UAV manuscript No.
(will be inserted by the editor)

Vision based Position Control for MAVs using one single Artificial Landmark

Daniel Eberli · Davide Scaramuzza · Stephan Weiss · Roland Siegwart

Abstract This paper presents a real-time vision based algorithm for 5 degrees-of-freedom pose estimation and set-point control for a Micro Aerial Vehicle (MAV). The camera is mounted on-board a quadrotor helicopter. Camera pose estimation is based on the appearance of two concentric circles which are used as landmark. We show that that by using a calibrated camera, conic sections, and the assumption that yaw is controlled independently, it is possible to determine the six degrees-of-freedom pose of the MAV. First we show how to detect the landmark in the image frame. Then we present a geometric approach for camera pose estimation from the elliptic appearance of a circle in perspective projection. Using this information we are able to determine the pose of the vehicle. Finally, given a set point in the image frame we are able to control the quadrotor such that the feature appears in the respective target position. The performance of the proposed method is presented through experimental results.

Multimedia Material

Please note that this paper is accompanied by a high definition video which can be found at the following address:
`http://www.youtube.com/watch?v=SMFR2aFR2E0`
In this video, the algorithm described in this paper is used for automatic take-off, hovering, and landing. The navigation part is performed using another approach which is described in [1]. In the video, you can also see the landmark described in this paper, composed of two concentric black and white circles. The rectangular box surrounding the landmark denotes the search area used to speed up the tracking of the landmark.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 231855 (sFly). Daniel Eberli is currently Master student at the ETH Zurich. Davide Scaramuzza is currently senior researcher and team leader at the ETH Zurich. Stephan Weiss is currently PhD student at the ETH Zurich. Roland Siegwart is full professor at the ETH Zurich and head of the Autonomous Systems Lab.

D. Eberli · D. Scaramuzza · S. Weiss · R. Siegwart
ETH Autonomous Systems Laboratory, 8092, Zurich, Switzerland, www.asl.ethz.ch
E-mail: eberlid@student.ethz.ch, davide.scaramuzza@ieee.org, stephan.weiss@mavt.ethz.ch, rsiegwart@ethz.ch

The detection, conversely, is done using the entire image. The cables visible are used for streaming the images to the laptop and for securing the helicopter during flight.

1 Introduction

In this paper, we describe a real-time vision based pose estimation algorithm and a set-point control strategy for a quadrotor helicopter. As we use a single camera, the pose must be determined by the appearance of a known shape object in the camera image.

Because of the high processing power restrictions and due to the high frequency of the control updates needed to stabilize the helicopter, we focus on the implementation of an efficient feature detection and pose estimation algorithm. Furthermore, the algorithm should be robust to view point and illumination changes, motion blur, and occlusions. As will be shown, the framework we developed is able to control the quadrotor in hovering robustly.

This paper is organised as follows. In section 2 the related work is presented. Section 3 describes the feature detection algorithm. The computation of the pose and the visual servoing algorithms are described in sections 4 and 5. The experimental results are presented in section 6 while the conclusions are given in 7.

2 Related Work

In computer vision, the problem of estimating the 6 Degrees-of-Freedom (DoF) motion of a camera from a set of known 2D-3D point correspondences is known as the Perspective-n-Point (PnP) problem [2], where n is the number of 3D points which are used to estimate the pose of the camera. By knowing the exact relative position of the points, this problem can be solved from a minimum of 3 points; however at least 4 points are actually required to avoid ambiguities. Some applications of point based flight control for MAVs have been reported by [3] and [4]. Other kinds of camera pose estimation methods rely on line correspondences [5–7] or constrained lines [8,9], while others use closed shapes like rectangles or circles [10–13]. The main drawback of PnP and line based approaches is however their high sensitivity to image noise and motion blur. As an example, a few pixel error reflects onto a position error of several centimeters or decimeters (depending on the distance to the object) with bad effects on the stability of the MAV. Furthermore, point occlusions and wrong point correspondences make the task even harder.

For the purpose of controlling a micro flying robot, appearance based visual servoing approaches have been also used [3]. For our application, we chose a circular landmark. A circle is indeed straightforward to detect and the estimation of its parameters can be done quite robustly even in presence of partial occlusions or blur. By exploiting the fact that a circle is mapped onto an ellipse in the camera frame, we developed a geometric and intuitive approach for estimating the pose of the camera. As will be also shown, by using two concentric circles instead of just one makes it possible to disambiguate the multiple solutions of the problem and determine the pose of the camera in 5 DoF, which is actually enough to stabilize and control our helicopter.

3 Feature Detection

The approach described in this section is a complete framework to calculate the 5 DoF pose of a camera from the appearance of two concentric circles in the image frame. The landmark we used is composed of two concentric circles (the smaller one is white while the other one black).

The algorithm starts by an adaptive image thresholding which classify pixels into black and white. Next it computes all connected pixel components. Finally some criteria, like image moments, are used to identify the landmark (see later).

If we look at the circle from a viewpoint not lying on its normal axis, the circle will appear as an ellipse whose parameters can be calculated. In addition the ray bundle outgoing from the camera optical center to the circle contour points form the shell of an oblique circular cone (Fig. 1). Considering the ellipse parameters, conic section theory and the position of the ellipse in the image frame, we can determine how much the camera coordinate system is tilted with respect to the world coordinate system.

