# Saliency-driven Visual Inspection of Vessels by means of a Multirotor

Alberto Ortiz, Francisco Bonnin-Pascual, Emilio Garcia-Fidalgo and Joan P. Company

*Abstract*— **Vessel maintenance entails periodic visual inspections of the hull in order to detect defects such as coating breakdown, corrosion or cracks. These inspections are typically performed by well-trained surveyors at a great cost because, for the survey to be valid, the inspector is required to be at arms' reach from the surface under inspection, what entails the provision of access means (e.g. scaffolding) to reach any point of interest in the hull. The work that is presented in this paper builds upon a previous development of a multirotor to be used as an assistant during this kind of inspections, as a remote camera for close-up inspection. On this occasion, we propose a semi-autonomous platform whose control architecture implements a supervised autonomy (SA) approach to enlarge the range of affordable inspections because of the inclusion of the operator in the main control loop. According to the SA paradigm, the control architecture provides a number of autonomous functions to make simpler the inspection operation. As part of these functions, and on the basis of the relationship between image saliency and the presence of defects, we propose an image saliency-driven capability which aims at collecting close-up images from relevant areas of the hull and at the same time fully cover the surfaces under inspection.**

## I. INTRODUCTION

The movement of goods by vessels has been one of the most cost and time effective transportation methods so far. An always increasing safety standard level has to be achieved in order to improve the transportation service quality. To this aim, constant surveying and subsequent repairs have to be carried out to ensure the structural reliability of these ships.

Nowadays, this structural integrity is assessed through periodical inspections, carried out by surveyors who operate using well established methodologies and techniques. The effort in the inspection tasks is huge and often has to be carried out by a single surveyor within a short amount of time in order to return the vessel to service. Besides, the surveying and subsequent repairs require many of the tanks (cargo, fuel, ballast, etc) to be cleaned and ventilated, and suitable access means to be arranged —scaffolding, cherry pickers, etc. In the case of older ships, the preparation of each compartment is necessary to allow the repair crew to enter and access the element that has failed; on these occasions, the surveyor can perform the inspection using the same facilities without additional cost. However, in a younger ship, the number of required repairs is small, so the bulk of these arrangements is only required for the survey itself. As a result, a greater proportion of the preparation cost is entirely due to the survey, which for many well run ships is likely to reveal no defects in the early years.

However, the maritime industry is forced to follow the general trend of rationalization so as to reach a higher level of standardization of the procedures related to marine transportation that will enable higher performance according to safety and financial criteria. One way to achieve this is through the incorporation of novel technological means that increase the level of automation. One of the main goals of the already concluded EU FP7 project MINOAS (Marine INspection rObotic Assistant System) was to develop a fleet of robotic platforms with different locomotion capabilities with the aim of teleporting the human surveyor to the different vessel structures to be inspected. Given the enormity of these structures and the requirement for vertical motion as part of the inspection process, a multirotor platform (also named as *micro-aerial vehicles* or MAVs by some authors) was selected as one of the members of the fleet due to their small size, agility and fast deployment time.

These platforms have become increasingly popular in recent years, and, as a consequence, a number of MAV navigation solutions, comprising platform stabilization, self-localization, mapping, and obstacle avoidance, can be found in the related literature. They differ mainly in the sensors used to solve these tasks, the amount of processing that is performed onboard/offboard, and the assumptions made about the environment. The laser scanner has been used extensively due to its accuracy and speed. For instance, [1], [2] proposed full navigation systems using laser scan matching and IMU fusion for motion estimation embedded within SLAM frameworks that enable such MAVs to operate indoors. In [1], [3], a multilevel approach is described for 3D-mapping tasks. Infrared or ultrasound sensors are other possibilities for implementing navigation solutions. Although they typically have less accuracy and require higher noise tolerance, several researchers (see [4]–[6]) have used them to perform navigation tasks in indoor environments, since they are a cheaper option than laser scanners. Finally, vision-based navigation has turned out to be of paramount relevance lately, mostly because of their particular suitability for MAVs as they are low-power, low-price and small, but rich sensors. This richness, however, entails higher computational costs, what has led researchers to find optimized solutions that can run over low-power processors. Among them, some researchers propose visual navigation solutions (embedded within SLAM frameworks or not) based on feature tracking, either adopting a frontal mono or stereo camera configuration

This work is partially supported by the European Social Fund through grant FPI11-43123621R (Conselleria d'Educacio, Cultura i Universitats, Govern de les Illes Balears) and by project INCASS. This project has received research funding from the EU FP7 under GA 605200. This publication reflects only the author's views and the European Union is not liable for any use that may be made of the information contained therein.

