# A Cross-Platform Comparison of Visual Marker Based Approaches for Autonomous Flight of Quadrocopters

A. Masselli, S. Yang, K. E. Wenzel and A. Zell

*Abstract*— In this paper, we compare three different marker based approaches for six degrees of freedom (6DOF) pose estimation, which can be used for position and attitude control of micro aerial vehicles (MAV). All methods are able to achieve real time pose estimation onboard without assistance of any external metric sensor. Since these methods can be used in various working environments, we compare their performance by carrying out experiments across two different platforms: an AscTec Hummingbird and a Pixhawk quadrocopter. We evaluate each method's accuracy by using an external tracking system and compare the methods with respect to their operating ranges and processing time. We finally compare each method's performance during autonomous takeoff, hovering and landing of a quadrocopter.

## I. INTRODUCTION

Micro aerial vehicles (MAV) have become a major trend in robotics research. Kumar et al. outlined the opportunities and challenges of this fast-growing field in [10] . While controllers of MAVs already show an amazing performance, most systems lack autonomy and depend on global localization based on external sensors or processing. The goal is to eliminate this dependency and to enable MAVs navigating through the environment using only onboard hardware. This is still a challenging task because of the limited payload.

Common pose estimation methods for MAVs use artificial landmarks. Such methods play an important role for landing of MAVs, when normally MAVs require accurate pose estimation related to a specified target. In our previous work, we developed three different methods for this task.

Andreas Masselli, Shaowu Yang and Karl E. Wenzel are PhD students with the Chair of Cognitive Systems at the Faculty of Science, University of Tuebingen, Tuebingen, Germany {`andreas.masselli, karl.e.wenzel, shauwo.yang`}`@uni-tuebingen.de`

Andreas Zell is a full professor of Cognitive Systems at the Faculty of Science, University of Tuebingen, Tuebingen, Germany `andreas.zell@uni-tuebingen.de`

In [18], we estimate the pose using a WiiMote camera combined with a group of infrared (IR) markers. It is a low cost solution for autonomous flight of quadrocopters. But the working environment of this method is limited to indoor scenarios with existing IR markers. In [21], our monocular vision method for pose estimation was shown to work robustly in cluttered environments with a regular landing pad, which consists of a letter "H" surrounded by a circle. This method would enable autonomous flight of MAVs with more flexibility, at the expense of much more computational effort. We also developed a method for pose estimation based on a pattern of four table tennis balls [12], which has moderate computational cost and is independent of IR markers as used in [18], while sensitive to color information in the environment.

These methods use different approaches to achieve the same task. Therefore it is instructive to comprehensively compare their performance in both accuracy and processing time, to see which of the methods should be chosen for which scenario. Thus we chose two popular and very different quadrocopter platforms, as will be described in section III, to evaluate the performance of those methods for enabling autonomous flight of MAVs.

After presenting related work in section II, we describe our quadrocopters in section III and our visual marker based approaches in section IV. In section V, we test all methods using ground truth data from an external tracking system and conclude our results in section VI.

## II. RELATED WORK

In Mellinger et al. [14] the ability to perform quick and precise movements of quadrocopters is shown, using external pose estimation based on infrared markers on the MAVs as well as an external flight controller. Low level flight commands are

sent back to the MAV wirelessly. Wenzel et al. [18] demonstrated an intermediate system for visual tracking, whose IR markers are on the ground and all sensing and processing is done onboard of a quadrocopter. This method has been improved and successfully tested with the pattern mounted on a moving platform [19]. Xu et al. [20] also use a cooperative object and an infrared camera for pose estimation of MAVs, but only the yaw angle is computed. The technology presented by Grzonka et al. [6] enabled autonomous quadrotor navigation in indoor environments using a modified 2-D laser range finder setting.

Previous work on controlling UAVs (unmanned aerial vehicles) visually mainly focuses on the landing task, since landing is the most challenging part of a flight, requiring precise pose estimation of the UAV relative to the landing target. In [16][15][11] and[19] methods incorporating pre-defined targets have been successfully used. Hovering and landing control of a UAV on a large textured moving platform enabled by measuring optical flow is achieved in [7]. Other work focuses on visually finding a suitable area for landing [5][1].