A concentric smaller circle is needed to disambiguate between the two possible solutions of the problem. It also allows us to compute the xy coordinates of the vehicle with respect to a set point in the image frame. By knowing the size of the cirle, we can also calculate the height of the camera above the circle. Finally, given a set point, the quadrotor can be controlled with a LQG-LTR approach in a stable mode.

3.1 Feature Detection Algorithm

In this section, we describe how we identify our elliptical features (blobs).

The detection algorithm is based on a growing of interest regions approach [14]. First, from each row of the image we extract so-called line blobs. A line blob is defined as a sequence of pixels where the intensity value of each pixel lies in the range $[thres_l, thres_h]$. The goal is to set the thresholds such that the intensity values of the black circle are within this range while those of the white circle and the white background are not.

As described in algorithm 1, the frame is analyzed row-wise pixel by pixel. If the threshold check for a pixel in the i-th row and in the j-th column returns $true$, the start column col_s is set equal to j. Then the algorithm increments j until a pixel does not satisfy the threshold check. The end of the line blob col_e is set equal to $j-1$. The data is stored in a map $imgData$ whose elements are vectors with elements like (col_s, col_e). In a second step, every line blob is checked neighboring blobs, in which case the two will be fused into a single blob. In contrast to the one dimensional line blob, a blob is two dimensional. The scanning process let a blob grow by including the line blob which is touching the blob. At the end of the inspection procedure, we get a number of isolated regions whose intensity values are in $[thres_l, thres_h]$.

We now have to select the blob which corresponds to the appearance of the black circle. Because the feature we are searching for has an elliptical shape, we consider scale invariant image moments up to third order. The raw moments are calculated by iterating through n lines of the respective blob where they are first initialized to zero

Algorithm 1 Line Blob Detection

```
procedure LineBlob
    for $i \leftarrow 0, frame_{height} - 1$ do
        for $j \leftarrow 0, frame_{width} - 1$ do
            if $thresCheck = true$ then
                $col_s \leftarrow j$
                while $thresCheck = true$ do
                    $col_e \leftarrow j$
                    $j \leftarrow j + 1$
                end while
                $imgData\,[i]\,.pushback\,(col_s, col_e)$
            end if
            $j \leftarrow j + 1$
        end for
        $i \leftarrow i + 1$
    end for
end procedure
```

and afterwards updated as follows:

$$M_{00} = \sum_{_f y=1}^{n} {_f x_2} -_f x_1$$

$$M_{10} = \sum_{_f y=1}^{n} \frac{_f x_2 +_f x_1}{2} \left({_f x_2} -_f x_1\right)$$

$$M_{01} = \sum_{_f y=1}^{n} y \left({_f x_2} -_f x_1\right)$$

$$M_{11} = \sum_{_f y=1}^{n} y \frac{_f x_2 +_f x_1}{2} \left({_f x_2} -_f x_1\right)$$

$$M_{20} = \sum_{_f y=1}^{n} \frac{2_f x_2^3 + 3_f x_2^2 +_f x_2}{6} - \frac{2_f x_1^3 - 3_f x_1^2 +_f x_1}{6}$$

$$M_{02} = \sum_{_f y=1}^{n} {_f y^2} \left({_f x_2} -_f x_1\right)$$

$$M_{30} = \sum_{_f y=1}^{n} \left({_f x_2} \frac{_f x_2 + 1}{2}\right)^2 - \left(\left({_f x_1} - 1\right) \frac{_f x_1}{2}\right)^2$$

$$M_{03} = \sum_{_f y=1}^{n} {_f y^3} \left({_f x_2} -_f x_1\right).$$

Where $_f x_1$ and $_f x_2$ are respectively the minimum and maximum horizontal coordinate of the respective line with vertical coordinate $_f y$. The central moments are obtained by the well known relations (see [15]):

$$
\begin{aligned}
\mu_{00} &= M_{00} \\
\mu_{10} &= 0 \\
\mu_{10} &= 0 \\
\mu_{20} &= M_{20} -_f \overline{x} M_{10} \\
\mu_{02} &= M_{02} -_f \overline{y} M_{01} \\
\mu_{30} &= M_{30} - 3_f\overline{x} M_{20} + 2_f\overline{x}^2 M_{10} \\
\mu_{03} &= M_{03} - 3_f\overline{y} M_{02} + 2_f\overline{y}^2 M_{01}
\end{aligned}
$$

where ${}_f\overline{x}$ and ${}_f\overline{y}$ are the centroid coordinates of the respective blob. The scale invariant moments are obtained by applying:

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}.$$

As a matter of fact, the third order moments of an ellipse are zero and the normalized area (1) of the ellipse represented by the invariant moments is equal to 1 for a perfect ellipse. Furthermore, we calculate the perimeter (2) of the ellipse represented by the raw moments [16].

