2015 Third International Conference on Image Information Processing

Autonomous Visual Tracking and Landing of a Quadrotor on a Moving Platform

Juhi Ajmera[1], Siddharthan PR[2], Ramaravind K.M.[3], Gautham Vasan[4], Naresh Balaji[5], V. Sankaranarayanan[6]
[1, 2, 3, 4, 5 ,6]Department of Electrical and Electronics Engineering
National Institute of Technology
Tiruchirappalli, Tamilnadu, India
[1]juhi.ajmera@gmail.com, [2]sperundurairajas@wpi.edu, [3]raam.arvind93@gmail.com,
[4]vasan@ualberta.ca, [5]balajiravi25@gmail.com, [6]vsankar@nitt.edu

***Abstract*—This paper deals with the problem of autonomous visual tracking and landing of a quadrotor on a moving platform. The application of this problem can be seen in autonomous landing of UAV on a ship. The solution consists of two parts, a sensor framework to estimate the state of quadrotor using vision based approach and a controller design which generates appropriate actuator commands. A computer vision approach is proposed which tracks the pre-specified oriented roundel object continuously while maintaining a fixed distance from the roundel and also simultaneously keeping it approximately in the center of the image plane. Quadrotor's relative pose is estimated through dead reckoning and control approach is implemented which seeks for full autonomy of the robot, by considering only internal sensors and processing unit and also, experimental results are presented.**

Keywords - Computer Vision, Quadrotor, Autonomous Tracking

I. INTRODUCTION

Unmanned Aerial Vehicle(UAV) has grown great interest in both civil [1] and military scopes focusing on searching, rescue, inspection and so on. In March 2011, it was used for exploration of damaged nuclear reactors in Fukushima and is considered as the most noticeable contribution. Afterwards, there is rapid increase in the scope for its commercial and research opportunities, specially a type, Quadrotors, which became standard platform worldwide to perform such research activities [2].

Although many sensors like GPS and bluetooth are available, precise position control is always a challenging task for navigation of quadrotors for which computer vision serves as an excellent solution which supports the vehicle autonomy using visual informations [3], [4]. Visual automation indicates landing site as well as tracking of an object or human. In [3], a real time vision based landing algorithm for an autonomous unmanned helicopter is shown, which is implemented for the landing of a helicopter on moving target in [4]. In [5], a novel nonlinear autopilot for quadrotor UAVs is proposed with a tracking and landing controller. In [6], a vision based target following and landing system for quadrotor on a moving platform is shown. In [7], full landing sequence of quadrotor in indoors are introduced using image-based visual servoing.

Object tracking problem is considered as most basic and fundamental behaviour for robots, which drags attention to important and interesting facts in the visual world. Numerous visual object tracking algorithms came into light. Most of them primarily include three steps: (a) extracting features of the object of interest in the initial frame of a continuous video stream, (b) detecting object in the successive frames by finding object of similar features, (c) tracking detected object from one frame to another. Mean-shift algorithm, which comprises all of the above mentioned steps, is used as object tracker for the rest of the discussion due to its simplicity and efficiency for implementation in real-time tracking [8]. Many works have been done for making UAV autonomous [9], [10], [11]. In [9], basic control algorithm and making of quadrotor is shown, whereas, [10] shows full autonomous controller of a quadrotor. In [11], it is shown that shared control of quadrotor is more useful in practical cases.

In this paper, we propose a controller to track a moving mobile robot and landing on the helipad autonomously. A computer vision approach is defined which tracks the pre-specified oriented roundel object continuously while maintaining a fixed distance from the roundel and also simultaneously keeping it approximately in the center of the image plane. The main sensor used here is the onboard bottom facing camera to detect the target and calculate the relative position and send it for off-board image processing to ground control station. The procedure of image processing and controlling of quadrotor should be robust to disturbance with moving target and landing on it.

The organization of the paper is as follows. Section II introduces the mathematical model and tracking control of quadrotor. In section III, computer vision based trajectory tracking algorithms are explained. In section IV, control problem is defined followed by control architecture in the section V. Experimental results are given in section VI and concluding remarks are given in the VII section.