Saripalli et al. [16] solved the landing task of a helicopter quite early by using image moments for object recognition and estimating the relative position to the landing pad with precise height of the helicopter provided by differential GPS. However, this approach is bound to fail in either cluttered or GPS-denied environments. A distinct landing pad with five circle triplets in different scale was designed by Merz et al. [15] in order to estimate the relative pose in a coarse-to-fine process. Lange et al. [11] achieved autonomous landing and position control of an MAV by estimating the relative position from a landing pad consisting of several concentric circles. This approach only works as long as it is guaranteed that the MAV is flying exactly parallel to the landing pad.

### III. OUR QUADROCOPTERS

We evaluated the performance of the three techniques on two types of quadrocopters of different weight and onboard processing capabilities. This way we could prove a method's portability, and could also compare the differences of the methods with the effect of changing the platform.

*1) AscTec Hummingbird:* Our first platform to test and compare our methods was a Hummingbird made by Ascending Technologies GmbH (*AscTec*), Munich, Germany (see Fig. 1). It is known for its high agility, shown in [14]. With the Wi-iMote/Microcontroller setup used for the first technique, it weighs 480 g including battery. With the PointGrey FireFly/Gumstix setup for the second method it has a total weight of 570 g.



Fig. 1.   AscTec Hummingbird

*2) Pixhawk Cheetah:* Our second test platform was a Pixhawk Cheetah quadrocopter developed at the ETH Zurich [13]. It is equipped with an Intel Core 2 duo processor with 1.8 GHz and weighs 1370 g with our setup (see Fig. 2), including battery and the two cameras. While carrying more computing power, it is less maneuverable than the Hummingbird, mainly caused by its heavier weight.

Both MAVs are controlled by four independent, classical proportional-integral-derivative (PID) controllers via a serial interface to maintain a constant position and yaw.

### IV. METHODS

The following sections describe the three methods for visual pose estimation in detail:

#### A. IR-LED Pattern and WiiMote

The follwing method was formerly used in [19] and has been further developed. It uses a pattern of four infrared LEDs, which is tracked by a single camera taken from a Nintendo Wii controller. From the pixel positions of the IR spots, all 6 degrees

Fig. 2.    Pixhawk Cheetah

of freedom are determined using an 8 bit ATmega microcontroller.

*1) Spot Recognition:* A pattern of four high-power infrared LEDs was constructed as an artificial landmark, measuring 38 cm in diameter. The LEDs are coplanar and parallel to the ground plane. Three of them form an isosceles triangle, one is used as a redundant spot for determining a confidence (see below). It is tracked by an infrared camera from the Nintendo Wii controller, which communicates via $I^2C$ and already provides the pixel positions of four infrared light sources at once. Because of its small size and weight the camera is especially suited as an onboard sensor for MAVs. Having a physical resolution of 128×96 pixels, the coordintates of the IR spots are returned with a resolution of 1024×768 pixels using eight times subpixel analysis.

The field of view (FOV) of the camera is about 42 degrees in diagonal. A lightweight pan/tilt unit was constructed with two miniature servo motors. This unit enlarges the operating range by directing the camera towards the IR-pattern. In order to estimate the current servo angle, two analog inputs of the microcontroller were connected to the potentiometer of each servo motor. A calibration function from voltage measurements gives precise and fast results compared to estimating the servo position over time by means of the PWM signal.

*2) Retrieving 6DOF Pose:* With the image coordinates of four points and the knowledge of the corresponding points' positions in the world/pattern frame, the full pose of the camera and therefore of the quadrotor can be determined relative to the pattern. The underlying problem is known in literature as the *Perspective-4-Point Problem*, or *P4P*, in general *PnP* for *n* points: Given *n* points in world coordinates and the corresponding coordinates in a camera image which the points are projected on, the task is to find the set of feasible camera poses. For *n* = 4 patterns can be found such that in general there is only one unambiguous solution. We solve the problem in the following manner: Since the operating range is restricted, i.e. we can assume that the quadrocopter is flying in an upright position and in front of the LED pattern, we can find the correspondences by simply sorting the four image points along their x-coordinates of the camera image to identify them as *L*, *M*, *R* and *V*. The next step is solving the P4P, for which we first consider the points *L*, *M* and *R*, technically solving a P3P, and then adding *V* to verify the solution. In P3P, up to four solutions can be found [3]. We can already exclude three solutions by considering the operating range again and therefore gain a full pose estimate using only three markers. However, with the 4th marker we add redundancy to achieve a confidence measurement, successfully rejecting most of the false positive detections of the pattern.