$$A_m = 4\pi\sqrt{\eta_{02}\eta_{20} - \eta_{11}^2} \tag{1}$$

$$P_m = \pi\left(a_m + b_m\right)\left(1 + \frac{3\lambda_1^2}{10 + \sqrt{4 - 3\lambda_1^2}}\right) \tag{2}$$

where

$$
\begin{aligned}
\phi_m &= \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \\
C_1 &= 8 \cdot \frac{M_{11} -_f \overline{x} \cdot_f \overline{y} \cdot M_{00}}{M_{00}\sin\left(2\phi_m\right)} \\
C_2 &= M_{00}\cos\left(\phi_m\right)^2 \\
C_3 &= M_{00}\sin\left(\phi_m\right)^2 \\
C_4 &= 4\left(M_{20} - \overline{x}^2 M_{00}\right) \\
a_m &= \sqrt{\frac{C_4 + C_1C_3}{C_2 + C_3}} \\
b_m &= \sqrt{\frac{C_4 - C_1C_2}{C_2 + C_3}} \\
\lambda_1 &= \frac{a_m - b_m}{a_m + b_m} \\
\delta_1 &= \frac{\mu_{20} + \mu_{02}}{2} + \frac{\sqrt{4\mu_{11}^2 + \left(\mu_{20} - \mu_{02}\right)^2}}{2} \\
\delta_2 &= \frac{\mu_{20} + \mu_{02}}{2} - \frac{\sqrt{4\mu_{11}^2 + \left(\mu_{20} - \mu_{02}\right)^2}}{2} \\
\epsilon &= \sqrt{1 - \frac{\delta_2}{\delta_1}}
\end{aligned}
$$

Finally, our two-concentric-circle based feature is accepted as correct if it satisfies simultaneously the following conditions, whose parameters were found empirically:

1. M_{00} of the black blob bigger than a given threshold
2. A_m of the black blob in a given range
3. A_m of the white blob in a given range
4. difference between ϵ of the black and ϵ of the white blob in a given range
5. η_{30} and η_{03} of the black blob smaller than a threshold
6. ratio of M_{00} of the black blob and M_{00} of the white blob in a given range

If the threshold for the criteria are set properly, the algorithm will not detect a false positive in a natural scene. Once the landmark is detected, we restrict the search to a rectangular region around the last position in order to speed up the algorithm (see the bounding box shown in the a video described at the beginning of this paper).

3.2 Calculating Ellipse Parameters

Once the two concentric circles – which may appear as ellipses in the projection plane – are detected, n contour points ${}_f(x, y)$ of the outer black circle are transformed into n three dimensional vectors ${}_c\boldsymbol{x}^j$ (where $j = \{1 \dots n\}$) representing the directions of appearance of the respective pixels (Fig. 1). They are stored in a $M^{3\times n}$ matrix whose column entries corresponds to ${}_c\boldsymbol{x}^j$. The ${}_c\boldsymbol{x}^j$ are computed by using the camera calibration toolbox described in [17,18]. The obtained ${}_c\boldsymbol{x}^j$ represent the shell of a virtual oblique circular cone which is illustrated in Fig. 1.

A rotation matrix R is composed (4), such that the average vector (3) satisfies the condition (5). Due to the fact that we only know the directions of appearance, we need to normalize them such that they all have the same z coordinate α_0 which is the zeroth-order coefficient of the camera calibration coefficients.

$$ {}_c\overline{x}_i = \frac{\sum_{j=1}^{n} M_{ij}}{n}, \quad i = \{1, 2, 3\} \tag{3}$$

$$\begin{aligned}
\alpha &= -atan2\left({}_c\overline{x}_2, -{}_c\overline{x}_3\right) \\
\beta &= -atan2\left({}_c\overline{x}_1, -{}_c\overline{x}_2 \sin\alpha - {}_c\overline{x}_3 \cos\alpha\right)
\end{aligned}$$

$$\begin{aligned}
R_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \\
R_2 &= \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \\
R &= R_2\ R_1
\end{aligned} \tag{4}$$

$$\begin{bmatrix} 0 \ 0 \ z_0 \end{bmatrix}^T = R\ {}_c\overline{\boldsymbol{x}} \tag{5}$$

$$\hat{M} = R\ M \tag{6}$$

$$N_{ij} = \frac{\hat{M}_{ij}}{\hat{M}_{3j}} \alpha_0, \quad \begin{array}{l} i = \{1, 2, 3\} \\ j = \{1 \dots n\} \end{array} \tag{7}$$

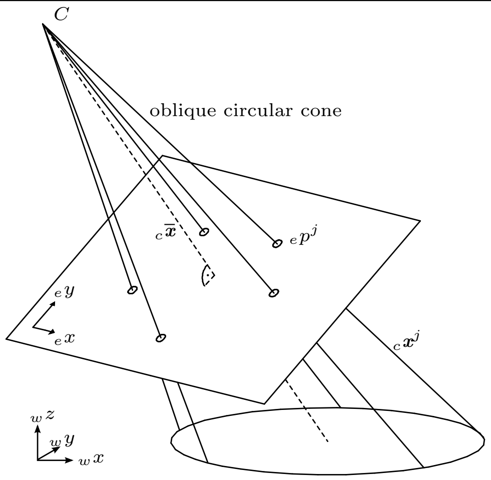


Fig. 1 Ray bundle outgoing from the camera C to the circle's contour points forms an oblique circular cone.

