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

Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle

Karl E. Wenzel, Andreas Masselli and Andreas Zell

the date of receipt and acceptance should be inserted later

Abstract We present a system consisting of a miniature unmanned aerial vehicle (UAV) and a small carrier vehicle, in which the UAV is capable of autonomously starting from the moving ground vehicle, tracking it at a constant distance and landing on a platform on the carrier in motion.

Our visual tracking approach differs from other methods by using low-cost, lightweight commodity consumer hardware. As main sensor we use a Wii remote infrared (IR) camera, which allows robust tracking of a pattern of IR lights in conditions without direct sunlight. The system does not need to communicate with the ground vehicle and works with an onboard 8-bit microcontroller. Nevertheless the position and orientation relative to the IR pattern is estimated at a frequency of approximately 50 Hz. This enables the UAV to fly fully autonomously, performing flight control, self-stabilisation and visual tracking of the ground vehicle.

We present experiments in which our UAV performs autonomous flights with a moving ground carrier describing a circular path and where the carrier is rotating. The system provides small errors and allows for safe, autonomous indoor flights.

1 Introduction

Fully autonomous flights require high precision in aircraft positioning, especially in takeoff and landing phases. When landing needs to be accomplished safely on a moving target, the problem becomes highly complex. Our system allows a miniature Unmanned Aerial Vehicle (UAV) to takeoff and depart from a carrier vehicle, track it by holding a constant position, approach and land on the vehicle. Our carrier is represented by a Pioneer 2-DX robot with a platform of $43\,cm \times 82\,cm$.

Until recently, onboard navigation and target tracking with UAVs have mostly been done by using industrial helicopters of significant size and weight [12, 19]. Thanks to great progress in high capacity batteries and energy efficient brushless motors, UAVs

Karl E. Wenzel and Andreas Masselli are PhD students with the Chair of Computer Architecture, headed by Professor Andreas Zell, in the Department of Computer Science, University of Tübingen, Sand 1, 72076 Tübingen, Germany {`karl.e.wenzel, andreas.masselli, andreas.zell`} `@uni-tuebingen.de`

have become smaller and smaller. Equipped with specialized sensors, these miniature flying robots are of high interest for civil and military use, especially for surveillance tasks. Their small size and very light weight also makes them ideal aircrafts for indoor experiments. As most accurate sensors require great processing power, miniature UAVs are often controlled by the help of external sensors and ground stations.

In this paper, we present a low-cost onboard tracking solution. We demonstrate how an inexpensive infrared (IR) consumer electronic camera can be used as the main sensor for stable flight control. The camera is capable of detecting up to four infrared blobs and provides the pixel position of each blob. By leaving the image processing out to the integrated circuit of the camera, the control algorithm can run on an onboard microcontroller at a high frequency without external sensors or a base station. The whole weight of the prototype vision system is below 10 g. This is an important aspect for miniature UAVs, which often are capable of carrying only a few dozen gramms of payload.

The key idea of our approach is to track a T-shaped 3D-pattern of infrared lights, attached to a moving target. The distinct pattern allows for the estimation of the current position, relative to the target. As the camera requires free line of sight to the IR lights, the operation radius is limited to a region from 15 cm to 250 cm in front of the pattern.

As further enhancement to our previous work of autonomous hovering above and landing on a stationary target with the Wii remote sensor [21**?**], the position of the aircraft is now extracted only from the shape of the pattern. The use of an inertial measurement unit or other sensors for target tracking is no longer required. This makes the system applicable for a huge variety of miniature aircrafts.

As Bouabdallah et al. demonstrated in [2], a classical proportional-integral-derivative (PID) controller is able to control a quadrotor in the presence of minor perturbations. This relatively simple algorithm can be run on the onboard microcontroller at a high frequency of about 50 Hz. The aircraft is landed in a $20 \times 20\,cm^2$ area, small enough for automatic charging in future projects. Tracking of a fast moving and turning target is possible, as long as the motion change is smooth.

2 Related Work

Most onboard vision-based tracking and navigation have been done on aircrafts of significant size using industrial cameras and high performance processors. Shakernia et al. [16] and [17] introduced a large, single rotor helicopter system (overall length 3.6 m), where the position and velocity to a planar landing pad was estimated by a vision system. A vision-based autonomous landing algorithm was presented by Saripalli et al. [14]. The helicopter of Nordberg et al. [12] is approximately $2 \times 1\,m^2$ and has a maximum payload of 30 kg. They presented vision methods for motion estimation, navigation and tracking.

A low-cost flight control system for a small outdoor helicopter (60-class) where all processing is performed onboard was described by Roberts et al. [13]. Frew et al. [4] showed a vision-based road-following algorithm for an autonomous fixed-wing aircraft with a wingspan of 2.8 m. All of these aircrafts have in common that they are capable of carrying several kilograms of load. This allows for using relatively heavy personal computers and accurate stereo cameras onboard. Miniature UAVs, like the Asctec Hummingbird quadrocopter are very limited in payload capacity. Therefore, other solutions have to be found to attain similar results.

Kemp [10] presented a visual control method by using a sub-miniature onboard camera of only 9 g weight. Guenard et al. [5] presented a visual servo control capable of stationary or quasi-stationary flight. Herisse et al. [9] used optical flow for hovering flight and vertical landing control. In these projects calculations are done by the ground station, which reduces ballast and allows for great processing power, but leads to restrictions in autonomy.

Experiments with an external motion capture system were conducted by Gurdan et al. [6]. They achieved a maximum deviation of 10 cm from the desired position. Also Watanabe et al. [20] were using a field monitored by stationary cameras. They presented an assistant and training system for controlling RC helicopters during flight. These tracking systems provide very accurate position estimates at a high frequency but are limited to a relatively small area of a few cubic meters. This rules out most practical autonomous tasks.

Our approach differs from these described above, as we use commodity hardware as main sensor. The very lightweight infrared camera of a Wii remote tracks four IR light-emitting diodes as external landmarks. The idea of using LEDs as landmarks is not completely new. Mak et al. [11] described how a miniature helicopter can be visually tracked in six degrees of freedom with only three onboard LEDs and a single on-ground camera. Also the Wii remote camera can be found in former projects. Hay et al. [8] presented an optical tracking system using the camera data of two Wii remotes.