P3P is solved using an own method developed to run in real time on the microcontroller. We compared it to the method introduced in [9] and found it to be about 20 times faster while being comparably accurate. However, it provides only one solution and is restricted to symmetric patterns using isosceles triangles, which is appropriate for our application. Given the points *L*, *M* and *R* in world coordinates and their corresponding image points $L_i$, $M_i$ and $R_i$, we first want to determine the camera's position *P* in world coordinates. In a second step the orientation can be inferred, if needed.

By applying a standard pinhole camera model, we can determine the light rays *l*, *m*, *r* going through $L_i$, $M_i$ and $R_i$ and meeting in *P* in camera coordinates. Let *m'* be the projection of *m* onto the plane spanned by *l* and *r*, so we can calculate the angles $\alpha = \angle(l, r)$, $\beta = \angle(m', r)$ and $\gamma = \angle(m', m)$.

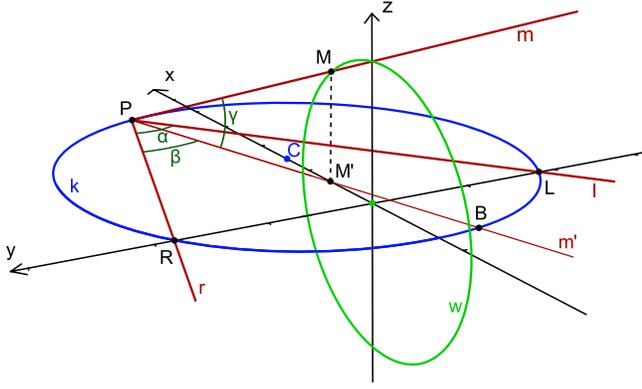We now define a right-handed intermediate coordinate system as follows (see Fig. 3):



Fig. 3. Scheme of the solving approach to the P3P problem.

Let the origin be the middle of $L$ and $R$, the $y$-axis go through $L$ and $R$ pointing towards $R$, and the $x, y$-plane go through $L$, $R$ and $P$ with the $x$-axis being directed such that $P_x \geq 0$.

Using the inscribed angle theorem, we know that $P$ must lie on a circle $k$ with center $C(\frac{y_R}{tan\alpha}, 0, 0)$ and radius $d(C, R)$.

Using the inscribed angle theorem a second time, we find that $m'$ intersects $k$ in $B$, independent of where $P$ on $k$ is located. $B$ can directly be constructed by rotating $R$ around $C$ by $2\beta$.

From the definition of our intermediate coordinate system and from the symmetry of the pattern, we know that $M$ lies on a circle $w$ in the $x, z$-plane with the origin as center and the offset of $M$ in world coordinates as radius $r_w$.

We therefore know that

$$z_M = \sqrt{r_w^2 - x_M^2}. \tag{1}$$

Since $M$ lies on $m$, we can also derive that

$$z_M = d(P, M')tan\gamma, \tag{2}$$

with $M'(x_M, 0, 0)$ being the foot of $M$ in the $x, y$-plane. With $m'$ going through $B$ and $M'$, $P$ can be described dependent on $x_M$ by intersecting $m'$ with $k$ (and picking the solution unequal to $B$). We can finally equate (1) and (2) to determine $x_M$. Squaring and further expanding yields the quartic polynomial

$$c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0 = 0 \tag{3}$$

with $x = x_M - x_C$ and coefficients

$$c_4 = m^2 + 1$$
$$c_3 = 2(x_C - x_B)$$
$$c_2 = v + (1 - 2x_C^2)B^2 - 4x_C x_B$$
$$c_1 = 2(x_C B^2 - v x_B)$$
$$c_0 = v B^2 + m^2 (B^2)^2,$$

with $v = x_C^2 - r^2$ and $B^2 = x_B^2 + y_B^2$.

From the limited operating range it can be inferred that the largest root will yield the correct estimate. It is found using Newton's method. Back substituting $x$ will get $M$ and $P$ in the intermediate coordinate system. $P$ can then be transformed into world coordinates by simply rotating around the $y$-axis by $\epsilon = atan2(z_M, x_M)$.