The x and y coordinates of the j normalized contour points (7) represent the intersections of ${}_c\boldsymbol{x}^j$ with a plane normal to ${}_c\overline{\boldsymbol{x}}$ through ${}_c\,(0,0,\alpha_0)$ (8). Therefore we introduce the coordinate system of the ellipse plane denoted e with origin in ${}_c\,(0,0,\alpha_0)$.

$$
{}_e p_i^j = N_{ij}, \qquad \begin{matrix} i = \{1,2\} \\ j = \{1 \dots n\} \end{matrix} \tag{8}
$$

The ${}_e p^j$ describe an ellipse which is defined by an implicit second order polynomial (9). The polynomial coefficients are found by applying a numerically stable direct least squares fitting method [19] on

$$
F\,(x,y) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{9}
$$

with an ellipse specific constraint

$$
b^2 - 4ac < 0.
$$

The a,b,c,d,e,f are the ellipse parameters and ${}_e\,(x,y)$ are the coordinate points obtained by the normalization procedure above. The result of the least squares fitting are two vectors

$$
a_1 = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, a_2 = \begin{pmatrix} d \\ e \\ f \end{pmatrix}
$$

containing the parameters which satisifes the least squares condition.

The conic section equation (9) can be transformed with the substitution

$$
\begin{aligned}
x &= u\cos\phi - v\sin\phi \\
y &= u\sin\phi + v\cos\phi
\end{aligned}
$$

and the well known definition

$$
\tan 2\phi = \frac{b}{a-c}
$$

into a form where the cross coupled terms disappear. In a geometric meaning this is a derotation of the ellipse such that the main axis is parallel to the x axis.

$$\begin{aligned} F(u,v) &= Au^2 + Bv^2 + Cu + Dv + E = 0 \\ A &= c + \frac{b}{2\tan\phi} \\ B &= c - \frac{1}{2} b \tan\phi \\ C &= d\cos\phi + e\sin\phi \\ D &= -d\sin\phi + e\cos\phi \\ E &= f \end{aligned}$$

With the substitution

$$\begin{aligned} u &= \eta - \frac{C}{2A} \\ v &= \zeta - \frac{D}{2B} \end{aligned}$$

the ellipse is translated to the origin of the coordinate system which yields the form

$$\eta^2 A + \zeta^2 B = \left(\frac{C^2}{4A} + \frac{D^2}{4B} \right) - E. \tag{10}$$

Dividing (10) by is right-hand-side we get the well known normalized ellipse equation

$$\frac{\eta^2}{\chi^2} + \frac{\zeta^2}{\psi^2} = 1. \tag{11}$$

where

$$\begin{aligned} \chi &= \sqrt{\frac{C^2B + D^2A - 4EAB}{4A^2B}} \\ \psi &= \sqrt{\frac{C^2B + D^2A - 4EAB}{4AB^2}}. \end{aligned}$$

4 5 DoF Pose Estimation

To estimate the tilt, we introduce an identity matrix I_3 which represents the initial configuration of the camera coordinate system. The goal is to find a rotation matrix R_c such that ${}_wC = R_c I_3$. The only information available to find R_c is the appearance of the two circles in the image frame. In the first step, we analyze the ellipticity and orientation of the ellipse. In a second step, we consider the position of the appearance of the ellipse.

Considering Fig. 2 and the conic section theory we know that the intersection of a right elliptic cone with height h and the ellipse (11) as base (where $\chi > \psi$) with a plane P yields a circle with radius r. The normal axis of P lies in the yz plane and takes

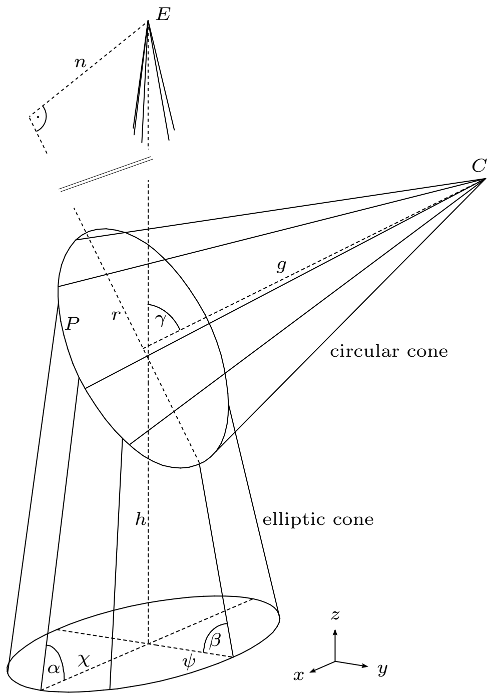


Fig. 2 Intersection of a right elliptic cone with a plane P (whose normal axis lies in the yz plane and takes an angle γ w.r.t the xz plane) yields a circle.

an angle γ with respect to the xz plane. The distance between E and the intersection plane is n.