Our previous Wii remote related projects [22] and [21] required an inertial measurement unit with accurate roll and pitch estimates to work properly and did not consider a moving start and landing platform. The current system is able to calculate the current position only by interpreting geometric information. No other data has to be collected and the system is applicable for an even larger variety of aircrafts.

3 The UAV and the Carrier System

This section describes the miniature aircraft including our vision controller and the ground carrier, realized by a wheeled mobile robot. A ground station is used only for monitoring, data logging and parameter variation between different experiments. The whole system can be used indoors.

3.1 Quadrocopter

Quadrocopters are helicopters with four independent, fixed motors. By varying the speeds of the four corresponding rotors, the aircraft can move in six degrees of freedom. Bouabdallah et al. [1] give a detailed analysis of the advantages of quadrotors and a description of their dynamic model.

The model we used is a "X3D-BL Hummingbird Autopilot" quadrocopter distributed by Ascending Technologies (Fig. 1). It is 53 cm in diameter at a total weight of approximately 550 g. A 2.1 Ah lithium-polymer accumulator powers the brushless electronic motors and provides a flight time of up to 23 minutes. With a maximum of 200 g additional payload the flight time is reduced to approximately 12 minutes. In fact, the flight maneuvers necessary for autonomous position hold are highly energy-consuming and lead to a flight time of below 10 minutes. The flexibility of the 8” propellers makes the aircraft relatively safe, even for indoor flights.

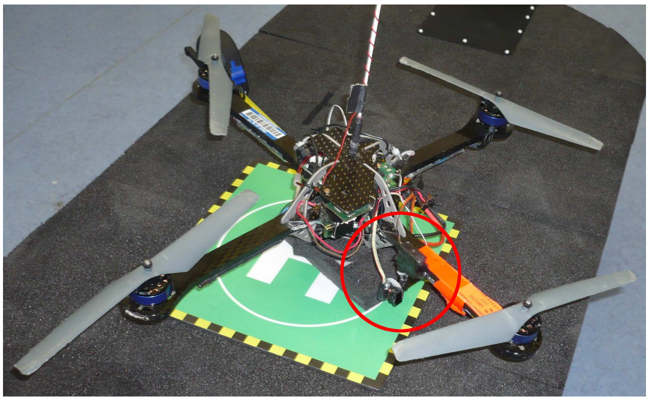


Fig. 1 The Hummingbird quadrocopter on top of the ground carrier. The circle marks the onboard pan/tilt unit, holding the infrared camera.

The Autopilot platform comes with a circuit board, including two 60 MHz 32 bit ARM microcontrollers, a three-axis gyroscope, an accelerometer, a magnetometer (compass), a GPS sensor and a pressure sensor. A ZigBee-Pro serial radio transmission module allows data rates up to 11,500 Kb/s over up to 1 km. The microcontroller and the inertial measurement unit are fast enough to stabilize the quadrocopter sufficiently, making flying relatively easy compared to common model helicopters. The pressure sensor allows for keeping a specific height. Compass and GPS are used to assure a given position in the air or to fly to waypoints, transmitted by the ground station connected via ZigBee. So, the Hummingbird provides basic autonomy without any additional hardware wherever GPS is available, but also permits high-level applications to control the flight and to connect additional devices via two serial ports.

Our controller is realized by an additional microcontroller and the Wii remote camera. The vision-system separated like this, we were able to do experiments with different aircrafts. All processing is done by the Atmel ATmega 644P microcontroller, interfacing the quadrocopter via a serial port. A second serial port is used to send sensor information via ZigBee to the base station. In our case, the station is used for monitoring the current aircraft status and for varying controller parameters or initiating departure and approach. The quadrocopter is capable of flying autonomously without the ground computer.

The additional hardware, including the microcontroller, servo motors and the IR-camera consumes 110 mA (idle) to 250 mA (both servo motors operating) at 5V. In contrast, the brushless motors of the quadrocopter require up to 20A at 11.1V. Since the servo motors are only moved when a correction is necessary, the consumed power is negligible.

3.2 Carrier

As carrier we used a MobileRobot Pioneer 2-DX driving robot with an extended platform of $43\,cm \times 82\,cm$ (Fig. 2). The differential drive robot is equipped with a microcontroller and a ZigBee module, similar to the quadrocopter. Commands from the ground station to drive a circle or rotate at a specific speed are executed accordingly. Accurate stepper motors enable the robot to maintain constant acceleration and driving speed over several runs. The differential drive allows for rotation around the middle axis, much like a naval carrier would turn on water.

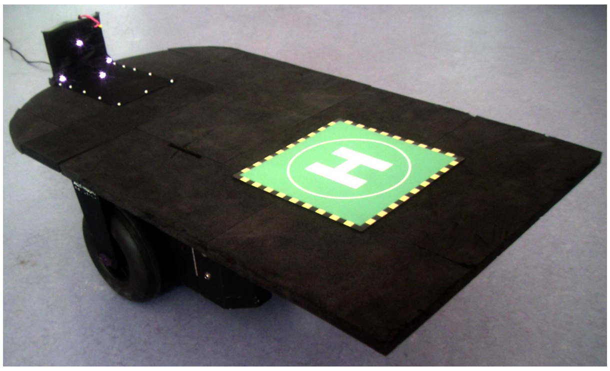


Fig. 2 A Pioneer-II robot is used as driving ground carrier. By tracking the pattern, mounted at the front of the robot, the aircraft can estimate the current relative position. The exact landing point for the aircraft is marked with a typical H-sign.

A pattern of four infrared light-emitting diodes on the front of the robot allows the aircraft to estimate its current position to the carrier. The platform is covered with 15 mm foam, assuring a safe touchdown and prevent of damage in case of any error during takeoff or landing. A pad with a typical H-sign, as used for helicopter landing places, marks the touchdown position in 60 cm distance to the pattern.

4 The Wii Remote Infrared Camera and Pattern

This section describes the main sensor of our vision system, the infrared camera of the Wii remote. The sensor is used to track the pattern of infrared spots, which enables the system to estimate and control the current aircraft position.