Finally, a confidence is calculated by re-projecting all four points based on the current pose estimate and determining the euclidean distances $e_i$ to the actual measurements in the image plane. Assuming Gaussian noise, the confidence $c$ then equals:

$$c = e^{-\frac{\sum e_i}{\sigma^2}}. \tag{4}$$

The confidence value can be used to determine the Kalman Gain in order to filter the pose estimate using a Kalman Filter,

### B. Table Tennis Balls and Color Segmentation

In this method the active IR-LEDs from the first method are replaced by passive markers. Four orange table tennis balls are glued onto a black piece of cardboard to form a distinct pattern, measuring 40 cm in diameter. Live images are taken with a PointGrey FireFly USB 2.0 color camera with VGA resolution (640×480), global shutter, and capable of 60 fps. Together with the lens, it has a diagonal FOV of about 48 degrees. The camera is turned by 90 degrees to maximize the vertical FOV. This is due to the fact that we can control the yaw angle of the quadrocopter in order to point the camera towards the pattern, whereas we have no influence on the pitch angle, since pitching the quadrocopter would immediately change its position and is therefore reserved for controlling the quadrocopter with respect to the target position. A (pan/)tilt unit would also be possible, but is harder to implement compared to the WiiMote

setup, since the PointGrey camera is larger and heavier.

This method has been published in [12]. The use of passive markers has shown to be more robust in the presence of sunlight compared to the first method.

The algorithm is designed to be fast even on limited hardware. To control the *AscTec Hummingbird*, all image processing plus controlling of roll, pitch, yaw and thrust is done on a *Gumstix Overo Fire*, a single-board computer, equipped with a 600 MHz ARM Cortex-A8 CPU. It runs in real-time on the ARM processor, processing all of the 60 frames coming from the camera.

Pose estimation is done in two steps: First the table tennis balls are detected within the camera image. After that all six degrees of freedom are determined based on the image coordinates of the balls and the P4P solving algorithm described above.

*Table Tennis Ball Recognition:* We assume the table tennis balls to appear as an orange circular region with a radius of least three pixels within the image. Figure 4 shows the steps for detecting the table tennis marker balls: Each frame is received as raw image containing color information from the Bayer pattern. To ensure fast processing, we omit debayering the whole image. Instead, we only evaluate the color of a pixel when testing it. Initially, we only have to test each fourth pixel in both x and y direction to ensure a ball region of radius bigger than two is hit. If and only if the color matches the orange signal color of a ball, the surrounding region is searched pixel-wise by applying the floodfill algorithm. Floodfill also calculates a bounding box for each region incrementally. This reduction on a 4×4 search grid accelerates the region finding process by a factor of almost 16, since balls appear rarely in the image. Testing for the color is done directly on the RGB pixel values on the raw image using a binary look-up table of candidate ball colors. Each region of interest is then validated to have a feasible size. Then their surrounding contour is extracted and tested via a *Randomized Hough Transform* (*RHT*) for having a circular shape. The four largest regions passing *RHT* are taken as detections of the table tennis balls. In Fig. 4 it can be seen that

this final check discards false positives from the segmentation step and therefore ensures robustness even in cluttered environments. For the following pose estimation only the center coordinates of the bounding boxes are regarded. The size of the balls is too small to get reliable distance data.
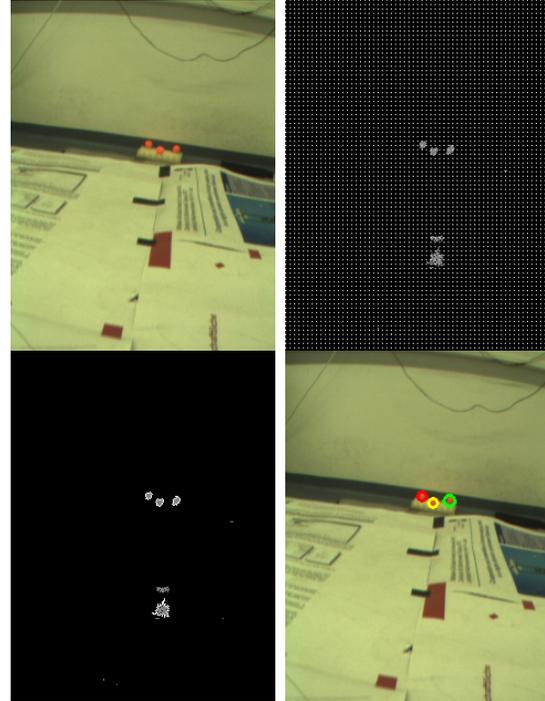


Fig. 4.    Detection of the table tennis ball pattern: Upper left: Input image from the camera Upper right: 4×4 pixel scan grid and segmented regions. Lower left: Output of contour following algorithm. Lower right: Successful detection.