All authors are with the Department of Mathematics and Computer Science, University of the Balearic Islands, 07122 Palma de Mallorca, Spain, email: alberto.ortiz@uib.es

(e.g. [7]–[10]) or choosing a ground-looking orientation (e.g. [11], [12]). Others focus on efficient implementations of optical-flow calculations, either dense or sparse, and mostly from ground-looking cameras (e.g. [13]) or develop methods for landing, tracking, and taking off using passive (e.g. [14]), or active markers (e.g. [15]).

The MINOAS aerial platform consisted in an autonomous MAV fitted with a flexible set of cameras to provide a first overview of the vessel structures inside cargo holds [16]. The platform was devised for indoor/semi-indoor flights (i.e. GPS-denied scenarios), and featured way-point navigation and obstacle avoidance. These tasks were implemented by means of a 30 m-range laser scanner that was used to feed a laser scan-based odometer and a SLAM solution. The mission was specified as a list of way-points to reach and the actions to perform once reached, e.g. take a picture.

The INCASS project (Inspection Capabilities for Enhanced Ship Safety), a follow-up of MINOAS, is currently under development also in the framework of EC funding. One of the tasks which is still running has to do with the re-design of the aerial platform in accordance to some constructive advice from end-users during the final field trials of MINOAS [17]. Ship surveyors essentially referred to the usability and flexibility of the platform: instead of way-point navigation, what required the specification of a precise list of points for each mission, they suggested the implementation of a friendly, flexible and robust way to interact with the platform so that they could take the robot to any point of e.g. a cargo hold without the need to be an expert pilot.

In this regard, we have developed a novel semi-autonomous aerial platform intended for the aforementioned visual inspection task. This robot has been devised to include an operator in the position control loop to extend the range of affordable inspections. Meanwhile, the control architecture autonomously takes care of attitude stabilization, as well as speed and height control, including hovering flight (i.e. keep current height and speed 0) whenever there is no command pending to be executed. A higher control layer implements additional functionalities which provide further assistance to the operator in the form of added autonomous functions. This includes collision prevention or avoidance of getting too far from the wall under inspection, among others. This shared control is implemented via a *supervised autonomy* (SA) approach [18] and making use of behavior-based control [19].

In this paper, we describe an on-going work that is to complement the upper control layer of the MAV with behaviors aiming at enhancing the quality and relevance of the collected images. The goal is twofold: (1) prevent the platform from performing movements that can negatively affect image quality and (2) focus the capture of images onto relevant areas from the visual inspection point of view. Regarding the latter, we show the correlation between image saliency and the presence of defects, what can be used to drive image taking towards the "hot spots" of the hull for close-up inspection.

The rest of the paper is organized as follows: Section II describes the aerial platform, Section III outlines the saliency-based defect detection approach, Section IV describes the control architecture, and Section V concludes the paper.

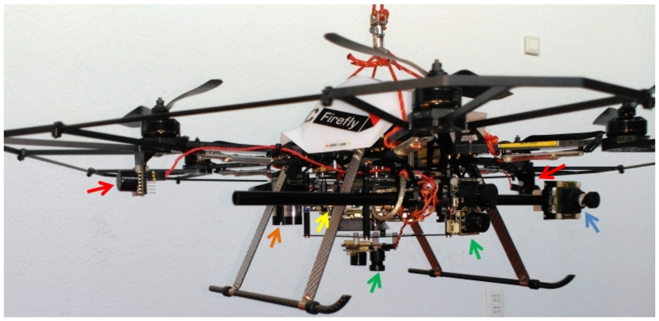

Fig. 1. Firefly platform and the visual inspection sensor suite: (green) front-looking / bottom-looking optical flow sensors, (red) left / right ultrasound sensors, (orange) height sensor, (yellow) embedded PC, (blue) camera.

