards, and thus a higher part of the tower (which is thinner) is measured. The spikes almost reaching zero from video clip # 1 are caused by detection of the tower of another wind turbine. In video clip # 2 there is most variation at the beginning towards the 150th frame because the hexa-copter moves forward and upwards.

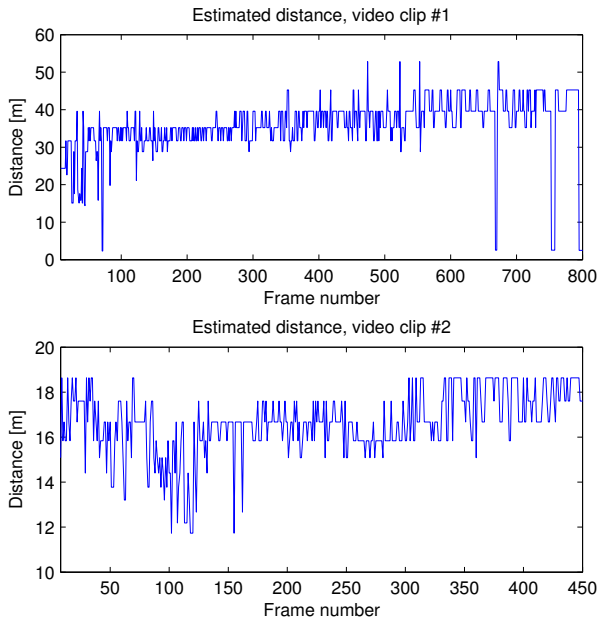


Fig. 8: Estimated distances for video clip # 1 and 2. Actual distances were roughly estimated to lie around 30 and 15 meters for clips # 1 and 2, respectively.

D. Yaw angle estimation

Yaw estimation was done using eq. (4). Distance estimates from the previous section was used for d. The results are shown in Figure 9. The average value does not lie too far from the expected values, but there is significant amounts of noise. An essential cause for the high sensitivity is that the actual distance between the hub center and the top of the tower is short compared to the distance from the hexa-copter and the measured length, δ. Since a low threshold value for the Canny edge detector was used, the hub center estimation was less accurate, affecting the measurement of δ which relies on this value.

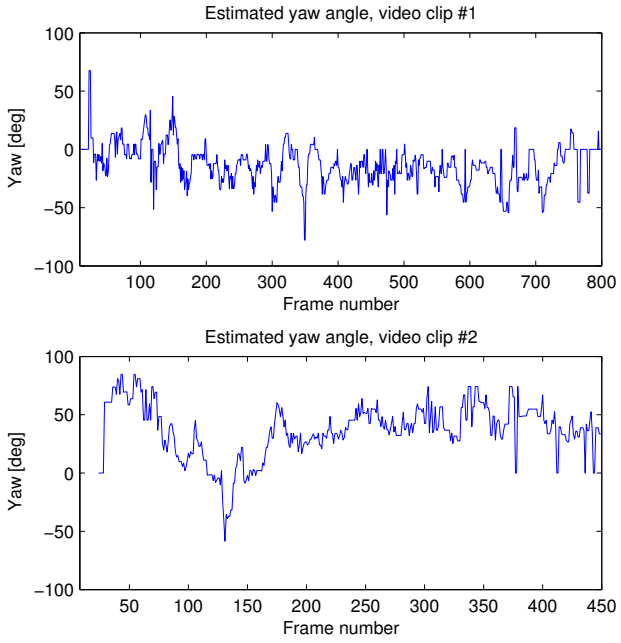


Fig. 9: Yaw angle estimation for video clip # 1 and 2. The actual angles were roughly estimated to lie around -10 and 30 degrees for clips # 1 and 2, respectively.

E. Real-time performance

The computation time was examined on the PandaBoard SBC with an e-CAM-51 USB camera. The results show a computation time per frame mostly in the range of 90-140 ms which translates to around 7-11 Hz. The Hough transform accounts for the majority of the computations with 60-80 ms per frame.