### C. Landing Pad Detection

In this method, published in [21], we estimate the 6DOF pose of the quadrocoptor based on a single image of a typical landing pad which consists of the letter "H" surrounded by a circle with a diameter of 9 cm. This passive marker is natural to use for UAVs to guide approach and landing, since patterns of this type are commonly used to mark landing pads.

*1) Landing Pad Recognition:* Images are taken with a downward looking PointGrey FireFly USB 2.0 monochrome camera with VGA resolution and 60 fps. The FOV angle with the used lens is about 90 degrees. Since the landing pad is not particularly textured, whereas its background might be very cluttered, we decided to treat the recognition of it as a sign detection problem and

solve it similarly as in [17]. We use adaptive thresholding implemented using an integral image to binarize each camera image and find its connected components using blob detection. Next we classify these connected components using an artificial neural network to detect the "H" pattern. Finally we suppress false positives by enforcing that each letter "H" only belongs to a landing pad if it is surrounded by a circle.

Fig. 5 is an exemplary image processing result for landing pad recognition. It shows that different circles and letters "H" can be efficiently detected even if the perspective changes dramatically. False positives for the individual circle class or letter "H" class might appear, but would not be finally classified as a landing pad.
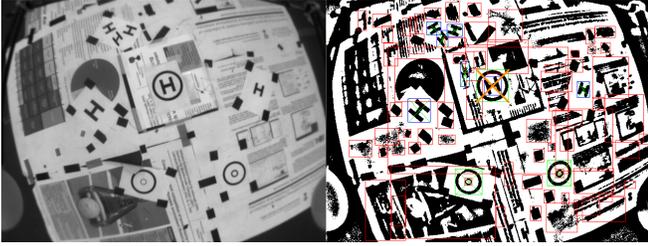


Fig. 5. Landing pad recognition results when the quadrotor hovers above the landing pad in a cluttered environment. While many "H"-patterns and circles are detected, only the proper combination is recognized.

*2) Ellipse Fitting:* After detecting the landing pad, we estimate the relative pose to the pad by obtaining the ellipse corresponding to the perspective projection of the landing pad circle and that of the letter "H". We do so by applying a Canny edge detector on the gray scale image pattern corresponding to the landing pad and fitting ellipses to these edges by implementing a so called direct least square fitting algorithm [4]. The lens distortion effect to the ellipse parameters is also taken into account during this process.

*3) 6DOF Pose from Ellipses:* We use one ellipse originating from the projection of a known circle for the 5DOF pose estimation of the quadrotor, including its 3D position $\vec{T}_W^B$ and the roll and pitch angles ($\phi$, $\theta$) of the quadrotor within the landing pad. $\phi$ and $\theta$ are derived from the normal vector of the landing pad plane. Then we extend this to a 6DOF pose estimation by using the orientation of the major axis of the ellipse fitted to

the letter "H" as yaw angle ($\psi$) approximation. We use the computational geometry results in [2] and [8] for the pose estimation, and then use an IMU-aided approach to eliminate ambiguity inherited from this method.

An ellipse in the image frame can be described by the following quadratic equation,

$$Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0. \quad (5)$$

Following the interpretation in [8], with $f$ being the focal length of the camera, we get a conic $Q$ in the form of

$$\vec{Q} = \begin{bmatrix} A & B & \frac{D}{f} \\ B & C & \frac{E}{f} \\ \frac{D}{f} & \frac{E}{f} & \frac{F}{f^2} \end{bmatrix} \quad (6)$$

We directly derive the results for the 5DOF pose of the circle in the camera frame from the conic $\vec{Q}$. Detailed proof of this result can be found in [8] and [2]. Let $r$ be the radius of the original circle, $\lambda_1$, $\lambda_2$, and $\lambda_3$ be the eigenvalues of $\vec{Q}$, and the corresponding unit eigenvectors $\vec{u}_2$ and $\vec{u}_3$. As $\vec{Q}$ has a signature of (2, 1) [8], we can assume that $\lambda_3 < 0 < \lambda_1 \leq \lambda_2$. Then the unit normal vector and the center of the landing pad described in the camera frame (denoted by $\vec{n}$ and $\vec{T}_C^W$) are given by

$$\vec{n} = S_1 \sqrt{\frac{(\lambda_2 - \lambda_1)}{(\lambda_2 - \lambda_3)}} \vec{u}_2 + S_2 \sqrt{\frac{(\lambda_1 - \lambda_3)}{(\lambda_2 - \lambda_3)}} \vec{u}_3, \quad (7)$$

