Robust marker-tracking system for vision-based autonomous landing of VTOL UAVs

M. Verbandt,* B. Theys,† and J. De Schutter‡
Department of Mechanical Engineering PMA, KU Leuven,
Leuven, 3000 Belgium

Abstract

This article presents the development of a vision-based VTOL UAV docking system for outdoor applications. A new marker type is constructed, along with a robust detection algorithm based on the Hough transform, which forms the basis of the positioning system. For verification purposes, a binary classifier based on support vector machines is trained. Test data show great robustness against varying lighting, partial occlusion and unsharpness. The practical usefulness is demonstrated by creating a single tracking module, based upon the PX4FLOW camera. Equipping a quadrotor with the new positioning sensor allows it to perform a precision landing.

1 Introduction

Unmanned Aerial Vehicles (UAVs) are being used by the army for a long time now, but technological advancement is making them more and more available to the larger public. This development opens a door to a variety of new applications, e.g. rescue operations, urgent transportation of goods to remote areas, map-building, etc. One of the major drawbacks is their limited operating time. However, landing on a ground station offers the possibility to automatically replace the empty battery by a charged one, dramatically increasing the operating range and time of the UAV.

The main difficulty in automatic docking is the positioning. Figure 1 sketches the problem. The UAV has to determine its relative position and orientation w.r.t. the landing platform and use this information to carry out a precision docking. [1] is a good example of successful UAV docking in which the main focus was to execute persistent missions involving a hot-battery swap.

Docking mobile robots has been a research-topic for decades, during which many different positioning systems have been developed. Three main categories exist: passive optical systems, active optical systems and ultrasound

---
*Email address: maarten.verbandt@student.kuleuven.be
†Email address: bart.theys@kuleuven.be
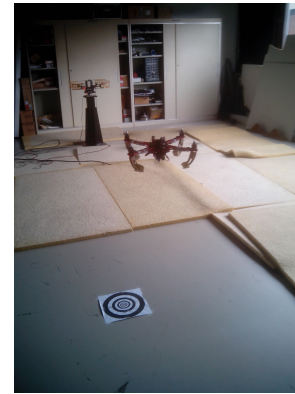‡Email address: joris.deschutter@kuleuven.be

Figure 1: Sketch of the docking problem: the UAV has to determine its relative position w.r.t. the landing platform and use this information to land on the target.

systems.

A passive optical positioning system consists of only light-sensitive devices, e.g. digital cameras. Detection of the specific docking target in the image allows estimation of the relative pose. A distinct pattern, called a marker, is usually added to ease the extraction from an image. Figure 2(a) gives an example of a passive optical positioning system. The marker is in this case a checkerboard pattern.
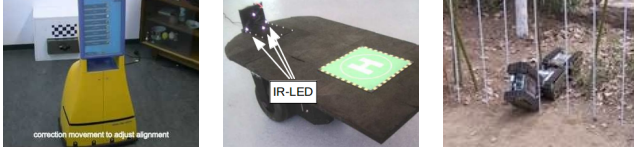
An active optical positioning system however contains additional controllable light sources on the target. Comparing images of the same scene with the sources switched on and off simplifies the extraction of the target. [2] describes a successful application on a UAV (figure 2(b)). Four IR-LEDs are mounted onto a landing platform which are then used as a reference for the UAV's position. A major drawback is that high intensity ambient lighting can overwhelm the source, making it impossible to locate the target.

An ultrasound positioning system, on the other hand, uses ultrasonic waves. Depending on the specific arrangement, multiple senders and receivers are attached to the robot and the target. This generally results in a set of distances, e.g. by measuring the time-of-flight, from which the relative pose can be calculated. Its main advantage is the decrease in computing power compared to image-based techniques. [3] and [4] describe applications in a 2D setting. Figure 2(c) shows an example.

(a) Passive optical system: detection of a checkerboard pattern[5].

(b) Active optical system: detection of IR-leds[2].

(c) Ultrasound system: intensity measurement[3].

Figure 2: Examples of different measurement systems to dock mobile robots.

This paper describes the development of a robust vision-based measurement system to make a quadrotor land autonomously on a target location, under varying ambient lighting conditions.