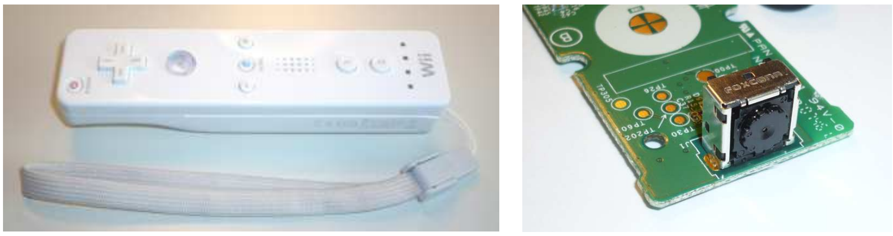
4.1 Wii Remote

The Nintendo Wii controller was designed for interfacing the Nintendo game console. The 40 € device has 12 digital buttons, a three-axis accelerometer, an IR camera tracker and is Bluetooth-compatible. It provides the opportunity to connect additional input devices, like the Nunchuck controller through an expansion port.

To determine the position and orientation of the controller, the optical sensor is normally used in conjunction with a strip containing two infrared spots. This enables to control a cursor on the screen.

The fact that the Wii remote enables access to the PC via Bluetooth is an advantage of which recent publications benefit. Sreedharan et al. [18] analyse the motion by interpreting the acceleration sensor information. Shou et al. [15] integrate this controller into a game environment in a multi-wall virtual reality theatre.

As weight is a crucial factor for miniature flying robot applications, we detached the camera from the controller. When operating onboard, the Bluetooth connection is



(a) The Wii remote.

(b) The internal camera sensor.

Fig. 3 The sensor of the Wii remote.

not required, but direct I^2C bus communication is appropriate. The remaining weight of 0.4 g at a dimension of $8 \times 8 \times 5\,\text{mm}^3$ makes it an ideal miniature UAV onboard sensor.

The camera, a multi-object tracking engine (MOT sensor), manufactured by PixArt Imaging tracks up to four IR blobs. The native resolution of 128×96 pixel is scaled up to 1024×768 pixels by eight times subpixel analysis. Different data modes provide basic information like the pixel position, but also more advanced knowledge as the blob size, intensity and bounding box. The horizontal field of view of approximately 45° is adequate for most applications. In Bluetooth mode, the device allows a refresh rate of up to 100 Hz. When directly communicating via I^2C bus mode, frequencies of 250 Hz and higher can be achieved. The sensor can easily be integrated in a microcontroller circuit, as the camera runs at 3.3 V and only needs an external synchronisation of 24 MHz.

4.2 Camera Pan/Tilt Unit

To assure robust tracking of the carrier, the pattern must be kept in the camera image center. This has been realized by attaching the camera to two servo-motors (Fig. 4). Whenever the pattern intents to move out of the picture, the motors move to center the pattern again.

An additional wire, soldered to the internal potentiometer of the servo allows to accurately measure the current servo angle. This helps to center the pattern in the camera image and is also required to correct the estimated yaw angle of the aircraft.

4.3 3D Pattern

The tracked pattern consists of four infrared (IR) light-emitting diodes (LEDs) attached to a standard circuit board. Three of the LEDs (**L**, **R**, **T**) are in the vertical yz plane, the fourth (**F**) is displaced in x (Fig. 5). This makes the distortion of the pattern unambiguous in the operating range. The black cover eliminates most disturbing reflections.

The size of the pattern depends on the desired operation range. As our system is intended to land on a relatively small moving target, the pattern has to fit in the camera image at a very small distance of several centimeters. On the other hand, a small pattern makes it difficult to estimate the precise position from a larger distance.

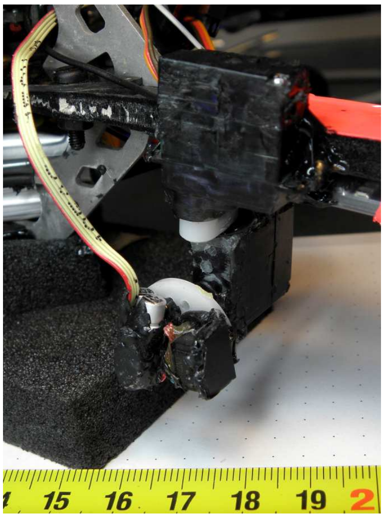
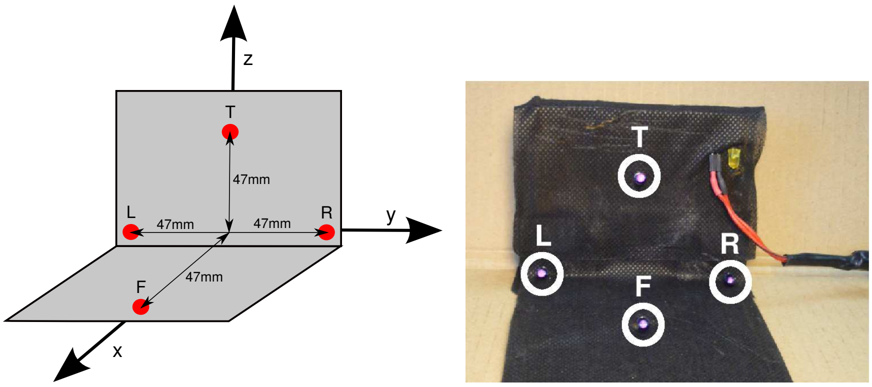


Fig. 4 The small camera with an infrared filter attached to servo motors at the quadrocopter frame. The motors assure, that the pattern stays in the center of the camera image.



(a) Schematic view of the pattern.

(b) The pattern on a standard circuit board.

Fig. 5 The tracked pattern. The four infrared LEDs represent the points left **L**, right **R**, top **T** and front **F**. The symmetric shape allows for unambiguous spot identification in the operating range.

The current configuration allows for tracking from 15 cm to 250 cm. However, when the aircraft is moving near to the pattern, single spots of the pattern usually get lost due to the inclination, even at 40 cm distance.

The pattern measures 94 mm from left to right and 47 mm from bottom to top. One LED is placed symmetrical between the right and left LED and displaced by 47 mm. The odd distances are a result of the breadboard arrangement.

Several LEDs have been tested for light intensity and angle of radiation. The light intensity can directly be read from the camera and was measured in typical working distances. The angle of radiation limits the operating range, but was overall similar for all tested LEDs. The best intensity at an angle of radiation of approximately 45 ° was measured with the Vishay TSAL6200 LED. As the angle of radiation is about 45 °, the displacement level of the front LED was chosen to allow unambiguous identification up to this line of sight.