II. MATHEMATICAL MODEL AND TRACKING CONTROL

A. Mathematical Modeling

The quadrotor system consists of four identical rotor and propellers, located at the vertices of a square. Control of quadrotor is done by the thrust generated by individual rotor

[12]. We choose inertial frame $\{\vec{X}, \vec{Y}, \vec{Z}\}$ and one body fixed frame $\{\vec{x}, \vec{y}, \vec{z}\}$. The origin of the body fixed frame is located at the center of the mass of the quadrotor. Rotation of quadrotor about x axis is defined by the roll angle, ϕ, followed by a rotation about y axis through the pitch angle θ, followed by a third yawing rotation about z axis through yaw angle ψ. The system is underactuated with six degrees of freedom corresponding to only four controlling inputs. Define,

$\zeta \in \Re^3$	the position vector of the center of mass in the inertial frame
$m \in \Re$	the total mass of quadrotor
$l \in \Re$	the distance from the center of mass to the center of each rotor in the body fixed plane
$f_i \in \Re$	the thrust generated by the i^{th} propeller along z axis
$I \in \Re^{3\times 3}$	the inertia matrix with respect to the body fixed frame where I_x,I_y and I_z are inertia with x,y,z axis respectively
$R \in SO(3)$	the rotation matrix from the body fixed frame to the inertial frame
$\Omega \in \Re^3$	angular velocity in the body fixed frame
$v \in \Re^3$	the velocity vector of the center of mass in the inertial frame
$T \in \Re$	the total thrust, i.e., $T = \sum_{n=1}^{4} f_i$
$\tau_i \in \Re^3$	the total moment vector in the body fixed frame
$\tau_{R,i} \in \Re$	the torque generate by the i^{th} propeller along z axis

The orientation of body fixed frame with respect to the inertial frame is represented by the rotation matrix R [13].

$$R = \begin{pmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - S_\phi C_\psi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{pmatrix}$$

where $C_{()} = cos()$ and $S_{()} = sin()$

The total thrust and the moment vector in the body fixed frame can be written as:

$$\begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ l & 0 & -l & 0 \\ -c_\tau & c_\tau & -c_\tau & c_\tau \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{1}$$

The above 4×4 matrix is invertible when $l \neq 0$ and $c_\tau \neq 0$ and equations are treated as control inputs in this paper [14].

The non linear model of the quadrotor can be represented taking ξ as state vector [15];

$$\dot{\xi} = f(\xi) + \sum_{n=1}^{4} g_i(\xi) u_i \tag{2}$$

where,

$$\xi^T = (x, v_x, y, v_y, z, v_z, \phi, p, \theta, q, \psi, r)$$

$$f(\xi) = \begin{bmatrix} v_x \\ 0 \\ v_y \\ 0 \\ v_z \\ -g \\ p + sin\phi tan\theta q + cos\phi tan\theta r \\ \frac{I_r}{I_x} q\Omega_r + \frac{I_y - I_z}{I_x} qr \\ cos\phi q - sin\phi r \\ \frac{I_r}{I_y} p\Omega_r + \frac{I_z - I_x}{I_y} pr \\ sin\phi sec\theta q + cos\phi sec\theta r \\ \frac{I_x - I_y}{I_z} pq \end{bmatrix} \tag{3}$$

$$g_1(\xi) = \begin{pmatrix} 0 & g_{11} & 0 & g_{12} & 0 & g_{13} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

$$g_2(\xi) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_x} & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

$$g_3(\xi) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_y} & 0 & 0 \end{pmatrix}^T$$

$$g_4(\xi) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{I_z} \end{pmatrix}^T$$

where,

$$g_{11} = \frac{1}{m}(cos\psi sin\theta cos\phi + sin\psi sin\phi),$$
$$g_{12} = \frac{1}{m}(sin\psi sin\theta cos\phi - sin\phi cos\psi),\ g_{13} = \frac{1}{m}(cos\theta cos\phi),$$
$$u_1 = T, u_2 = \tau_\phi, u_3 = \tau_\theta, u_4 = \tau_\psi.$$

The linearization of the quadrotor is done around the following equilibrium points:

$$\bar{\xi} = \begin{pmatrix} \bar{x} & 0 & \bar{y} & 0 & \bar{z} & 0 & 0 & 0 & 0 & 0 & \bar{\psi} & 0 \end{pmatrix}$$

and

$$\bar{u} = \begin{pmatrix} \bar{T} & 0 & 0 & 0 \end{pmatrix}$$

where, $\bar{T} = mg$. The linearized model of the quadrotor is controllable and observable at any of its equilibrium point [16].