2 MARKER DETECTION

A passive optical system is perferable in this case, since it allows for a stand-alone sensor which can be easily attached to any UAV. The proposed marker is based on circles, because they carry positional information in a very natural way, namely by their center-coordinates. In order to improve robustness and to be able to detect the marker from closeby, a series of concentric circles with exponentially distributed radii is considered. This assures a similar pattern for varying viewing distance. A multitude of circles also increases the amount information indicating the marker's center. The proposed marker is shown in figure 3. These circles can be detected by the Hough transform[6] (HT), which is known as one of the most robust detection algorithms in computer vision and will be the core of the detection algorithm described in this paper.

Figure 3: Marker for the developed positioning system.

2.1 Edge detection

The computation of the HT relies on the edges, detected in an image, in which the specified shape is sought. This edge information is extracted by first converting the grayscale image to a binary form. The threshold separating dark and bright pixels is calculated locally with Otsu's algorithm[7], in order to cope with non-uniform lighting conditions. Since the marker contains only very strong edges, crucial information is preserved while weaker edges disappear. The latter increases the signal-to-noise ratio. Additionally, thresholding lowers the required memory to store the image and speeds up the edge detection as it is implemented by logical operations

(a) Image from a scene with the marker.

(b) Black-and-white image after local thresholding of 4(a).

(c) Edges and gradients extracted from respectively (4(b)) and (4(a)). Only 10% of the gradients is shown.

Figure 4: Graphical illustration of the edge and gradient extraction as part of the localization algorithm.

only. The binary image, resulting from 4(a), is shown in figure 4(b). Edges are detected as black-white transitions.

The calculation of the HT can be improved if also the orientation of the edges is available. This information is captured by the gradient of the grayscale image at the edge's location. Sobel's masks are applied to this extent. Such an implementation is referred to as a generalized Hough transform[8] (GHT). Figure 4(c) shows the extracted edges (crosses) and part of the gradients (arrows).

2.2 Reduced Hough Transform

The basic idea of the generalized Hough Transform is combining the position and gradient information of an edge. This yields a remaining set of possible parameter combinations, describing the particular shape. An accumulation array stores how many times each combination is suggested by one of the edge points. Local peaks in the accumulation array indicate a high probability of the presence of the searched shape with that specific combination of parameters.

The equations of a circle and its gradient are given by respectively (1) and (2). (x_c, y_c) denotes the circle's center, (x, y) one of the extracted edge points and R the radius. These equations can be rewritten in a more useful form: (3) and (4). For any possible radius, the corresponding center can be calculated. This forms a line in the parameter space, indicating all combinations of x_c, y_c and R suggested by the specific edge point. Figure 5 illustrates the 3D accumulation array constructed from the edges shown in 4(c). Local maxima emerge where (x_c, y_c, R) corresponds to an actual circle in the image.

$$\mathcal{C} \leftrightarrow (x - x_c)^2 + (y - y_c)^2 = R^2 \quad (1)$$

$$m = \frac{dy}{dx} = -\frac{\partial \mathcal{C}/\partial x}{\partial \mathcal{C}/\partial y} = -\frac{x - x_c}{y - y_c} \quad (2)$$

$$x_c = x + \frac{m}{\sqrt{1 + m^2}} R \quad (3)$$

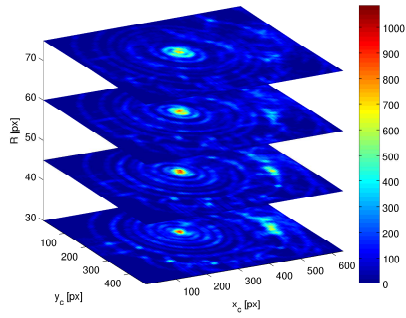$$y_c = y - \frac{1}{\sqrt{1 + m^2}} R \quad (4)$$

Figure 5: Graphical representation of the accumulation array after the generalized HT (based on the edges from figure 4(c)). Each section corresponding to one of the actual circles has a local maximum at the marker center.