5 Retrieving the position From the Camera Image

The data coming from the camera contains the pixel position of the four blobs and optional information as intensity and bounding box. The following section shows, how this data is processed to gain the current aircraft position. With fast computers, the necessary steps of image processing are mostly done like described by Hartley and Zisserman [7]. Since our algorithm has to be executed on a microcontroller and the input shape is well-known, we decided to use a simple, optimized algorithm which is fast and assures robust recognition on the given hardware. This means that we had to use as few floating point, logarithm or trigonometric functions as possible.

5.1 Spot Identification

As the pixel data is in random order, it first has to be sorted to identify the four spots as top, left, front and right point.

Since it can be assumed that the quadrocopter is not in an upside-down position or behind the pattern, the spots can simply be sorted by their pixel positions. The leftmost and rightmost spots are identified first, the upper and lower spots of the two remaining ones correspondingly.

5.2 Pose Calculation

Having all points identified, the current position $P(x, y, z)$ and yaw angle of the aircraft can be calculated. The task of calculating the position of a camera from a known pattern of control points and their projection onto the camera's image plane is well known as the perspective-n-point (PnP) problem, described in [3]. Since the operating range of the quadrocopter is restricted in many aspects, we used an algorithm which is optimized to solve this special case of the PnP, resulting in faster processing. We also focus on getting the position and yaw angle from the IR-Pattern, while the pitch and roll angle of the quadrocopter are obtained from the IMU.

Given are four points in the world coordinate system $L(0, -47, 0)$, $T(0, 0, 47)$, $F(47, 0, 0)$, $R(0, 47, 0)$, and four corresponding points in the image plane L_i, T_i, F_i, R_i.

First, only L, F, R and their corresponding image points are used to obtain P. If only three control points are used, a maximum of four locations of P are possible [3]. In our restricted operating range the solutions can be easily narrowed down to the one which represents the actual position of P. The remaining control point T together with T_i can then be used to rate the position estimation.

Consider the plane $z = 0$, in which L, R and F are located (Fig. 6). Then we know from the inscribed angle theorem, that the projection of P onto the plane $z = 0$ lies on a circle k with center

$$M_k = \left(\frac{R_y}{tan(\alpha + \beta)}, 0 \right) . \tag{1}$$

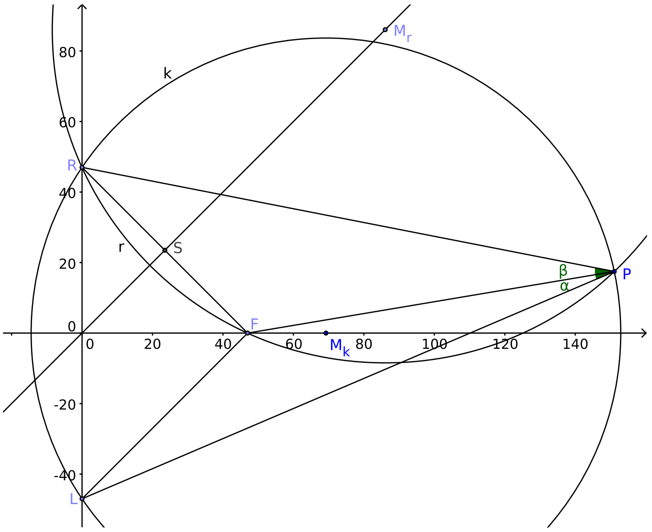


Fig. 6 Top view of the the plane $z = 0$, where 3 points of the pattern are located. The x and y coordinates of the desired point P can be estimated by intersecting the circles k and r.

We also know from the inscribed angle theorem, that the projection of P onto the plane $z = 0$ lies on a circle r with center M_r, which is located on the perpendicular bisector with

$$d(M_r, S) = \frac{d(F, S)}{tan(\beta)}, \tag{2}$$

where d is the euclidean distance of two points and $S\left(\frac{R_x+F_x}{2}, \frac{R_y+F_y}{2}\right)$.

If we intersect both circles, we get two solutions: R and the projection of the desired point P. P_x and P_y can be obtained by the following formula, while P_z will be calculated below.

$$P_y = -\frac{b \cdot M_{ry} + \Delta x \cdot \sqrt{r_k^2 \cdot a - b^2}}{a} \tag{3}$$

$$P_x = M_{kx} + \sqrt{r_k^2 - P_y^2}, \tag{4}$$

where

$$\Delta x = M_{rx} - M_{kx} \tag{5}$$

$$\Delta y = M_{ry} - M_{ky} \tag{6}$$

$$a = \Delta x^2 + \Delta y^2 \tag{7}$$

$$b = M_{rx} \cdot M_{kx} - M_{ry} \cdot R_y - M_{kx}^2 \tag{8}$$

and $r_k = \sqrt{M_{kx}^2 + R_y^2}$ being the radius of circle k.

The angles α and β are calculated from the image points as follows:

Consider the image plane. Let g be the straight line through L_i and R_i, and h the line perpedicular to g through F_i. Let H_i be the foot of h on g. Then

$$\alpha = d(L_i, H_i) \cdot \rho, \beta = d(R_i, H_i) \cdot \rho, \tag{9}$$

where d is the euclidean distance in pixels and $\rho = 7.61 \cdot 10^{-4}$ is the camera view angle per pixel; ρ has been determined by calibrating the WiiMote camera. Again, this only approximates the true angles, but the error is negligible, since the noise of the blob positions of the WiiMote camera produces higher errors.

To provide numerical stability for small angles, the calculation of P is slightly changed if $\beta < \alpha$. Then a circle l through the points L, F and P is used instead of r. If both angles are small, the prediction becomes numerically unstable. However, this is not the case within the operating range.

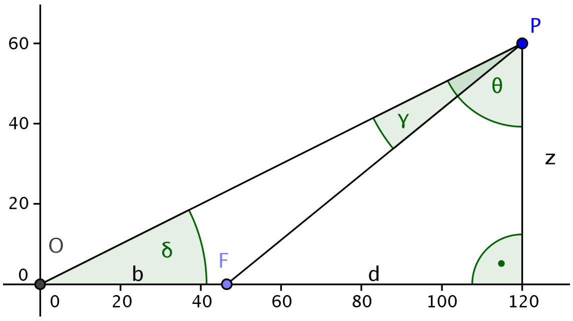


Fig. 7 Side view of the the plane $y = 0$. The estimation of P_z is done with the knowledge of γ. The shown proportions are only for illustration and untypical for flights.