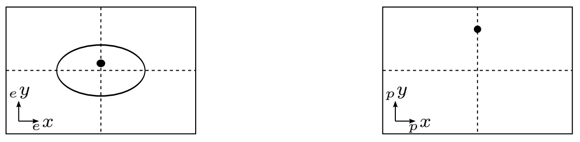
$$\cos\gamma = \frac{\cos\beta}{\cos\alpha}, \qquad \begin{aligned} \alpha &= \pi/2 - \arctan\chi/h \\ \beta &= \pi/2 - \arctan\psi/h \end{aligned} \tag{12}$$

$$r = n\frac{\chi^2}{\psi h} \tag{13}$$

$$h = \alpha_0. \tag{14}$$

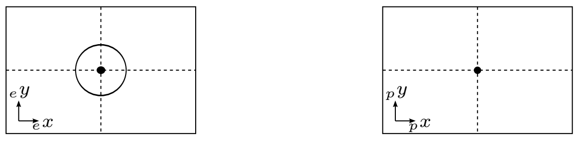
For an intuitive geometric interpretation, assume C (in Fig. 2) as the camera single view point and the base of the circular cone as the physical blob. In this configuration, the camera lies in the normal axis g of the outer circle, therefore the circle appears as a circle. Shifting the camera position from g causes the appearance of the circle to become an ellipse. Considering E as the shifted camera position, the circle appears as (11).

For an illustration in more detail, consider Fig. 3a as the obtained ellipse in the ellipse plane with the appearance of the centre of the inner of the two concentric circles marked with a black dot. The black dot in Fig. 3b represents the intersection point of ${}_c\overline{\boldsymbol{x}}$ with the projection plane as it is illustrated as D in Fig. 6. Note that the black dot in Fig. 3a does not lie in the centre of the ellipse. This is due to the fact that the appearance of the physical centre of a circle does not coincide with the centre of its appearance in a perspective projection. This allows us to distinguish between



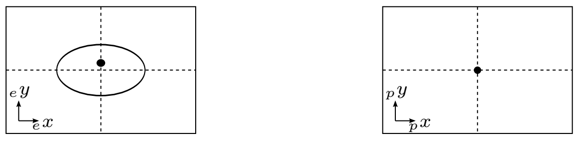
(a) Ellipse plane (b) Projection plane

Fig. 3 Considering C in Fig. 1 as the camera view point. Contour of the outer circle with the appearance of the circle's centre point (Fig. 3a) and the position of the centre point in the projection plane (Fig. 3b).



(a) Ellipse plane (b) Projection plane

Fig. 4 Assuming observing the circle from a up right position.



(a) Ellipse plane (b) Projection plane

Fig. 5 After applying R_a the ellipse in Fig. 5a has its desired shape and orientation. But the optical axis goes still trough the centre of the appearance (Fig. 5b). A rotation R_b forces the centre to appear in the desired position showed in Fig. 4b.

the identical solutions ϕ and $\phi + \pi$ (considering only the shape and orientation of the ellipse).

The initial situation is showed in Fig. 4 as the data obtained by an observation of the circle from a position lying in g and the camera coordinate's xy plane coplanar with respect to the xy plane of the world coordinate system. The identity matrix I_3 represents this initial pose. As the first operation, (19) is composed as a rotation around χ with γ. This is shown in Fig. 2 represented by a change of the view point from C to E. Indeed this is a big issue because of the undefined main axes of a circle and the fact that (12) will never be 1 in an attempt. It means that this approach does not work in practice if the camera lies in g. How to avoid this problem is showed later on. Note that this pose of the camera plane is illustrated in Fig. 5. The centre point D appears in the optical centre of the projection plane and the ellipse has its desired shape, orientation and physical centre point position.

To bring D from the position given in Fig. 5b to the one defined by Fig. 3b we have to introduce an additional operation (21) which completes the procedure by a rotation with δ around s. The axis s is defined as the vertical to $\overline{OD}$ (the reader may notice κ as the angle of $\overline{OD}$ with respect to the $_ex$ axis) and (15) as the angle between the optical axis and $_c\overline{\boldsymbol{x}}$.

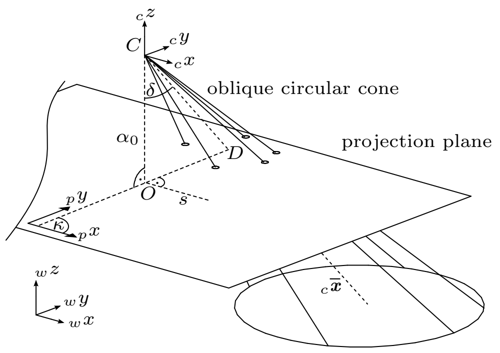


Fig. 6 The projection plane is perpendicular to the optical axis. The rotation R_b is defined by the rotation axis s and the angle δ.

Now all information given by the obtained data in Fig. 3 is used. The tilt of the camera coordinate system with respect to the world coordinate system is completely described by $_wC = R_c I_3$ where $R_c = R_b R_a$.

$$\delta = \arccos\left(\frac{_c\overline{x}_3}{\|_c\overline{\boldsymbol{x}}\|}\right) \tag{15}$$

$$\kappa = atan2\left(_c\overline{x}_2, {_c\overline{x}_1}\right) \tag{16}$$

$$\epsilon = \kappa - \pi \tag{17}$$

$$R_\phi = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$R_a = R_\phi^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} R_\phi I_3 \tag{19}$$

$$R_\epsilon = \begin{bmatrix} \cos\epsilon & -\sin\epsilon & 0 \\ \sin\epsilon & \cos\epsilon & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

$$R_b = R_\epsilon^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\delta & -\sin\delta \\ 0 & \sin\delta & \cos\delta \end{bmatrix} R_\epsilon I_3 \tag{21}$$