## II. THE INCASS AERIAL PLATFORM

Figure 1 shows a Firefly hexacopter from Ascending Technologies configured for the visual inspection application, fitted with a lightweight sensor suite[1] comprising:

- Two optical flow sensors, one looking to the ground and the other pointing forward, to estimate the vehicle speed. PX4Flow sensors [20] are used to this end.
- Range sensors pointing in different directions for collision prevention. Currently, we use two Maxbotix HRLV-EZ4 sensors oriented to the left and to the right, providing range data up to 5 m at 10 Hz.
- An additional height sensor that provides an extended range above the 5 m covered by the PX4Flow oriented downwards. To this end, we use the LIDAR-Lite laser range finder (http://pulsedlight3d.com/), which provides range data up to 40 m at 50 Hz.
- A reconfigurable set of cameras that collect the requested images from the vessel structures under inspection. For instance, the Firefly shown in Fig. 1 features a minimalistic configuration comprising a single forward-looking uEye UI-1221LE camera.

The vehicle carries an additional processing board which avoids sending sensor data to a base station, but process them onboard and, thus, prevents communications latency inside critical control loops. In contrast to the two microcontrollers integrated in the Firefly's *flight control unit*, this processor will be referred to as the *high-level processor* from now on. The configuration shown in Fig. 1 incorporates a Mastermind board (Intel Core2Duo 1.86 GHz processor, 4 GB RAM).

## III. SALIENCY DETECTION FRAMEWORK

### A. Bayesian Approach for Saliency Computation

We consider defects as rare phenomena that may appear on a regular surface or structure. Since they are rare, the probability that an area is affected by a defect is rather low.

[1] $2 \times 17$ g (PX4Flow) + $2 \times 5$ g (HRLV-EZ4) + 16 g (LIDAR-Lite) + 12 g (uEye camera) = 72 grams

In this work, we use this low probability for detecting salient areas in digital images.

Similarly to [21], we make use of a Bayesian approach for computing a saliency map $\Sigma_{ij}$:

$$\Sigma_{ij} = \underbrace{\frac{1}{p(F=f_{ij})}}_{\substack{\text{Independent}\\ \text{of target}\\ \text{(bottom-up saliency)}}} \underbrace{\underbrace{p(F=f_{ij}|C=\tau)}_{\text{Likelihood}}\underbrace{p(C=\tau|L_{ij})}_{\text{Location prior}}}_{\substack{\text{Dependent on target}\\ \text{(top-down knowledge)}}} \tag{1}$$

where $f_{ij}$ is the feature value (F) found at the image point $L_{ij}$ and $\tau$ is the target class (C), i.e. the defect class in our case. Hence, equation 1 combines top-down information with bottom-up saliency to find the pointwise mutual information between the feature and the target. Furthermore, since defects do not depend on their location in the image, we drop the rightmost term of equation 1:

$$\Sigma_{ij} = \frac{1}{p(F=f_{ij})} p(F=f_{ij}|C=\tau) \tag{2}$$

Using this formulation, the saliency at a given point $(i, j)$ decreases as the probability of feature value $f_{ij}$ gets higher, and increases as the probability of feature value $f_{ij}$ for the defect class increases.

### B. Contrast-based Saliency

As said before, we consider defects as rare phenomena that catch the visual attention of the observer during visual inspection. Following this idea, we propose to describe defects by means of features typically used in cognitive models to predict human eye fixations. In this regard, one of the most influential saliency computational models is based on contrast [22]. The contrast levels in intensity, color and orientation are computed as center-surround differences between fine and coarse scales over image pyramids of several levels; that is to say, the difference between each pixel on a fine (or center) scale $c$ and its corresponding pixel in a coarse (or surrounding) scale $s$ is calculated as $M(c, s) = |M(c) \otimes M(s)|$, where $\otimes$ is the center-surround operator. Given an RGB colour image, this process is performed over: ($\oplus$ denotes the across-scale addition operator)