$$\vec{T}_C^W = z_0 \left( S_1 \frac{\lambda_3}{\lambda_1} \sqrt{\frac{(\lambda_2 - \lambda_1)}{(\lambda_2 - \lambda_3)}} \vec{u}_2 + S_2 \frac{\lambda_2}{\lambda_1} \sqrt{\frac{(\lambda_1 - \lambda_3)}{(\lambda_2 - \lambda_3)}} \vec{u}_3 \right),$$

$$(8)$$

where $z_0 = S_3 \frac{\lambda_1 r}{\sqrt{-\lambda_2 \lambda_3}}$, and $S_1$, $S_2$ and $S_3$ are undetermined signs. As $\vec{n}$ faces to the camera, and the center of the circle is in front of the camera, we can get two constraints for the undetermined signs, then two solutions remain. If we assume that the error of the attitude estimates by the IMU is small, and the landing pad is placed horizontally oriented, the gravity vector ($\vec{g}$) calculated form this attitude estimation can provide us the third constraint: we chose $\vec{n}$ to be the one closer to $\vec{g}$. From (7) and (8) and the three constraints, we get the 5DOF pose of the circle. Together with the yaw angle calculated from the letter "H", we can obtain the 6DOF pose of the quadrotor in the landing pad frame by performing a coordinate transformation.

## V. EXPERIMENTS

Tests were performed in an indoor flying lab, equipped with a Natural Point OptiTrack[1] tracking system to provide ground truth pose data. It runs at 100 Hz, covering a test area of about $2 \times 2 \times 2\,\text{m}^3$.

We tested the methods with two scenarios: In the first the quadrocopters were remotely flown in order to evaluate the range and accuracy of the pose estimation methods. In the second scenario the task was to perform an autonomous flight consisting of take-off, hovering and landing. On the Pixhawk all methods have been implemented. Since the Hummingbird cannot provide enough computation power for our landing pad method, we only implemented the IR-pattern method and the table tennis ball method on the Hummingbird. This yielded five setups for our cross-platform comparison experiments as shown in Table I.

TABLE I

Setups for our cross-platform experiments

|  | IR-pattern | Table tennis | Landing pad |
|---|---|---|---|
| Hummingbird | × | × | n.a. |
| Pixhawk | × | × | × |

### A. Manual Flight

We performed several flights remotely controlling the quadrocopters. Hereby we made sure to cover a large part of the operating range of each method, which is usually not the case during the autonomous flight, since the controller tries to keep a constant position. In Fig. 6 we can see one representative trajectory of the flights, using the Hummingbird with the IR pattern. The $y$ direction is similar to $x$ and is omitted for clarity. The curves are well in line with the ground truth data. The other four setups had similar errors. Table II shows the mean error for three flights of each setup, defined as the absolute difference between the pose estimate of the tracking system and of each method, respectively. The table tennis method and the "H" landing pad method performed slightly better than the IR-LED method, still all approaches provide tolerable errors for our application.
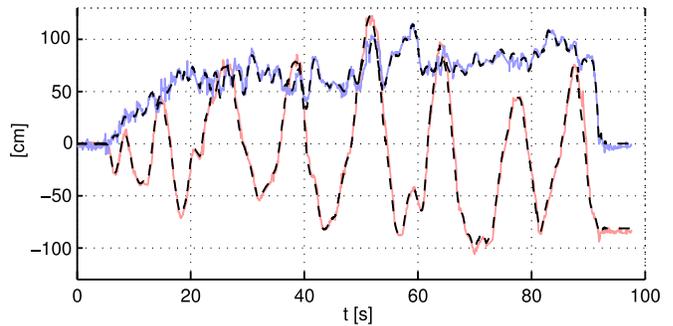
[1]http://www.naturalpoint.com/optitrack



Fig. 6. Pose estimate of the IR-LED method on the Hummingbird during manual flight, compared to ground truth. Red line: $x$. Blue line: $z$ (height). Dashed lines: Ground truth.

TABLE II

Mean error during manual flights. X, Y and Z in *cm*

|  | X | Y | Z |
|---|---|---|---|
| Hummingbird with IR-LEDs | 3.42 | 3.86 | 3.12 |
| Hummingbird with table tennis balls | 1.42 | 2.56 | 1.59 |
| Pixhawk with IR-LEDs | 3.64 | 4.18 | 3.02 |
| Pixhawk with table tennis balls | 3.44 | 3.95 | 1.82 |
| Pixhawk with landing pad | 2.20 | 1.91 | 0.44 |

### B. Autonomous Flight

We performed ten flights for each of the four setups, using the corresponding onboard estimates as input to the controller. The target position for the first three setups was 170 cm in front of the pattern at 100 cm height, for the last setup the target position was 100 cm directly over the middle of the landing pad. The mean error in $x$, $y$, $z$ and yaw was evaluated during hovering using the data of the tracking system and defining the error as the absolute difference between current and target pose. The results are listed in table III. We also evaluated the precision of the landing procedure by observing the mean error in $x$, $y$ and yaw after landing. See table IV for the results.