The 3-dimensional parameter space is only useful to extract both the center and radius of all circles. However, localization only involves extracting the center coordinates. Moreover, the circles are concentric which can be exploited to obtain a more robust estimation of the center. This is done by eliminating R from the previous set of equations, giving (5). This implementation is further referred to as the reduced Hough transform.

$$L \leftrightarrow x_c = m\,(y - y_c) + x \qquad (5)$$

This way of computing has several advantages. The 3-dimensional parameter space is now reduced to only 2 dimensions, leaving out the unimportant radial information. This not only reduces the required memory but also fuses the position information of the different circles. It also simplifies the actual localization. In case of the standard HT, all peaks require extraction one by one. The reduced HT displays only one dominant peak which is faster and easier to detect. As an example, figure 6(a) presents the 2D accumulation array. The dominant peak in the center indicates the marker. The smaller peak on the right emerges due to a large local concentration of edge points, but is only suggested half as many times.

Because of the reduction from multiple local maxima to only one, an implementation similar to the fast Hough transform[9] (FHT) is very suitable. It iteratively searches for peaks while increasing the amount of detail. This way, only interesting regions are carefully examined. The inherent drawback of the FHT is the need for a threshold, separating useless from interesting regions. Since the marker will cause just one very strong peak, the algorithm should only care about the overall largest value, so the threshold becomes superfluous. This implementation further decreases the required amount of memory and speeds up the detection. The fast implementation is visualized in figure 6(b).
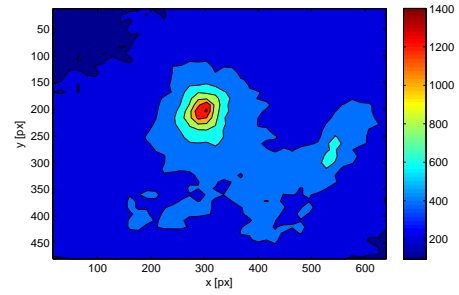
(a) Accumulation array of the reduced Hough transform, based upon the edges from figure 4(c). The central large peak emerges from the marker. The smaller peak on the right side comes from the local exessive amount of non-marker edge points. The latter has only half the magnitutde of the central peak.
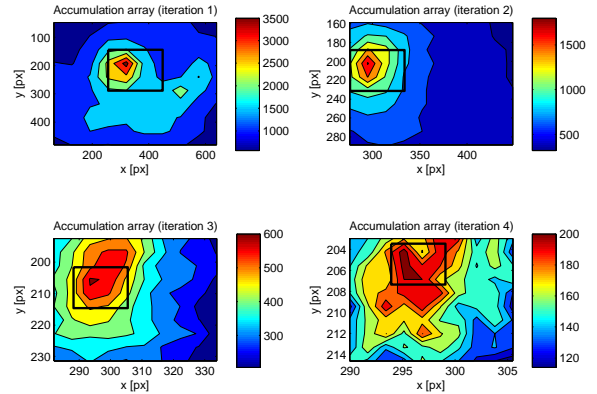
(b) Iterative implementation of the reduced Hough transform, based on a fast Hough transform implementation. The estimation of the marker position is gradually refined, as indicated by the black boxes.

Figure 6: Graphical illustration of the marker-localization step as part of the localization algorithm.

3   MARKER CLASSIFICATION

In computer vision, classification means labeling the detected objects in an image. In this case, there are only two labels: marker or non-marker. The ability to distinguish between those two labels can be used to filter a badly extracted marker center or verify if a landing platform is present in the scene.

Two distinct measurements (features) are chosen to decide on the eventual label. The first verifies how well the succession of circles matches the expectations. The second compares the amount of retrieved edge points on each circle with the expected amount. Optimally combining these two features is done with a support vector machine (SVM).

Since both features rely strongly on the individual circles, a modified histogram of the radii is constucted. This allows for computing the proposed features, presented in the following 2 sections. An optimal classifier is trained in the last section.

3.1 Histogram of radii

As a first step in extracting the described features, the individual circles are identified. This is done by constructing a histogram of the distances from the edge points to the estimated marker center. Local peaks emerge at distances corresponding to actual circle radii because all points on a circle have the same distance to the center. The histogram of the sample picture (figure 4(a)) is given in figure 7.

One would expect a series of exponentially separated peaks, but this is not the case. There are two main reasons. First, the amount of extracted edge points per circle is proportional to its radius, so the histogram shows a linearly ascending behavior. Dividing each cell by the corresponding radius flattens the histogram, equalizing the height of the peaks. Second, absolute deformation of the observed circles, due to an angle between the marker plane and image plane, is proportional to this angle and also its radius. This deformation is therefore more pronounced for larger circles, affecting the histogram. A suitable way to solve this problem is to use an exponential grid instead of the standard linear grid. The size of each cell increases proportionally to the corresponding radius, allowing more absolute deformation at greater radii. Knowing the ratio of subsequent radii on the marker allows for choosing the base of the exponential grid so that peaks emerge at regular intervals, e.g. one every five cells.