The position of the camera optical center in the world coordinate system with respect to the centre point of the circle can be estimated by first calculating the height above the circle plane by solving (13) for (23). The radius r of the circle is measured by hand and is the only scale information for the entire process. To calculate the relative position of the camera with respect to the centre of the circle in the x and y coordinates of the world coordinate system, a derotation (22) is carried out, such that the configuration of the camera coordinate system and $_c\hat{\overline{\boldsymbol{x}}}$ can be regarded as if the xy plane of the camera coordinate system is coplanar with respect to the xy plane of the world coordinate system.

$$ {}_c\hat{\bar{\boldsymbol{x}}} = R_b R_a {}_c\bar{\boldsymbol{x}} \tag{22} $$

$$ n = rh\frac{\psi}{\chi^2} \tag{23} $$

Regarding the blob's centre point as the origin of the world coordinate system and ${}_c\boldsymbol{o} = [0, 0, -1]^T$ as the optical axis, the position of the camera can be expressed as (26). The scaling (24,25,27) is needed to obtain the position of the respective points in the xy plane of the world coordinate system.

$$ {}_c\tilde{\boldsymbol{o}} = n\frac{{}_c\boldsymbol{o}}{{}_co_3} \tag{24} $$

$$ {}_c\tilde{\boldsymbol{x}} = n\frac{{}_c\hat{\bar{\boldsymbol{x}}}}{{}_c\hat{\bar{x}}_3} \tag{25} $$

$$ \begin{bmatrix} {}_wc_1 \\ {}_wc_2 \\ {}_wc_3 \end{bmatrix} = \begin{bmatrix} {}_c\tilde{o}_1 - {}_c\tilde{x}_1 \\ {}_c\tilde{o}_2 - {}_c\tilde{x}_2 \\ n \end{bmatrix} \tag{26} $$

Where we considered ${}_f(r_x, r_y)$ as the position in the image frame where the feature should appear. The corresponding three dimensional vector ${}_c\boldsymbol{r}$ represents the desired direction of appearance. The set point coordinate (28) stands for the position of the camera in the world coordinate system.

$$ {}_c\tilde{\boldsymbol{r}} = n\frac{{}_c\boldsymbol{r}}{{}_cr_3} \tag{27} $$

$$ \begin{bmatrix} {}_wq_1 \\ {}_wq_2 \\ {}_wq_3 \end{bmatrix} = \begin{bmatrix} {}_c\tilde{o}_1 - {}_c\tilde{r}_1 \\ {}_c\tilde{o}_2 - {}_c\tilde{r}_2 \\ n \end{bmatrix} \tag{28} $$

5 Visual Servoing

The knowledge of the camera and the set point coordinates in the world fixed coordinate system allows us to control the quadrotor with an observer based LQR-LTR approach [1]. Since we assume that the yaw is controlled independently (and in fact this is done by the low level controller of our MAV) we are able to control the quadrotor in all six degrees of freedom.

As mentioned in 4 the calculation of the pose does not work if the camera is in a up right position of the blob. Assuming the quadrotor to be in a horizontal pose for hovering (or slightly tilted of very small angles), we can set $R_c = I_3$. This means that we neglect the tilt of the camera plane. However, experimental results have shown that this approach is not suitable in this case. Hence, we retrieved the tilt angle of the MAV from the on-board IMU and filled it directly into the R_c matrix. By doing so, we were able to control the quadrotor even with non-neglectable tilt angles.

6 Results

The camera used in our experiments is the *uEye* camera from IDS with a 60° field-of-view lens. As a first experiment, we evaluated the accuracy of our pose estimation algorithm.

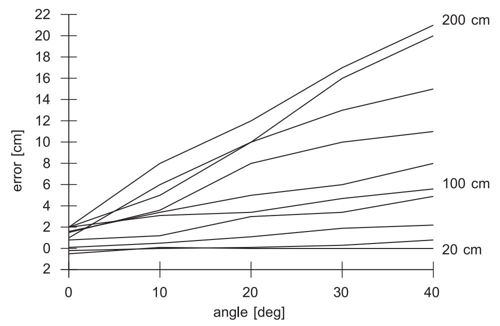


Fig. 7 Height error w.r.t. angle, $r = 4.5$ cm.

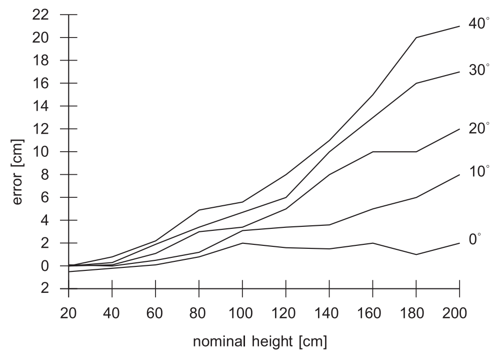


Fig. 8 Height error w.r.t nominal height, $r = 4.5$ cm.