### C. Runtime Analysis

For all setups the processing time was measured (see table V). P4P took 21 ms on average on the ATMega microcontroller, 6 ms on the Gumstix Overo Fire and 1 ms on the Pixhawk's Intel Core 2 duo processor. Ball detection took 9 ms on the Gumstix and 1 ms on the Intel Core 2 duo. Together with P4P it yields an overall detection time of the table tennis ball method of 15 ms on the Gumstix and 2 ms on the Intel Core 2 duo.

|  | X | Y | Z | yaw |
|---|---|---|---|---|
| Hummingbird with IR-LEDs | 7.30 | 6.53 | 3.44 | 8.5 |
| Hummingbird with table tennis balls | 10.78 | 9.23 | 6.89 | 4.66 |
| Pixhawk with IR-LEDs | 3.34 | 7.34 | 10.28 | 4.80 |
| Pixhawk with table tennis balls | 12.89 | 12.64 | 7.14 | 2.77 |
| Pixhawk with landing pad | 8.95 | 7.90 | 2.52 | 3.13 |

TABLE IV

MEAN ERROR AFTER LANDING. X AND Y IN *cm*, YAW IN DEGREES

|  | X | Y | yaw |
|---|---|---|---|
| Hummingbird with IR-LEDs | 6.64 | 6.54 | 7.1 |
| Hummingbird with table tennis balls | 8.38 | 15.32 | 7.03 |
| Pixhawk with IR-LEDs | 3.70 | 10.50 | 7.43 |
| Pixhawk with table tennis balls | 11.74 | 7.01 | 5.95 |
| Pixhawk with landing pad | 3.79 | 7.86 | 5.26 |

Detection of the H-Pattern on the Pixhawk took 10 ms on average, the subsequent pose estimation 1 ms. Using the respective processing hardware, all methods could be run at 60 Hz, the full frame rate of the camera.

## VI. CONCLUSIONS

In this paper, we evaluated three methods for estimating an MAV's pose by using visual markers and a monocular camera. All methods are applicable for autonomous take-off, hovering and landing of an MAV. The error in pose estimation was significantly less than the error introduced by the controller during autonomous flight. Also the operating range for each method is large enough to tolerate the movements of the quadrocopter. Though the test setup was with a downward looking camera for the "H" pattern and a sideways

TABLE V

RUNTIME ANALYSIS OF ALL METHODS

|  | Microcontroller | Gumstix | Intel |
|---|---|---|---|
| IR-LEDs | 21 ms | 6 ms | 1 ms |
| table tennis balls | - | 15 ms | 2 ms |
| landing pad | - | - | 10 ms |

looking camera for the other two approaches, one should also be able to place each pattern and mount the camera according to the current application, no matter which method one chooses.

Regarding the processing time, the IR-LED based system is the fastest, and also the only one being able to run on a microcontroller, exploiting the abilities of the Wii Remote camera. Yet it is the least accurate system in our comparison. One further drawback is its dependence on active markers, therefore requiring additional power and being more sensitive to sunlight.

The table tennis ball approach is faster than the "H"-pattern approach, but depends on color information. It is comparable in accuracy to the last method. For the hovering task it is interesting to notice that it was the worst of all three in position error, but in both cases better in yaw error compared to the other method on the same platform, respectively. This is due to the fact that for this method yaw was the most critical quantity to control in order to keep the pattern in sight of the camera. Therefore the PID parameters were adjusted based on a high priority on yaw. Using passive markers, the method can be used under sunlight conditions and is therefore also suited for outdoor scenarios, where the first method will fail.

The last method is the most computing intensive and thus slowest method, impractical in porting it onto our smaller Hummingbird platform. However, it is notably the most accurate along the normal of the pattern, in our case the *z*-axis, resulting in a stable altitude controller, proven during autonomous hovering. It also got the quadrocopter closest to the landing pad, with the least yaw error. The use of passive markers also enables it for outdoor use. Compared two the second method, no color camera is needed.