The modified counterpart of the standard histogram is shown in figure 8. The local maxima emerge at regular intervals (one every five cells), are more distinct and easier to extract.
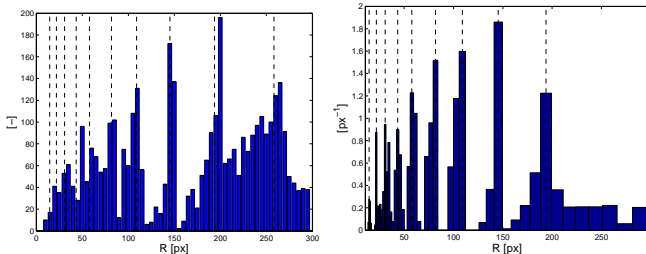

Figure 7: Standard histogram of raw data with linear separation of its cells. The dashed lines indicate the estimated radii.

Figure 8: Exponentially separated histogram, including all measures presented in 3.1. The dashed lines estimated the predicted radii.

3.2 Feature 1: circle separation

A first suitable feature for the marker checks the separation between subsequent circles. This is where another advantage of the exponential histogram shows up. Since the amount of cells in between two peaks is fixed, this comes down to measuring the periodicity in the histogram. An excellent tool is the Discrete Fourier Transform (DFT), since periodicity causes a local maximum at the corresponding frequency. However, a direct application of the DFT yields suboptimal results due to aliasing and differing begin and end

values of the histogram.

In order to improve this basic technique, the histogram is cross-correlated with a series of gaussian functions, which separation matches the expected separation of the peaks in the histogram. The resulting sequence has a high value if the gaussians are aligned with the peaks in the histogram. Therefore, it helps distinguishing between the marker and other peaks. The cross-correlation is illustrated in figure 9. Since the periodicity is more outspoken at larger radii, only these are further analyzed (colored red). In order to avoid boundary effects, the sequence is copied and mirrored such that the analyzed signal is smooth. The power spectral density of the periodic sequence is shown in figure 10. It can be seen that more than 50% of the total energy is concentrated at the expected frequency, compared to only 25% when the DFT is applied directly. This power density corresponding to the separation of subsequent circles is used as a first feature: F_1.
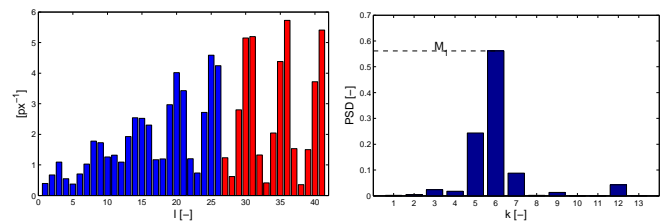

Figure 9: Cross-correlation of the histogram and a series of gaussians. The part with the most outspoken sinusoidal behavior is colored red.

Figure 10: PSD of the cross-correlation, rescaled w.r.t. the total power density after removal of the DC component.

3.3 Feature 2: amount of edge points

The second feature compares the amount of edge points per circle to the expectation. The former is directly retrievable from the histogram, the latter is to be calculated from the circle's radius in the image. In the ideal case, they are equal and their ratio becomes 1. In a real application, this ratio will drop, but it is expected to be larger when an actual marker is detected than in absence of marker. Averaging this ratio over n subsequent circles results in a more stable measurement, hence this average is taken as a second feature: F_2.

It can be proven this average is proportional to the value of the previously calculated cross-correlation at the index corresponding to radius R. The scaling factor depends solely on the amount of circles included in the average.

3.4 Training the classifier

The previously extracted features are now combined in such a way that the success rate of the classification is maximized. This is done by extracting both features from a set of images containing the marker and a set of random images. Figure 11 shows the distribution of the two classes. It can be seen that the classes are linearly separable, meaning that a

line can be drawn in between the classes (solid line).