- the intensity channel $I = (r + g + b)/3$ to build the *intensity conspicuity map* $IM = \oplus_{c=c_a}^{c_b} \oplus_{s=s_a}^{s_b} N(I(c, s))$, where $r$, $g$ and $b$ are the original red, green and blue channels;
- the color channels RG and BY defined as $RG = R - G$ and $BY = B - Y$, with $R = \max\{0, r-(g+b)/2\}$, $G = \max\{0, g-(r+b)/2\}$, $B = \max\{0, b-(r+g)/2\}$ and $Y = \max\{0, (r+g)/2 - |r-g|/2 - b\}$, to build the *color conspicuity map* $CM = \oplus_{c=c_a}^{c_b} \oplus_{s=s_a}^{s_b} N(RG(c, s)) + N(RG(c, s))$; and
- the orientation channels $O(\theta)$, calculated by convolution between channel $I$ and Gabor filters oriented at 0°, 45°, 90° and 135°, to build the *orientation conspicuity map* $OM = \sum_\theta N\left(\oplus_{c=c_a}^{c_b} \oplus_{s=s_a}^{s_b} N(O(c, s; \theta))\right)$.

The map normalization operator $N(\cdot)$ is intended to highlight saliency peaks in maps where a small number of strong peaks of activity (conspicuous locations) is present, while globally suppressing peaks when numerous comparable peak responses are present. To this end: (1) the map is normalized to a fixed range, (2) the global maximum $M$ is found, (3) the local maxima average $\overline{m}$ is determined, and (4) the map is multiplied by $(M - \overline{m})^2$.

Finally, the three conspicuity maps are normalized and summed into the final output:

$$F = \frac{1}{3}\left(N(IM) + N(CM) + N(OM)\right) \tag{3}$$

### C. Defect Detection Performance

Figure 2(a) shows *probability density functions* (PDFs) for contrast, i.e. $p(F = \text{contrast}_{ij})$, and contrast conditioned on the presence of defects, i.e. $p(F = \text{contrast}_{ij}|C = \text{defect})$, both determined by means of the Parzen windows method [23] from a training image set comprising surfaces and structures containing cracks, coating breakdown and corrosion.

Using the statistics learnt, we have obtained the detection results which can be viewed in Fig. 3. Moreover, we have evaluated the detection approach using *leave-one-out cross-validation* [24]: one image is selected from the dataset, while the rest is used to obtain the feature PDFs that make up the defect detector (training step), the selected image is next used to validate the detector, and the process is repeated for each image in the dataset. Final performance is shown in Fig. 2(b) in the form of a ROC curve relating *true positive rate* (TPR) and *false positive rate* (FPR). (FPR,TPR) points result from thresholding the defect maps at different contrast levels and comparing the resulting binary image with a ground truth. The *area under the curve* [25] turned out to be 0.88, quite above the performance of a random classifier.

To finish, Fig. 2(c) shows a histogram of execution times for 320 × 240-pixel images. As can be observed, contrast can be computed fast (on average lesss than 80 ms), and with little variation from image to image ($\sigma \approx 7$ ms).

## IV. CONTROL ARCHITECTURE

The aerial platform implements a control architecture that follows the *supervised autonomy* (SA) paradigm [18]. This is a human-robot framework which allows the user to be within the general platform control loop although the robot implements a number of autonomous functions, including self-preservation and other safety-related issues, which make simpler the intended operations for the user, who, in this way, can focus in accomplishing the task at hand.

The control software has been configured to be hosted by any of the research platforms developed by Ascending Technologies (the quadcopters Hummingbird and Pelican, and the Firefly hexacopter, shown in Fig. 1), although it could be adapted to other systems. In more detail, the control software has been organized around a layered structure distributed among the available computational resources. On the one hand, the *low-level control* layer implementing attitude stabilization and direct motor control executes over the main microcontroller as the platform firmware provided by the