From the manual and autonomous flight experiments, it can be said that all methods have a comparable accuracy. Therefore one should choose the method depending on processing power and lighting conditions.

## VII. ACKNOWLEDGMENTS

Andreas Schilling from the computer graphics group in our faculty for providing us the tracking system.

## REFERENCES

[1] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, S. Longhi, "A Vision-Based Guidance System for UAV Navigation and Safe Landing using Natural Landmarks", *Journal of Intelligent & Robotic Systems*, Vol. 57, No. 1-4, 2010, pp. 233-257.

[2] Q. Chen, H. Wu, T. Wada, "Camera Calibration with Two Arbitrary Coplanar Circles", *ECCV-2004, LECS*, Vol. 3023/2004, 2004, pp. 521-532.

[3] 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*, 24(6):381–395, 1981

[4] A. Fitzgibbon, etc., "Direct least square fitting of ellipses", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 21(5), 1999, pp. .

[5] P.J. GarciaPardo, G.S. Sukhatme, J.F. Montgomery, "Towards vision-based safe landing for an autonomous helicopter", *Robotics and Autonomous Systems*, Vol. 38(1), 2002, pp. 19-29.

[6] S. Grzonka, G. Grisetti, W. Burgard, "A Fully Autonomous Indoor Quadrotor", *IEEE Transactions on Robotics*, Vol. 28(1), Feb. 2012, pp. 90-100 .

[7] B. Herisse, T. Hamel, R. Mahony, F.-X. Russotto, "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow", *IEEE Transactions on Robotics*, Vol. 28(1), Feb. 2012, pp. 77-89.

[8] K.Kanatani and L.Wu, "3D Interpretation of Conics and Orthogonality", *Image Understanding*, Vol. 58, 1993, pp. 286-301.

[9] L. Kneip, D. Scaramuzza, R. Siegwart: A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation. In: Proc. of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[10] V. Kumar and N. Michael, "Opportunities and Challenges with Autonomous Micro Aerial Vehicles", *The International Journal of Robotics Research*, DOI: 10.1177/0278364912455954, online 21 August 2012.

[11] S. Lange, N. Suenderhauf, P. Protzel, "A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments", *Proceedings 2009 International Conference on Advanced Robotics*, Munich, June 2009, pp. 1–6.

[12] A. Masselli, A. Zell, "A Novel Marker Based Tracking Method for Position and Attitude Control of MAVs", *Proceedings of International Micro Air Vehicle Conference and Flight Competition (IMAV2012)*, Braunschweig, Germany, July 2012.

[13] L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision", *Proceedings 2011 IEEE International Conference on Robotics and Automation*, Shanghai, May 2011, pp. 2992-2997.

[14] D. Mellinger, N. Michael, V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors", *The International Journal of Robotics Research*, Jan. 2012 online first, doi:10.1177/0278364911434236.

[15] T. Merz, S. Duranti, G. Conte, "Autonomous landing of an unmanned helicopter based on vision and inertial sensing", *Experimental Robotics IX*, Vol. 21, 2006, pp. 343-352.

[16] S. Saripalli, J.F. Montgomery, G.S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle ", *IEEE Transactions on Robotics and Automation*, Vol. 19(3), 2003, pp. 371-380.

[17] S. A. Scherer, D. Dube, P. Komma, A. Masselli, A. Zell, "Robust Real-Time Number Sign Detection on a Mobile Outdoor Robot", *In Proceedings of the 6th European Conference on Mobile Robots (ECMR 2011)*, Orebro, Sweden, September 2011.

[18] K. E. Wenzel, P. Rosset, A. Zell., "Low-Cost Visual Tracking of a Landing Place and Hovering Flight Control with a Microcontroller", *Journal of Intelligent and Robotic Systems*, 2009, Vol. 57(1-4), pp. 297-311.

[19] K. E. Wenzel, A. Masselli, A. Zell, "Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle", *Journal of Intelligent & Robotic Systems*, Vol. 61, 2011, pp. 221-238.

[20] G. Xu, Y. Zhang, S. Ji, Y. Cheng, Y. Tian, "Research on computer vision-based for UAV autonomous landing on a ship", *Pattern Recognition Letters*, Vol. 30(6), 2009, pp. 600-605.

[21] S. Yang, S. A. Scherer, A. Zell, "An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle", *Journal of Intelligent & Robotic Systems*, 2012 Online First, DOI: 10.1007/s10846-012-97497.