Of the many classification techniques that exist, a linear support vector machine is most suitable in this case. Here, it comes down to a linear combination of the features. If the resulting value is positive, the object is classified as a marker, otherwise it is not. The optimization problem is formulated such that the majority of samples has a value larger than 1 in case of a marker-image and -1 in case of a non-marker image. This is called the margin and is indicated by the dashed lines. Figure 13 gives an overview of the localization of the marker on different sample pictures of 96x96 pixels, as well as their feature values and the SVM's output. If the latter is positive, the object is classified as a marker. For the images with marker, it can be seen that a good localization ($e_{tot} < 1\,px$) yields the correct label. However, the combination of a small marker and bad ambient conditions such as unsharpness and shadow, cause incorrect localization. The extraction is therefore correctly labeled as non-marker. This leaves two mistaken classifications: images 2 and 13, the first being caused by bad lighting, the second by blur. Samples 17 to 20 show how the classifier behaves when the marker is replaced by another round object. The algorithm estimates the marker center to be near the actual circle center. All are classified correctly except image 18. This is because of the high amount of concentric circles makes the object very similar to the actual marker.
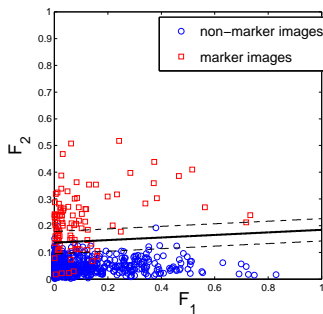

Figure 11: Suport Vector Machine, separating the marker and non-marker classes. F_1 and F_2 represent features described in 3.2 and 3.3 resp.
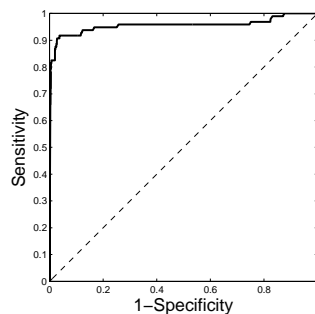
Figure 12: ROC-curve of the Support Vector Machine when varying the classification threshold.

4   PERFORMANCE ASSESSMENT

This part verifies the overall performance of the algorithm. This does not only involve tracking accuracy but also robustness against non-uniform lighting and blur, as well as dependence on the size of the marker in the image.

Four basic pictures of 480x640 pixels are used, consisting of the marker on a white background. Throughout the images, the size of the marker is varied: 19, 48, 90 and 152 pixels. Shadow and overexposure are mimicked by altering the contrast as well as the average intensity of half of the marker.

(a) $r = 19\,px$
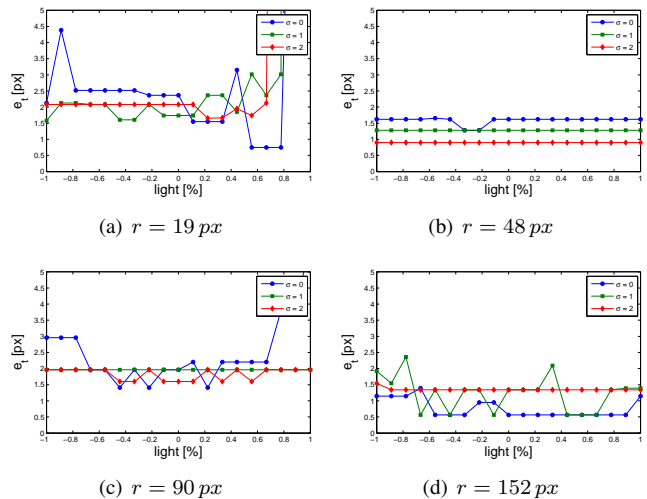
(b) $r = 48\,px$

(c) $r = 90\,px$

(d) $r = 152\,px$

Figure 14: Performance of the localization algorithm under varying lighting, sharpness and distance.

Blur is added by applying a gaussian filter to the images. The resulting degree of unsharpness depends on the variance of the gaussian, σ_G. 19 levels of lighting are treated, going from complete darkness to complete overexposure, along with 3 levels of blurring (σ_G going from 0 to 2), adding up to a total of 228 analyzed articfial images. Examples of these images are found in figure 13 (images 9 to 16).

The presented algorithm is then used to extract the marker from each image. By providing ground truth coordinates for the marker center, the actual accuracy is determined. Figures 14(a) to 14(d) graphically present the results.