Measurement errors from a height of 20 cm up to 200 cm with a step size of 20 cm and a circle radius $r = 4.5$ cm are shown in Fig. 7. The smallest error is obtained for the smallest nominal height. The error is positive, linear with respect to the angle of appearance and, as shown in Fig. 8, also with respect to the height. The angle shown in the plots is the angle of sight of the blob from the camera point of view. In Fig. 9 the relation of the measured height with respect to the nominal height is plotted. The knowledge of this plot allows us to conclude that at higher heights, due to motion blur and to the smaller appearance of the circle, the circle's contour is no longer precisely detectable.

As for the computation time, our algorithm is very quick. On a single core Intel Pentium M 1.86 GHz the overall algorithm takes 16 ms. The same algorithm was also tested on a Intel Core Duo 3 GHz CPU. Here, the calculation time decreased to a few milliseconds.

The stability of the LQG-LTR controller approach presented in [1] was tested in hovering position during 15 s. A landmark with a radius of $r = 2.75$ cm was placed on the floor. The set point was set straight above the blob at a desired height h. We performed the experiment by testing several heights $(20, 30, 40, 50, 60, 70, 80$ and 90 cm$)$.

In Fig. 10 the RMS value (in centimeters) of the norm of the error vector is plotted against the height. The higher the camera is, the smaller the blob appears. At high heights, it is no longer possible to detect the contour of the black circle precisely. This fact yields uncertainties in the calculation of the pose and introduces noise as visible

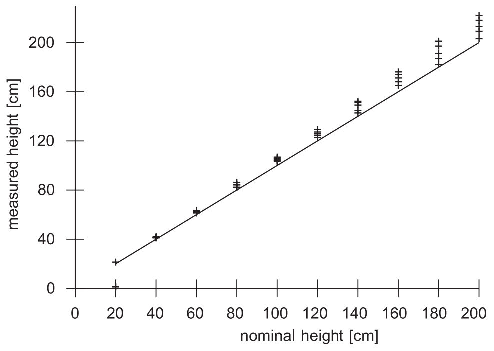


Fig. 9 Measured height w.r.t. nominal height, $r = 4.5$ cm. The line represents the ground truth.

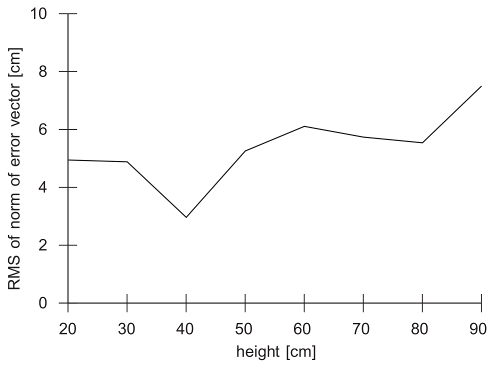


Fig. 10 RMS of norm of error vector w.r.t height, black blob radius $r = 2.75$ cm.

from the plot. A bigger blob would attenuate this problem. Indeed, the bigger the blob, the higher the limit of stable hovering.

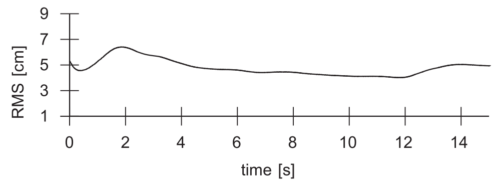
The xy trajectory of the measured position data over time at a nominal height of 80 cm is shown in Fig. 12.

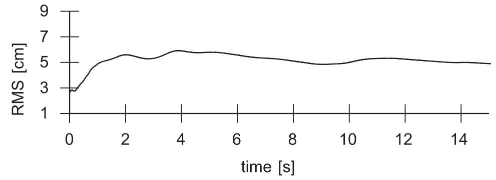
7 Conclusion

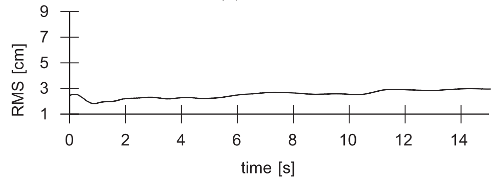
In this paper, we presented a full vision based framework to estimate the 5 degrees-of-freedom pose of a camera mounted on a quadrotor helicopter. As a landmark, we used two concentric circles. A description of the feature detection algorithm was given, and we illustrated how to determine the pose of the camera with respect to the detected blob by considering conic section theory and camera calibration.

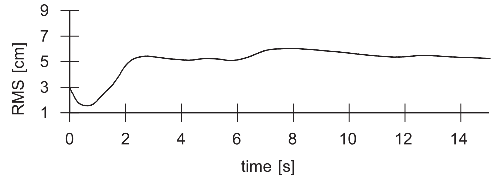
An analysis of the pose estimation errors was made. The behavior of the error allows us to control the quadrotor with a LQG-LTR controller approach [1]. The experiments have shown that the quadrotor is able to hover in a range of 20 cm up to 90 cm if the blob radius is 2.75 cm. The use of a bigger blob results in a higher operation height.

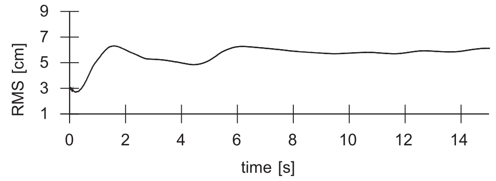
The analysis of the RMS values for different heights has shown that the quadrotor is able to hover stably at a given set point. Cycle rate measurements have shown that the whole framework works also on lower powered systems sufficiently fast and robustly.