The pixel distance of F and H multiplied by ρ describes γ (Fig. 7) and b is the displacement of the front spot to the origin O (47 mm). With the knowledge of γ and the assumption, that P_z is typically smaller than the distance d, P_z can be estimated:

$$\delta = \frac{\pi}{2} \pm \arccos\left(\frac{d \cdot \sin(\gamma)}{b}\right) \tag{10}$$

$$P_z = \tan(\delta) \cdot (b + d) \tag{11}$$

An alternative is to determine P_z using the pitch angle θ, which can be calculated using IMU, servo and camera information. An advantage of this is the independence of pattern rotations around the Y-Axis. However, experiments have shown bigger errors using this method since the IMU and servo angle estimations are noisy. In our scenario, the pattern does only move horizontally and therefore we prefer the estimation by distortion.

The calculation of the yaw angle ψ (and similar θ) consists of three parts. Assuming the aircraft is directed to the pattern, the yaw angle is $\psi_d = atan2(P_y, P_x)$. This is only true if the pattern is perfectly centered in the camera image and if the servo is in neutral position. The visual displacement ψ_v is estimated by the pixel displacement multiplied by ρ. To get the real yaw angle, ψ_v and the servo angle ψ_s must be added to ψ_d:

$$\psi = \psi_d + \psi_v - \psi_s \tag{12}$$

5.3 Outlier detection

Some influences such as sunlight or reflections prevent the correct recognition of the pattern and make an outlier detection necessary. Several methods have been tested and the following has shown good results. We calculate a value for the confidence of a valid measure.

One indicator is the calculated distance to the pattern. If it is smaller or larger than reasonably thresholds, the distance confidence c_d is zero. Otherwise, the corresponding distance of all pairs of points are looked up. A value depending on the difference of the calculated distance d_i in percent of the minimum (d_s) and maximum (d_l) distance is applied.

Additional indicators, summarized to the roll confidence c_r are the angles of all lines defined by point pairs. The mean of these angles defines the current roll angle (r) of the aircraft. Similar to the distance, the angle also has reasonable limits. The difference of the calculated angle in degrees of the minimum (r_s) and maximum (r_l) angle is critical:

$$\varDelta_r = r_l - r_s \tag{13}$$

$$c_r = \begin{cases} 0 & if \varDelta_r > 80 \vee r < -80\,^\circ \vee r > 80\,^\circ \\ 50 - \varDelta_r & otherwise \end{cases} \tag{14}$$

A perfect shape results in the maximum confidence c value of 100.

$$c = c_d + c_r \tag{15}$$

Typical values are between 50 and 100, even at a distance of more than 1 m. Invalid images are identified reliably and result in a confidence below 50. Thus a fast and robust outlier detection algorithm was found for the use on a microcontroller.

6 Target Tracking and Flight Control Algorithm

Since the angle of sight is limited and the pattern size only allows for robust tracking up to a distance of 2.5 m, the operating range is fixed to a segment of about 45 ° angle of aperture (See Fig. 8).

An accurate position estimation can be assumed if no outlier was detected. By knowing the actual position relative to the pattern, the robot can control its movement. A finite state machine controls the high-level behavior of the aircraft. The framework detects failures like still stand or if the pattern was lost and starts to sink or turns off the motors accordingly. The desired behavior consist of the five phases starting, departure, tracking, approaching and landing (Fig. 9).

Starting from a moving target requires a free line of sight to the pattern from the starting position. So positioning and heading have to be correct. The battery capacity is unknown but strongly influences the thrust required to hover. This consequently biases the behavior of the aircraft in the starting phase.

A switch on the remote control toggles between autonomous flying or manual control. When the switch is activated, the quadrocopter gets a boost and rises to a height of 25 cm. With the height of the robot carrier of 23 cm, this means a total height from ground of 48 cm. This is still a relatively safe position for a flying robot in indoor environment.

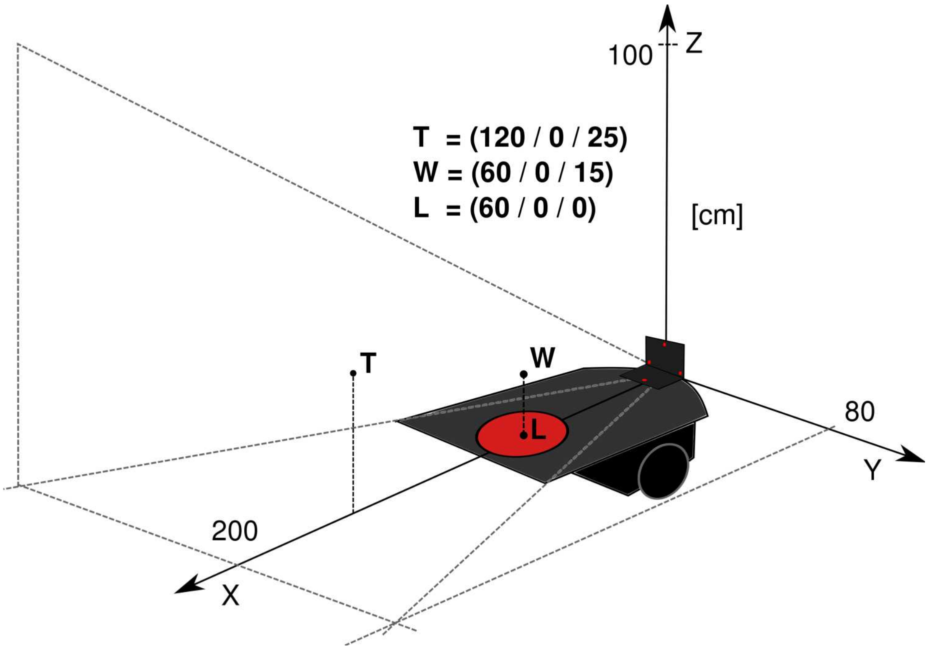


Fig. 8 The operating range of our tracking system. **T** is the tracking position, **W** is the waypoint for approach and **L** is the landing spot. All coordinates are given in *cm*.