The conclusions are threefold. First, the average accuracy appears to be fairly stable when the marker becomes smaller, which indicates a high overall signal-to-noise ratio for the detected marker points. Although objects in a real scene might cause the algorithm to choose a wrong position, the marker will be accurately detected if no other (circular) objects are overwhelming it. Second, moderate non-uniform lighting has little effect, thanks to locally thresholding the image. In the extreme case, where half of the marker is invisible, localization is still acceptable, with total errors ranging from approximately $1\,px$ to $3\,px$ for an average to large marker. However, when the marker becomes too small, the remaining information is insufficient to localize the marker, resulting in large errors. This can be seen in figure 14(a). Third, blur has little effect and can be even advantageous: it stabilizes the outcome of the algorithm. It reduces the high frequency noise, which is desirable in feedback applications.

5   IMPLEMENTATION ON A QUADROTOR

The developed algorithm forms the core of a positioning system, used for automatic docking of a quadrotor. As a quadrotor frame, the DJI f450 is used, which is fur-

1) $M_1=0.12$, $M_2=0.04$
SVM$=-2.47$, $e_{tot}=28.7$ px

2) $M_1=0.11$, $M_2=0.04$
SVM$=-2.33$, $e_{tot}=18.7$ px

3) $M_1=0.66$, $M_2=0.20$
SVM$=0.79$, $e_{tot}=1.1$ px

4) $M_1=0.03$, $M_2=0.27$
SVM$=3.09$, $e_{tot}=0.5$ px

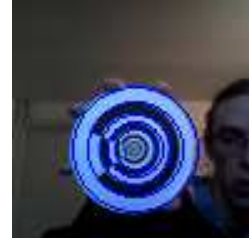5) $M_1=0.05$, $M_2=0.11$
SVM$=-0.61$, $e_{tot}=1.1$ px

6) $M_1=0.03$, $M_2=0.28$
SVM$=3.38$, $e_{tot}=0.7$ px

7) $M_1=0.00$, $M_2=0.28$
SVM$=3.42$, $e_{tot}=0.9$ px

8) $M_1=0.05$, $M_2=0.34$
SVM$=4.83$, $e_{tot}=0.8$ px

9) $M_1=0.07$, $M_2=0.03$
SVM$=-2.68$, $e_{tot}=32.8$ px

10) $M_1=0.49$, $M_2=0.05$
SVM$=-2.64$, $e_{tot}=7.2$ px

11) $M_1=0.00$, $M_2=0.24$
SVM$=2.49$, $e_{tot}=0.2$ px

12) $M_1=0.08$, $M_2=0.28$
SVM$=3.40$, $e_{tot}=0.3$ px
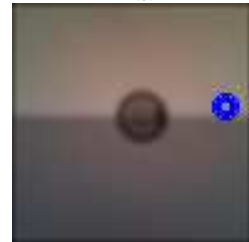
13) $M_1=0.00$, $M_2=0.12$
SVM$=-0.51$, $e_{tot}=1.5$ px

14) $M_1=0.19$, $M_2=0.32$
SVM$=4.04$, $e_{tot}=0.6$ px

15) $M_1=0.26$, $M_2=0.05$
SVM$=-2.40$, $e_{tot}=41.4$ px

16) $M_1=0.55$, $M_2=0.06$
SVM$=-2.53$, $e_{tot}=32.4$ px

17) $M_1=0.20$, $M_2=0.11$
SVM$=-0.96$

18) $M_1=0.04$, $M_2=0.37$
SVM$=5.57$

19) $M_1=0.01$, $M_2=0.08$
SVM$=-1.41$

20) $M_1=0.07$, $M_2=0.08$
SVM$=-1.45$

Figure 13: Classification of various images, with and without markers.

ther equipped with the PX4FLOW camera. The camera comes with an ultrasonic distance sensor, which is used to accurately measure the quadrotor's height up to two meters. PX4FLOW is also provided with an ARM processor which is programmed to run an optical flow algorithm. By reprogramming it to execute the marker detection algorithm, a stand-alone positioning module is created. The module can be easily transfered to any other type of VTOL UAV and transmits estimates of the marker's position in image coordinates at $25\,Hz$.