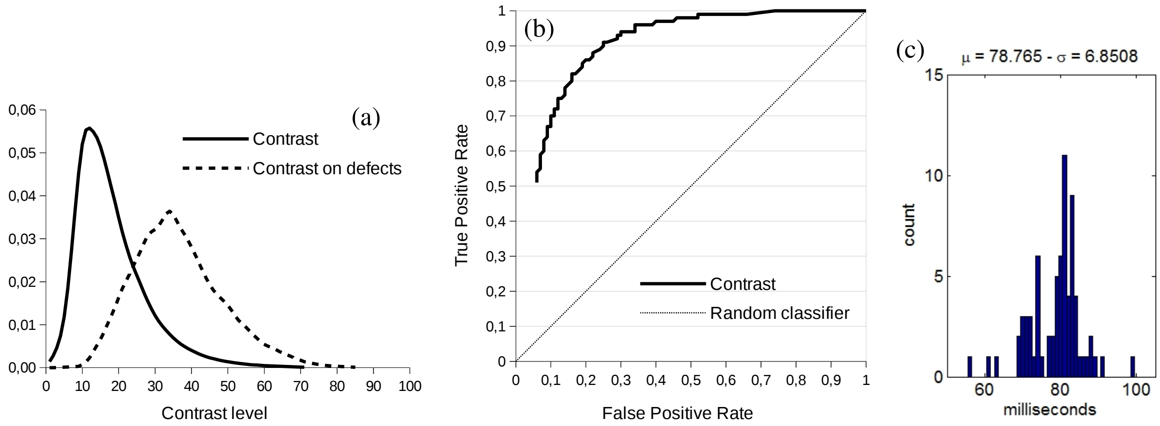

Fig. 2. (a) PDFs for contrast. (b) ROC curve for the defect detector [AUC = 0.88]. (c) Execution times.

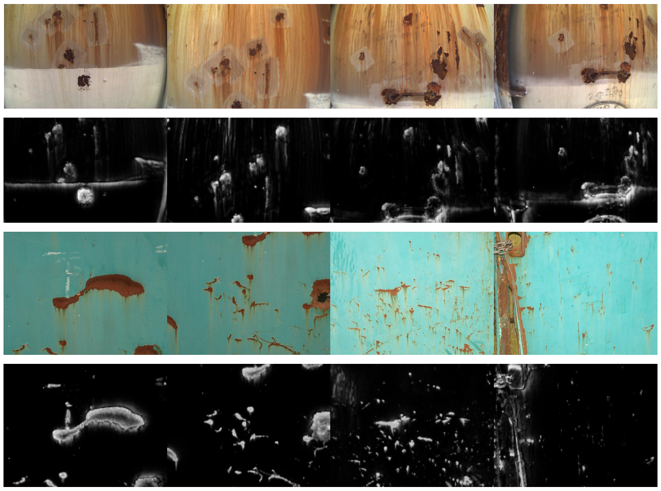

Fig. 3. Original images and corresponding defect maps.

manufacturer [26]. On the other hand, *mid-level control*, running over the secondary microcontroller, comprises height and velocity controllers which map input speed commands into roll, pitch, yaw and thrust orders. Lastly, the *high-level control* layer implements a reactive control strategy coded as a series of ROS nodes running over Linux Ubuntu, which combine the user desired speed command with the available sensor data —$x$, $y$, $z$ velocities, height $z$ and distances to the closest obstacles—, to obtain a final and safe speed set-point that is sent to the speed controllers. See [27] for the details.

Speed commands are generated through a set of robot behaviors organized in a hybrid competitive-cooperative framework [19]. That is to say, on the one hand, higher priority behaviors can overwrite the output of lower priority behaviors by means of a suppression mechanism taken from the *subsumption* architectural model. On the other hand, the cooperation between behaviors with the same priority level is performed through a *motor schema*, where all the involved behaviors supply each a motion vector and the final output is their weighted summation. An additional flow control mechanism selects, according to a specific input, between the output provided by two or more behaviours.

Figure 4(a) details the behavior-based architecture, grouping the different behaviors depending on its purpose. A total of four general categories have been identified for the particular case of visual inspection: (a) *behaviors to accomplish the user intention*, which propagate the user desired speed command, attenuating it towards zero in the presence of close obstacles, or keeps hovering until the WiFi link is restored after an interruption; (b) *behaviors to ensure the platform safety within the environment*, which prevent the robot from colliding or getting off the safe area of operation, i.e. flying too high or too far from the reference surface that is involved in optical flow measurements; (c) *behaviors to increase the autonomy level*, which provide higher levels of autonomy to both simplify the vehicle operation and to introduce further assistance during inspections; and (d) *behaviors to check flight viability*, which checks whether the flight can start or progress at a certain moment in time. Some of the behaviors in groups (a) and (c) can operate in the so-called *inspection mode*. While in this mode, the vehicle moves at a constant and reduced speed (if it is not hovering) and user commands for longitudinal displacements or turning around the vertical axis are ignored. In this way, during an inspection, the platform keeps at constant distance/orientation with regard to the front wall, for improved image capture.