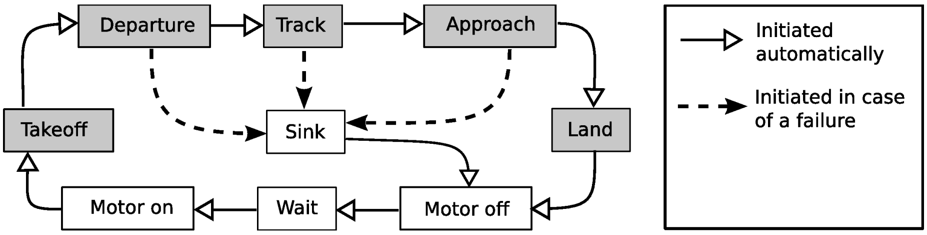


Fig. 9 The possible states of our system. Takeoff and approach are triggered manually by the base station, the other states are entered automatically.

Simultaneously the robot moves to the tracking position **T**. A distance of 120 cm to the target is a safe tracking position, since it allows for good recognition of the pattern and motion of the carrier without losing sight of the pattern. Tracking the moving target requires the aircraft to stay stationary at this point. Straight movement of the target has to result in straight motion of the flying robot. If the target turns, the robot has to move sideways to keep a proper viewing angle. A combination of both is required if the carrier drives a circular path.

The tracking position would also enable an alternative flight system to safely take over the control of the aircraft. GPS or camera based tracking would hardly provide a similar accuracy, but could lead the aircraft back to a position, where our system can again track the target and approach.

Approaching is initiated automatically 10 seconds after the tracking position was reached and sets the waypoint **W** 15 cm above the landing place. As the flying robot still has to compensate for the carrier motion and the landing area is designed to tolerate only a small displacement, the last phase of landing is highly critical. After reaching **W**, the touchdown is initiated by setting the desired z coordinate to zero. When the height falls short of a threshold, the thrust is reduced to zero in a stepwise fashion. Afterwards, the motors are turned off.

Departure and approach require to adjust the desired (control) position in a stepwise fashion, until the new endposition is reached. This equals a simple speed-controller.

When the quadrocopter leaves or enters the area above the carrier platform, the influence of the ground effect varies significantly. To reduce this effect, the departure and approach is performed at a height at which the ground effect is noticeably smaller.

The precise autonomous flight is achieved by four independent proportional-integral-derivative (PID) controllers. The output of the controllers is sent to the quadrocopter via serial interface and handled like stick commands from the radio control. Trim values allow for slight adjustment of the neutral command values for roll (t_r), pitch (t_p), and yaw (t_ψ). The parameters for the proportional (K_p), integral (K_i) and derivative (K_d) parts, as well as possible other parameters, were optimized during various experiments. The integral parameter of all controllers is the smallest and the differential parameter is the biggest value.

The output of the yaw controller $y(t)$ is used for steering and ensures that the pattern keeps centered in the camera frame. The controller is implemented as a standard PID controller, where the error ($e(t)$) is the aberration of the pattern from the image center in degrees:

$$\mathrm{y(t)} = t_\psi + K_p e_\psi(t) + K_i \int_0^t e_\psi(\tau)\, d\tau + K_d \frac{d}{dt} e_\psi(t) \tag{16}$$

The input for the other controllers are the position error of the corresponding axis in mm. Roll ($r(t)$) and pitch ($p(t)$) are managed by identical controllers. To achieve the required predictive behavior, a second derivative was added, taking the acceleration of the aircraft in account:

$$\mathrm{r(t)} = t_r + K_p e_r(t) + K_i \int_0^t e_r(\tau)\, d\tau + K_d \frac{d}{dt} e_r(t) + K_{d'} \frac{d^2}{dt^2} e_r(t) \tag{17}$$

$$\mathrm{p(t)} = t_p + K_p e_p(t) + K_i \int_0^t e_p(\tau)\, d\tau + K_d \frac{d}{dt} e_p(t) + K_{d'} \frac{d^2}{dt^2} e_p(t) \tag{18}$$

There is no constant value for the thrust required to hover the aircraft at any height. The thrust value for holding a height is called t_h and initiated to work for a fully charged battery. If the battery runs lower, and the aircraft keeps below the desired height for a time, an accumulator increases t_h. If the aircraft flies too high, t_h will be decreased. The fast response controller for the height ($h(t)$) could be realized by an unmodified PID loop:

$$\mathrm{h(t)} = t_h + K_p e_h(t) + K_i \int_0^t e_h(\tau)\, d\tau + K_d \frac{d}{dt} e_h(t) \tag{19}$$

The combination of these controllers has proven to enable robust position control.

Table 1 Controller characteristics of ten clockwise driven circular path experiments for departure, tracking, approach and landing. x, y, z are given in cm, ψ in °.

	Mean				Standard deviation				Max. absolute error			
	x	y	z	ψ	x	y	z	ψ	x	y	z	ψ
Dep.	5.8	-3.0	-2.5	11.1	9.1	6.9	5.0	7.1	30.8	18.6	15.3	30.3
Track.	3.8	-0.3	-0.1	4.4	7.6	6.2	2.9	8.0	31.7	18.0	9.9	29.7
App.	0.0	0.1	0.1	-5.2	5.9	6.4	2.8	12.4	16.9	19.6	9.2	31.2
Land.	-2.2	-3.1	-	-7.3	6.9	5.4	-	15.1	12.2	9.1	-	22.6

Table 2 Controller characteristics of ten counterclockwise driven circular path experiments for departure, tracking, approach and landing. x, y, z are given in cm, ψ in °.

	Mean				Standard deviation				Max. absolute error			
	x	y	z	ψ	x	y	z	ψ	x	y	z	ψ
Dep.	2.8	-3.8	-0.7	-0.7	8.7	8.2	3.8	12.2	27.2	28.3	15.0	30.5
Track.	3.6	-1.6	-0.0	0.6	8.9	6.4	3.0	14.8	32.2	20.8	10.5	31.4
App.	0.0	2.1	0.3	0.6	6.2	6.7	2.9	14.5	26.9	22.2	18.0	29.1
Land.	0.6	1.7	-	-5.2	3.5	3.4	-	16.8	7.7	9.7	-	21.9

7 Experimental Results

Following a large number of flights and retrieval of working parameters, the system has proven good tracking ability. Note that the Hummingbird quadrocopter provides a GPS position controller and an air-pressure sensor for height control, but our experiments were done without the help of these sensors.