Ardupilot 2.5 is used as autopilot. It is equipped with an integrated barometer which is used to estimate height when the ultrasonic distance sensor is out of range. Via an i^2c connection, the raw position data are read from the PX4FLOW unit and further processed.

The quadrotor arrives at its destination with a base in its viewing range. The algorithm detects the marker, allowing the quadrotor to estimate its position and use it to land on the target.

However, as depicted in figure 15, altering the quadrotor's angles in space causes the marker to shift in the image, while the quadrotor is at the same position. Also changes in the focal length and the distance to the object have to be taken into account. This is referred to as the calibration of the camera and makes it possible to derive a 3D position based on the marker's image coordinates and the distance to the ground. This information is used to estimate the velocity and acceleration of the quadrotor w.r.t. the base which is later used to control the quadrotor's position. Both calibration of the camera and control of the quadrotor are carried out by the autopilot.
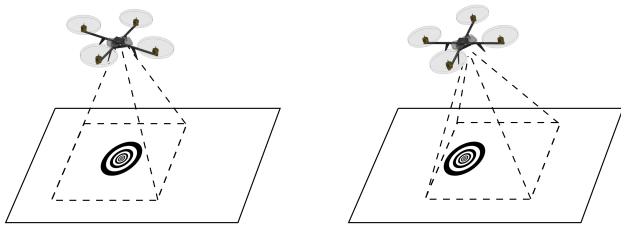
(a) Quadrotor with zero roll and pitch angles. The camera points straight to the ground, resulting in an image as indicated by the dashed square.

(b) Quadrotor with non-zero roll and pitch angles. The base is shifted in the image compared to figure 15(a), but the quadrotor's position in space remains the same.

Figure 15: Illustration of the need for compensation for non-zero roll, pitch and yaw angles.

5.1 Calibration

Camera calibration is generally split up in internal and external calibration. The internal calibration relies on camera parameters only. Modern digital cameras are usually sufficiently accurately described by a simple model, containing coordinates of the principal point and pixel mapping parameters. The transformation from image plane to camera coordinates is given in equation 6. $_c\vec{p}^{c,m}$ denotes a vector $\vec{p}$, pointing from the camera, c, to the marker, m, expressed in the camera's coordinate system. K represents the internal calibration matrix[10] and maps individual pixels to world coordinates. Since the system is underdetermined, it requires some distance measure to be known in advance, usually the distance to the object d.

$$_c\vec{p}^{c,m} = d\,K^{-1}\vec{u} \qquad (6)$$

The external calibration compensates for the camera's position w.r.t. the quadrotor's center of gravity, as well as the quadrotor's angles in space. The previously calculated coordinates in the camera frame are subsequently transformed to a stabilized quadrotor frame and fixed world frame.

The orientation of the stabilized quadrotor frame differs from the world frame only by a yaw angle. Computing the relative position of the marker therefore involves compensating for the roll and pitch angles as well as the relative positioning w.r.t. the camera. The transformation is captured by equation 7. $_c\vec{p}^{c,q}$ denotes the position vector from the camera to the quadrotor's center of gravity, expressed in the camera coordinate system. $_q^cR$ is the rotation matrix, transforming a vector expressed in {c} to {q}. It depends solely on the quadrotor's roll and pitch angles. They are also important for the basic angular control of the quadrotor and are already being calculated from the raw inertial measurement unit (IMU) data.

$$_q\vec{p}^{q,m} = {}_q^cR\left({}_c\vec{p}^{c,m} - {}_c\vec{p}^{c,q}\right) \qquad (7)$$

Transforming the position from quadrotor to world coordinates only requires compensating for the yaw angle. This is expressed in equation 8. $_w^qR$ is the rotation matrix taking the yaw angle into account. The yaw angle is measured by the onboard magnetic compass. Considering control in the world frame, it is more suitable to express the base-to-quadrotor position, rather than the other way around, hence the minus in (8).

$$_w\vec{p}^{m,q} = -{}_w^qR_q\vec{p}^{q,m} \qquad (8)$$

5.2 Filtering

Directly providing the measured position as feedback to the position controller is not feasible. This is because the marker position is determined by accumulating cells in the image plane which inherently causes high frequency quantization noise. State estimation is therefore necessary to have stable estimates of the velocity and acceleration. To this purpose, a Kalman estimator is used.