To finish, the *sweep* behavior of group (c) makes the MAV perform a zig-zag movement to sweep the surface in front of the vehicle in accordance to the *finite state machine* outlined in Fig. 4(b). The process is as follows: once the operator has placed the MAV in front of the surface to survey and a *sweeping* has been requested, the MAV starts moving in horizontal direction towards the right or the left depending on which side is farther; once the other side has been attained, the MAV increases its height in a certain amount $\Delta z$ to reach the next row of the sweeping; next, it moves to the opposite side in the reverse horizontal direction. This process repeats until the maximum height has been attained, moment at which the vehicle descends up to a predefined height, e.g. 1 m. During horizontal motion, the MAV captures images at a frequency compatible with its speed (e.g. 5 Hz) and computes the previously described saliency maps looking for

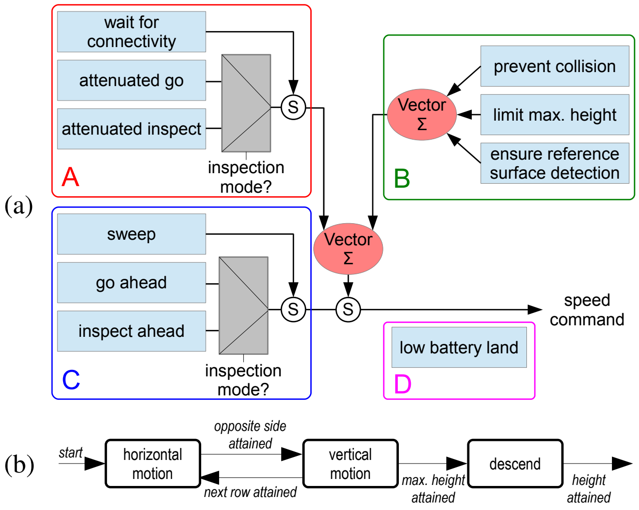

Fig. 4. (a) Behavior-based upper control layer. (b) *Sweep* behavior.

evidences of defects. Once a highly salient point is detected, the vehicle hovers for a while in front of the defect and takes a better, higher resolution image at close range.

## V. CONCLUSIONS

This paper has described an on-going work involving an aerial platform to be used for visual inspection of vessels, together with a defect detector based on image saliency. For the specific problem of visual inspection, we have described a behaviour-based control architecture embedded within a supervised autonomy framework which includes among its functionalities the enhancement of image capture thanks to the use of the saliency-based defect detector.

## REFERENCES

[1] I. Dryanovski, R. G. Valenti, and J. Xiao, “An Open-source Navigation System for Micro Aerial Vehicles,” *Auton. Robot.*, vol. 34, no. 3, pp. 177–188, 2013.

[2] S. Grzonka, G. Grisetti, and W. Burgard, “A Fully Autonomous Indoor Quadrotor,” *IEEE Trans. Rob.*, vol. 28, no. 1, pp. 90–100, 2012.

[3] A. Bachrach, S. Prentice, R. He, and N. Roy, “RANGE-Robust Autonomous Navigation in GPS-denied Environments,” *J. Field Robot.*, vol. 28, no. 5, pp. 644–666, 2011.

[4] S. Bouabdallah, P. Murrieri, and R. Siegwart, “Towards Autonomous Indoor Micro VTOL,” *Auton. Robot.*, vol. 18, pp. 171–183, 2005.

[5] A. Matsue, W. Hirosue, H. Tokutake, S. Sunada, and A. Ohkura, “Navigation of Small and Lightweight Helicopter,” *Trans. Japan Soc. for Aeronaut. and Space Sciences*, vol. 48, no. 161, pp. 177–179, 2005.