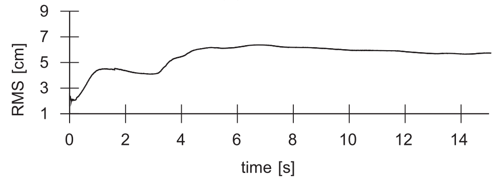

(a) h= 20 cm

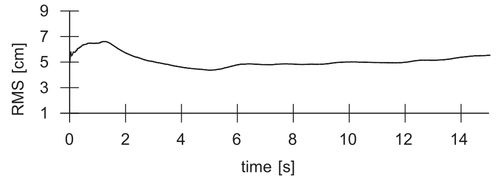

(b) h= 30 cm


(c) h= 40 cm


(d) h= 50 cm


(e) h= 60 cm


(f) h= 70 cm


(g) h= 80 cm

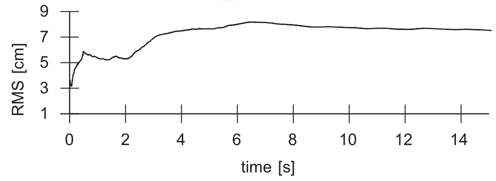

(h) h= 90 cm

Fig. 11 RMS in time plots for different quadrotor heights h above the circles plane. The radius of the black circle is $r = 2.75$ cm.

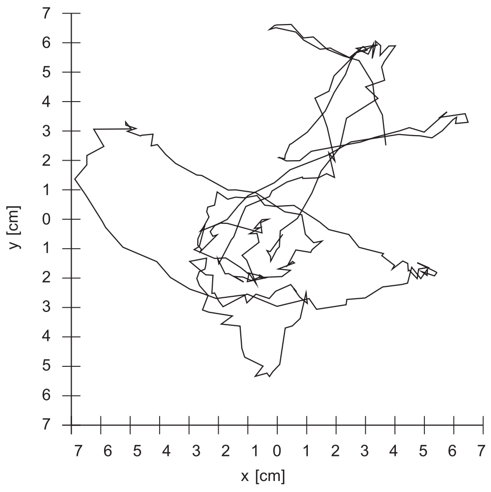


Fig. 12 Trajectory in the xy plane, black blob radius $r = 2.75$ cm, nominal height $h = 80$ cm.

References

1. M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments," in *IEEE International Conference on Robotics and Automation (ICRA'10), Anchorage, 2010.*, 2010.
2. M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
3. T. Hamel, R. Mahony, and A. Chriette, "Visual servo trajectory tracking for a four rotor vtol aerial vehicle," in *International Conference on Robotics and Automation*, 2002.
4. T. Cheviron, T. Hamel, R. Mahony, and G. Baldwin, "Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of uav," in *International Conference on Robotics and Automation*, 2007.
5. M. Dhome, M. Richetin, J. Laprest, and G. Rives, "Determination of the attitude of 3d objects from a single perspective view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1265–1278, 1989.
6. Y. Liu, T. S. Huang, and O. D. Faugeras, "Determination of camera location from 2-d to 3-d line and point correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 28–37, 1990.
7. A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 578–589, 2003.
8. R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* New York, NY, USA: Cambridge University Press, 2000.
9. G. Wang, H.-T. Tsui, Z. Hu, and F. Wu, "Camera calibration and 3d reconstruction from a single view based on scene constraints," *Image and Vision Computing*, vol. 23, no. 3, pp. 311–323, March 2005.
10. G. Jiang and L. Quan, "Detection of concentric circles for camera calibration," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, Oct. 2005, pp. 333–340 Vol. 1.
11. J.-S. Kim, P. Gurdjos, and I.-S. Kweon, "Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 637–642, 2005.
12. X. M. Hua, H. Li, and Z. Hu, "A new easy camera calibration technique based on circular points," in *TABLE 3 Input Images, Edge Map, and the Estimated Calibration Matrix.* Press, 2000, pp. 1155–1164.
13. G. Wang, J. Wu, and Z. Ji, "Single view based pose estimation from circle or parallel lines," *Pattern Recogn. Lett.*, vol. 29, no. 7, pp. 977–985, 2008.

14. E. van Kempten. [Online]. Available: http://geekblog.nl/entry/24
15. F. Chaumette, “Image moments: A general and useful set of features for visual servoing,” *IEEE Transactions on Robotics*, 2004.
16. R. Lee, P.-C. Lu, and W.-H. Tsai, “Moment preserving detection of elliptical shapes in gray-scale images,” *Pattern Recogn. Lett.*, vol. 11, no. 6, pp. 405–414, 1990.
17. D. Scaramuzza, A. Martinelli, and R. Siegwart, “A toolbox for easy calibrating omnidirectional cameras,” in *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006.
18. D. Scaramuzza, “Omnidirectional camera calibration toolbox for matlab, first release 2006, last update 2009, google for “ocamcalib”.”
19. R. Halir and J. Flusser, “Numerically stable direct least squares fitting of ellipses,” 1998.