In outdoor applications, wind will exert a secondary force on the quadrotor. The positioning system enables the Kalman filter to estimate the equivalent acceleration. This is done by adding this acceleration to the state-space of the quadrotor.

5.3 Preliminary results

Preliminary flight tests show that the measurement system enables the quadrotor to hold its position within a $10\,cm$ radius of the position setpoint (figure 16) at a height of $1\,m$. Note that the position measurements are taken directly from the quadrotor. In another experiment, a fan exerts an external force on the quadrotor. Figure 17 shows the external force, estimated by the Kalman filter. This force is compensated for by the controller so stable positioning is possible in windy conditions.
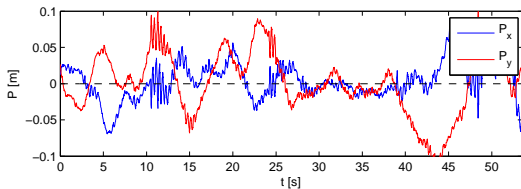

Figure 16: Positioning accuracy of the quadrotor, measured with the developed positioning system at a height of $1\,m$.
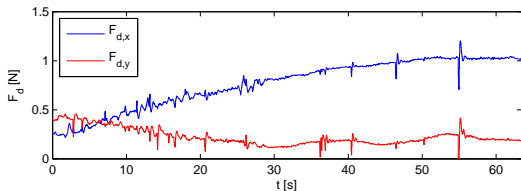

Figure 17: Estimation of the external disturbance of a fan, aligned with the world frame's *x*-axis. The estimate of the force builds up slowly to a steady value of $1\,N$.

During a final experiment, the quadrotor carries out six consecutive indoor landings. The eventual positioning errors are shown in table 5.3. The positioning system enables the UAV to land within a $5\,cm$ radius of the target.

Landing	1	2	3	4	5	6
e_x	2.18	3.12	0.76	0.00	0.66	-1.23
e_y	2.93	-0.85	-3.79	1.80	-3.31	-4.07
e_t	3.65	3.24	3.86	1.80	3.38	4.25

Table 1: Position errors after six successive indoor landings in cm.

6 CONCLUSION

The presented vision-based localization algorithm is able to accurately extract the newly developed marker from an image. Local thresholding and the application of an adapted Hough transform both add robustness to the algorithm. The repetitive structure allows detection of both small and large markers in the image, which is of great importance during landing. Also robustness against blurring and non-uniform lighting is proven. These properties make the algorithm especially suited for use outdoors, where varying lighting is one of the main issues. Apart from localization, validation of the target is also taken into account. This enables the UAV to assess whether a landing platform is present or not.

The developed algorithm is implemented on a stand-alone module that can be easily transfered to any other type of VTOL UAV and transmits estimates of the marker's position in image coordinates at 25Hz. This new position measurement system allows a quadrotor to autonomously maintain a steady position and land within a $5\,cm$ radius of the target. Knowledge of the position makes it also possible to actively estimate and counteract the external disturbances.

REFERENCES

[1] Bernard Michini, Tuna Toksoz, Joshua Redding, Matthew Michini, Jonathan How, Matthew Vavrina, and John Vian. Automated battery swap and recharge to enable persistent uav missions. *Proceedings of the Infotech@Aerospace 2011*.

[2] Karl Engelbert Wenzel, Andreas Masselli, and Andreas Zell. Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle. *Journal of Intelligent and Robotic Systems*, 61(1-4):221–238, 2011.

[3] Wei Wang, Zongliang Li, Wenpeng Yu, and Jianwei Zhang. An autonomous docking method based on ultrasonic sensors for self-reconfigurable mobile robot. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 1744–1749. IEEE, 2009.

[4] F Tong, SK Tso, and TZ Xu. A high precision ultrasonic docking system used for automatic guided vehicle. *Sensors and Actuators A: Physical*, 118(2):183–189, 2005.

[5] Companionable - mobile robot companion. smart home. http://www.companionable.net/, october 2013.

[6] Paul Hough. Method and means for recognizing complex patterns, 1962.

[7] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11:285–296, 1975.

[8] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[9] Nicolas Guil, Julio Villalba, and Emilio L. Zapata. A fast hough transform for segment detection. *Image Processing, IEEE Transactions on*, 4(11):1541–1548, 1995.

[10] Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/, april 2014.