[6] J. F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, “Quadrotor Using Minimal Sensing for Autonomous Indoor Flight,” in *Proc. ECCV*, 2007.

[7] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, Lorenz, Meier, P. Tanskanen, and M. Pollefeys, “Vision-Based Autonomous Mapping and Exploration Using a Quadrotor MAV,” in *Proc. IEEE/RSJ IROS*, 2012, pp. 4557–4564.

[8] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, “Vision-based state estimation for autonomous rotorcraft mavs in complex environments,” in *Proc. IEEE ICRA*, 2013, pp. 1758–1764.

[9] J. Engel, J. Sturm, and D. Cremers, “Scale-aware navigation of a low-cost quadrocopter with a monocular camera,” *Rob. Auton. Syst.*, vol. 62, no. 11, pp. 1646–1656, 2014.

[10] C. Troiani, A. Martinelli, C. Laugier, and D. Scaramuzza, “Low computational-complexity algorithms for vision-aided inertial navigation of micro aerial vehicles,” *Rob. Auton. Syst.*, vol. 69, pp. 80–97, 2015.

[11] M. W. Achtelik, S. Lynen, S. Weiss, L. Kneip, M. Chli, and R. Siegwart, “Visual-inertial slam for a small helicopter in large outdoor environments,” in *Proc. IEEE/RSJ IROS*, 2012, pp. 2651–2652.

[12] G. Chowdhary, E. Johnson, D. Magree, A. Wu, and A. Shein, “GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft,” *J. Field Robot.*, vol. 30, no. 3, pp. 415–438, 2013.

[13] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, “MAV Navigation through Indoor Corridors Using Optical Flow,” in *Proc. IEEE ICRA*, 2010.

[14] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, “PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight Using Onboard Computer Vision,” *Auton. Robot.*, vol. 33, no. 1–2, pp. 21–39, 2012.

[15] K. E. Wenzel, A. Masselli, and A. Zell, “Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle,” *J. Intell. Rob. Syst.*, vol. 61, no. 1–4, pp. 221–238, 2011.

[16] F. Bonnin-Pascual, E. Garcia-Fidalgo, and A. Ortiz, “Semi-autonomous Visual Inspection of Vessels Assisted by an Unmanned Micro Aerial Vehicle,” in *Proc. IEEE/RSJ IROS*, 2012, pp. 3955–3961.

[17] M. Eich, F. Bonnin-Pascual, E. Garcia-Fidalgo, A. Ortiz, G. Bruzzone, Y. Koveos, and F. Kirchner, “A Robot Application to Marine Vessel Inspection,” *J. Field Robot.*, vol. 31, no. 2, pp. 319–341, 2014.

[18] G. Cheng and A. Zelinsky, “Supervised Autonomy: A Framework for Human-Robot Systems Development,” *Auton. Robot.*, vol. 10, pp. 251–266, 2001.

[19] R. C. Arkin, *Behavior-based Robotics*. MIT press, 1998.

[20] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, “An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications,” in *Proc. IEEE ICRA*, 2013, pp. 1736–1741.

[21] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, “SUN: A Bayesian Framework for Saliency Using Natural Statistics,” *J. Vision*, vol. 8, no. 7, pp. 1–20, 2008.

[22] L. Itti, C. Koch, and E. Niebur, “A Model of Saliency-based Visual Attention for Rapid Scene Analysis,” *IEEE T. Pattern Anal.*, vol. 20, no. 11, pp. 1254 –1259, nov 1998.

[23] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Academic Press, 2009.

[24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley Interscience, 2000.

[25] T. Fawcett, “An Introduction to ROC Analysis,” *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[26] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, “Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz,” in *Proc. IEEE ICRA*, 2007, pp. 361–366.

[27] F. Bonnin-Pascual, A. Ortiz, E. Garcia-Fidalgo, and J. P. Company, “A Micro-Aerial Vehicle based on Supervised Autonomy for Vessel Visual Inspection,” Department of Mathematics and Computer Science, University of the Balearic Islands, Tech. Rep. A-02-2015, 2015. [Online]. Available: http://dmi.uib.es/~xbonnin/TR-A-02-2015.pdf