Takeoff with a used battery is critical, since the thrust value required to hover (t_h) may be incorrect. In such a case, the aircrafts lift-off is relatively small, and the pattern could be lost before the robot reaches a safe tracking position. t_h can be set remotely by the base station. Landing is also critical, since the aircraft has to leave the tracking position and fly too close to the pattern, to allow robust motion change reactions. Nevertheless, experiments have shown satisfactory results in all phases.

Two different scenarios were tested. In the first scenario, the carrier drove a circular path with a constant velocity. In the second scenario, the carrier emulated a ship turning on water.

Since the camera position estimation is of good quality (approximately 5% error), the position vector $\mathbf{p}$ was used as ground truth in the experiments.

7.1 Autonomous Flight with a Carrier Following a Circular Path

The Pioneer 2 robot is able to follow a circular path at a given speed. In our experiments, the carrier was driving a circle with a diameter of 2.5 m at 40 $\frac{cm}{s}$. A sequence of twenty autonomous flights, ten clockwise, ten counterclockwise, with an overall success rate of 90%, represents the characteristics of the system. Two flights failed, one in departure, one in the approaching phase.

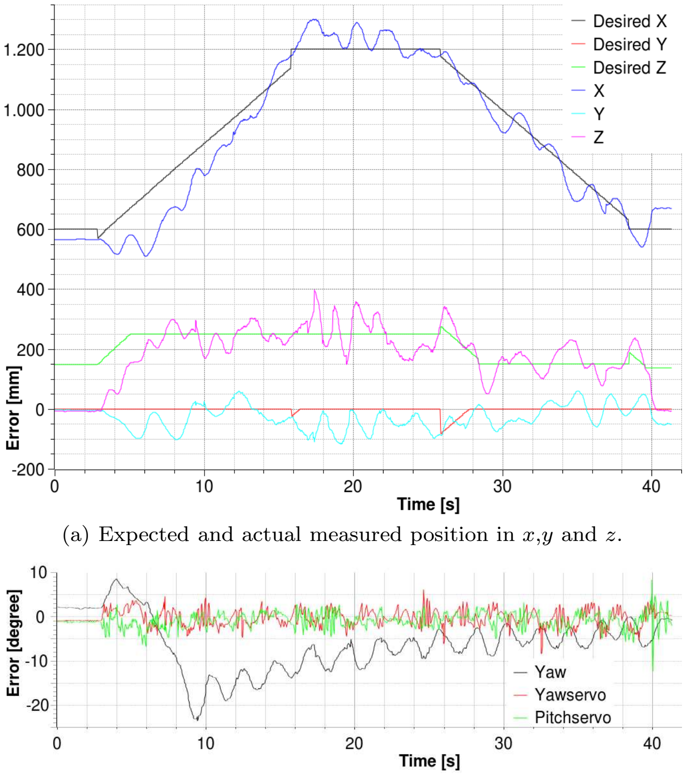
The mean, root mean square deviation and the maximum absolute error of the successful eighteen flights are shown in Table 2. The landing row shows the position error of touchdown.

During departure, the aircraft is relatively unstable, which results in relatively high x and y errors. Since the height and t_h have to be adjusted during departure, also the z error has its maximum during that phase. While tracking and approaching, z errors are noticeably smaller. This is obvious, since the carrier describes a smooth motion and induces only errors in x and y.

7.2 Autonomous Flight with a Rotating Carrier

In the second scenario, the Pioneer-II robot rotates its wheels in different directions. This leads to a turn on the spot. Since the quadrocopter has to keep a stationary position directly behind the carrier, lateral compensation is necessary. Experiments where the carrier rotates with a angular speed of up to $50\,^\circ/s$ have shown satisfactory tracking results. The rotation axis is 30 cm behind the pattern. This means the aircraft has to fly a circular path of 5.65 m circumference or 1.8 m diameter at a speed of $78\,\frac{cm}{s}$ and to persistently correct the yaw angle. The parameters of the quadrocopter were slightly adjusted.

For starting and landing, the robot was rotating at a slower constant speed of $10\,^\circ/s$, still resulting in a lateral aircraft motion of $16\,\frac{cm}{s}$. Of a sequence of twenty autonomous flights (ten clockwise, ten counterclockwise), two failed in the approaching phase. The success rate of 90% proves rather robust controlling ability.



(a) Expected and actual measured position in x,y and z.

(b) Yaw and servo angle errors.

Fig. 10 Example plot of the errors of one autonomous flight while the carrier was rotating.

Table 3 Controller characteristics of ten clockwise rotation experiments for departure, tracking, approach and landing. x, y, z are given in cm, ψ in °.

	Mean				Standard deviation				Max. absolute error			
	x	y	z	ψ	x	y	z	ψ	x	y	z	ψ
Dep.	3.9	-2.8	-2.9	7.3	9.0	6.6	6.3	10.3	27.8	22.7	25.8	28.4
Track.	1.9	-1.8	-0.7	1.2	6.0	7.3	4.1	11.0	18.0	22.2	16.9	28.5
App.	0.7	0.0	0.5	-1.9	8.0	6.9	3.7	12.9	21.7	25.0	13.8	30.8
Land.	-2.2	-5.7	-	-3.3	3.2	3.4	-	16.1	6.7	7.8	-	26.6

Table 4 Controller characteristics of ten counterclockwise rotation experiments for departure, tracking, approach and landing. x, y, z are given in cm, ψ in °.

	Mean				Standard deviation				Max. absolute error			
	x	y	z	ψ	x	y	z	ψ	x	y	z	ψ
Dep.	-9.6	-4.4	-3.7	-10.8	6.9	4.1	6.7	5.3	29.0	16.8	21.8	23.6
Track.	2.3	-0.2	-0.1	-7.0	6.4	5.9	4.7	2.5	19.4	25.6	30.3	13.3
App.	-0.4	-1.0	-0.0	-3.7	7.4	6.8	4.5	3.2	22.3	31.1	16.2	30.0
Land.	0.0	2.7	-	-0.1	5.1	7.2	-	3.3	7.6	16.6	-	6.6

Figure 10 represents the performance of the controllers over one autonomous flight. Figure 10(a) shows the desired (control) position and the actual measured position in x,y and z. Some oscillations still remain and more accurate PID-Parameters for hovering can be found, but do not work with postition changes caused by a moving pattern.