V. Conclusions

A machine vision algorithm for recognition and tracking of a wind turbine has been presented. Based on knowledge from the tracking data a method was suggested for approaching the wind turbine for inspection. The wind turbine recognition algorithm was based primarily on the Hough line transform, due to its ability to capture the main features of the wind turbine and its low computational demand. The hub center position was estimated by analyzing Hough lines and tracked using a Kalman filter. Experimental data acquired by a UAV at a wind park is used to evaluate the accuracy and robustness of recognition and navigation. It is found that under the tested conditions,

the method gives sufficient accuracy for the task and can execute in real time on a single board computer in the UAV.

ACKNOWLEDGMENTS

This work was supported by the Research Council of Norway through the Centers of Excellence funding scheme, Project 223254 - Centre for Autonomous Marine Operations and Systems (AMOS). The authors thank UAV operations manager Lars Semb at NTNU, Øystein Skotheim at SINTEF ICT, and TrønderEnergi Kraft AS for access to Bessakerfjellet wind park.

REFERENCES

[1] M. Williams, D. Jones, and G. Earp, "Obstacle avoidance during aerial inspection of power lines," *Aircraft Engineering and Aerospace Technology*, vol. 73, no. 5, pp. 472–479, 2001.

[2] S. Hrabar, G. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3309–3316.

[3] M. Blosch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2010, pp. 21–28.

[4] S. Saripalli, J. Montgomery, and G. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Trans. Robotics and Automation*, vol. 19, no. 3, pp. 371–380, June 2003.

[5] J. Courbon, Y. Mezouar, N. Guénard, and P. Martinet, "Vision-based navigation of unmanned aerial vehicles," *Control Engineering Practice*, vol. 18, no. 7, pp. 789 – 799, 2010, special Issue on Aerial Robotics.

[6] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi, "A vision-based guidance system for UAV navigation and safe landing using natural landmarks," in *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, 2009*. Springer Netherlands, 2010, pp. 233–257.

[7] Y. Zhou, T. Wang, J. Liang, C. Wang, and Y. Zhang, "Structural target recognition algorithm for visual guidance of small unmanned helicopter," in *Proc. IEEE Int. Conf. Robotics and Biomimetics, Guangzhou, China*, 2012, pp. 908–913.

[8] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. Robotics*, vol. 24, no. 2, pp. 331–340, April 2008.

[9] L. Mejias, S. Saripalli, G. S. Sukhatme, and P. Cervera, "Visual servoing for tracking features in urban areas using an autonomous helicopter," *J. Field Robotics*, vol. 23, pp. 185 –199, 2006.

[10] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, L. Eck *et al.*, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Trans. Robotics*, vol. 25, no. 3, pp. 743–749, 2009.

[11] E. Frew, T. McGee, Z. W. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta, "Vision-based road-following using a small autonomous aircraft," in *Proc. IEEE Aerospace Conference*, 2004, pp. 3006–3015.

[12] S. Rathinam, Z. Kim, A. Soghikian, and R. Sengupta, "Vision based following of locally linear structures using an unmanned aerial vehicle," in *Proc. IEEE Conf Decision and Control, Seville, Spain*, 2005, pp. 6085–6090.

[13] E. W. Frew, "Comparison of lateral controllers for following linear structures using computer vision," in *Proc. American Control Conference, Minneapolis*, 2006, pp. 2154–2159.

[14] S. Rathinam, P. Almeida, Z. W. Kim, S. Jackson, A. Tinka, W. Grossman, and R. Sengupta, "Autonomous searching and tracking of a river using an UAV," pp. 359–364, 2007.

[15] T. S. Bruggermann, J. J. Ford, and R. A. Walker, "Control of aircraft for inspection of linear infrastructure," *IEEE Trans. Control Systems Technology*, vol. 19, pp. 1397–1409, 2011.

[16] S. Du and C. Tu, "Power line inspection using segment measurement based on ht butterfly," in *IEEE Int. Conf. Signal Processing, Communications and Computing (ICSPCC)*, Sept 2011, pp. 1–4.

[17] N. Metni and T. Hamel, "A UAV for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, vol. 17, no. 1, pp. 3 – 10, 2007.

[18] M. Stokkeland, "A computer vision approach for autonomous wind turbine inspection using a multicopter," 2014, master thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim.

[19] S. Høglund, "Autonomous inspection of wind turbines and buildings using an UAV," 2014, master thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim.

[20] E. R. Davies, *Computer and machine vision: Theory, algorithms, practicalities*, 4th ed. Elsevier, 2012.

[21] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic Hough transform," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119 – 137, 2000.