Figure 10(b) shows the angle errors of the servo motors and yaw. The yaw response of the quadrocopter correlates with other movements. Because the servo motors ensure proper sight to the pattern, the yaw control can be weihted lower and thus limit the influence of the position control to a minimum.

Table 3 and Table 4 show the characteristics of the eighteen successful flights. The landing row shows the position error of touchdown.

A standard deviation of below 10 cm and a maximum offset of 32 cm allows for autonomous flights in narrow indoor environments and proves stable position hold capacity of the controllers even with a moving target.

8 Conclusion and Future Work

Most vision-based aircraft tracking requires relatively expensive camera hardware and great processing power. Our approach uses a lightweight, low-cost commodity hardware camera, a microcontroller and a pattern of infrared lights, with a total cost of about 50 €. Nevertheless our system has proven satisfactory tracking results. The accuracy allows for automatic takeoff, departure, tracking, approaching and landing on a moving target.

The success rate of the whole sequence is currently about 90 %, failures are mostly due to overshooting the tracking cone or to ground effects at departure or landing at the relatively small moving platform.

The accuracy achieved is comparable with those of stationary systems and allow for operation in indoor environment. Starting is still critical, since the battery level and therefore the required thrust is unknown. Tracking a moving target at a safe position is rather robust as long as the target motion is smooth. Landing is precise enough to attach landing pads for autonomous charging in future projects.

One obvious disadvantage of using the Wii remote camera is the need of artificial landmarks. The tracking of infrared LEDs can be prevented by sunlight, which makes the system not applicable for outdoor use at daylight. The pattern also limits the possible operating range to a segment in front of the target. An array of LEDs and a bigger pattern would enlarge the detection range of the target, but could not eliminate possible errors caused by sunlight.

Nevertheless the system has proven satisfactory results with a minimum of hardware and processing costs. It provides a relatively simple device for indoor tracking, even for helicopters with very limited payload capacity.

The setup can also be used for safe automatic approaching and landing after a successful autonomous GPS-based or manual flight. Future work will focus on tracking faster moving ground targets, or another aircraft, equipped with infrared lights. Precise tracking will allow for follow-the-leader scenarios or high-level formation controllers.

References

1. Bouabdallah, S., Murrieri, P., Siegwart, R.: Design and Control of an Indoor Micro Quadrotor. In: IEEE International Conference on Robotics and Automation (ICRA) (2004)
2. Bouabdallah, S., Noth, A., Siegwart, R.: PID vs LQ control techniques applied to an indoor micro quadrotor. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 2451–2456 (2004)
3. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981). DOI http://doi.acm.org/10.1145/358669.358692
4. Frew, E., Mcgee, T., Kim, Z., Xiao, X., Jackson, S., Morimoto, M., Rathinam, S., Padial, J., Sengupta, R.: Vision-based road-following using a small autonomous aircraft. In: IEEE Aerospace Conference, vol. 5, pp. 3006–3015 (2004)
5. Guenard, N., Hamel, T.: A practical visual servo control for an unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1342–1348 (2007)
6. Gurdan, D., Stumpf, J., Achtelik, M., Doth, K.M., Hirzinger, G., Rus, D.: Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 361–366. Roma, Italy (2007)
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, second edn. Cambridge University Press (2004)
8. Hay, S., Newman, J., Harle, R.: Optical tracking using commodity hardware. In: 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp. 159–160 (2008)
9. Herisse, B., Russotto, F.X., Hamel, T., Mahony, R.E.: Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 801–806 (2008)
10. Kemp, C.: Visual control of a miniature quad-rotor helicopter. Ph.D. thesis, Churchill College, University of Cambridge (2006)

11. Mak, L.C., Furukawa, T.: A 6 DoF visual tracking system for a miniature helicopter. In: 2nd International Conference on Sensing Technology (ICST), pp. 32–37 (2007)
12. Nordberg, K., Doherty, P., Farnebäck, G., Forssén, P.E., Granlund, G., Moe, A., Wiklund, J.: Vision for a UAV helicopter. In: International Conference on Intelligent Robots and Systems (IROS), workshop on aerial robotics. Lausanne, Switzerland (2002)
13. Roberts, J., Corke, P., Buskey, G.: Low-cost flight control system for a small autonomous helicopter. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 546–551. Taipai, Taiwan (2003)
14. Saripalli, S., Montgomery, J.F., Sukhatme, G.S.: Vision-based autonomous landing of an unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2799–2804. Washington, DC, USA (2002)
15. Schou, T., Gardner, H.J.: A Wii remote, a game engine, five sensor bars and a virtual reality theatre. In: OZCHI '07: Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design, activities, artifacts and environments, pp. 231–234. ACM, New York, NY, USA (2007)
16. Shakernia, O., Ma, Y., John, T., Sastry, K.S.: Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control. Asian Journal of Control **1**, 128–145 (1999)
17. Sharp, C.S., Shakernia, O., Sastry, S.S.: A vision system for landing an unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea, pp. 1720–1727 (2001)
18. Sreedharan, S., Zurita, E.S., Plimmer, B.: 3d input for 3d worlds. In: OZCHI '07: Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design, activities, artifacts and environments, pp. 227–230. ACM, New York, NY, USA (2007)
19. Valavanis, K.P. (ed.): Advances in Unmanned Aerial Vehicles. State of the Art and the Road to Autonomy. Springer (2007)
20. Watanabe, K., Iwatani, Y., Nonaka, K., Hashimoto, K.: A visual-servo-based assistant system for unmanned helicopter control. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 822–827 (2008)
21. Wenzel, K.E., Zell, A.: Automatic Take Off, Hovering and Landing Control for Miniature Helicopters with Low-Cost Onboard Hardware. In: Proceedings of the AMS'09, Autonome Mobile Systeme 2009, pp. 73–80. Karlsruhe, Germany (2009)
22. Wenzel, K.E., Zell, A.: Low-Cost Visual Tracking of a Landing Place and Hovering Flight Control with a Microcontroller. Journal of Intelligent and Robotic Systems **57**, 297–311 (2009)