B. Tracking Control

The control objective is to follow a given trajectory in the stable configuration manifold $\bar{\xi}$. The main objective of quadrotor is to track the desired trajectory. The overall control scheme is depicted in figure 1.

The control scheme is implemented by splitting the non-linear model of the quadrotor into four subsystems. Each subsystem is dedicated to control individual variable (for controlling of x, y, z, ψ). The idea here is to use a simple PD control algorithm by splitting the system into four subsystems. x and y subsystems have two second order equations, thus making it fourth order. z and ψ have two second order

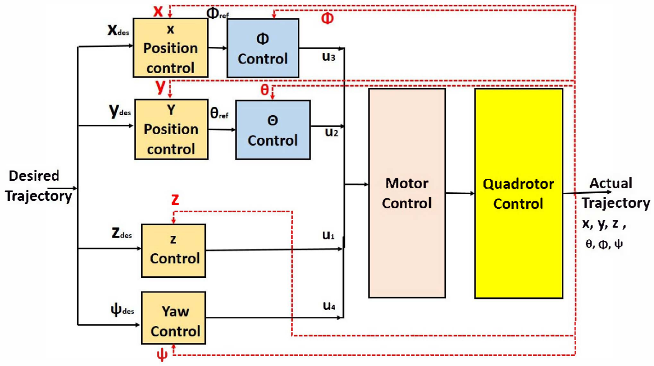


Fig. 1: Control Scheme for Quadrotor

equations. Consider the subsystem corresponding to x position. The Taylor series approximation of the fourth order system can be written as

$$\begin{aligned}\dot{x} &= v_x\\ \dot{v_x} &= \theta\frac{T}{m}\\ \dot{\theta} &= q\\ \dot{q} &= \frac{\tau_\theta}{I_y}\end{aligned}$$

Using two second order equations, the control law can be written as:

$$\theta_{ref} = kp_1(x_{des} - x) + kd_1(\dot{x}_{des} - \dot{x}) \tag{4}$$

and,

$$\tau_\theta = kp_2(\theta_{ref} - \theta) + kd_1(\dot{\theta}_{ref} - \dot{\theta}) \tag{5}$$

where, kp_1, kp_2, kd_1 and kd_2 are proportional and derivative control gains of the PD controller used in both subsystems respectively.

Therefore, the closed loop control law can be written as:

$$\begin{aligned}\tau_\theta = &\, kp_2(kp_1(x_{des} - x) + kd_1(\dot{x}_{des} - \dot{x}) - \theta)\\ &+ kd_2(kp_1(\dot{x}_{des} - \dot{x}) + kd_1(\ddot{x}_{des} - \ddot{x}) - \dot{\theta})\end{aligned}$$

The closed loop system of the linear approximation leads to the following transfer function

$$\frac{x}{x_{des}} = \frac{kd_1kd_2Ts^2 + (kd_1kp_2 + kp_1kd_2)Ts + kp_1kp_2T}{mI_ys^4 + mkd_2s^3 + (kd_1kd_2T + kp_2m)s^2 + (kp_1kd_2 + kd_1kp_2)Ts + kp_1kp_2T}$$

Above method needs several assumptions which are true in real time. For example, to do this control action $|\phi, \psi| \neq 90^o$, $T > 0$. Also, $\ddot{x}$ and $\ddot{x}_{des}$ should exist. That means there should not be any abrupt change in the given trajectory.

Similar analysis is carried out for y, z, ψ subsystems and it is found to be stable.

III. Computer Vision Based Trajectory Tracking

A. *Video Channel of Quadrotor Prototype*

The Quadrotor prototype used for this experiment is AR.Drone which has two on-board cameras, one pointing forward and one pointing downward. Downward camera is being used in our experiment which runs at 60 fps with a resolution of 176 × 144 pixels, covering a field of view of only 47.5° × 36.5° , but is afflicted only by negligible radial distortion, motion blur or rolling shutter effects. Both cameras are subjected to automatic brightness and contrast adjustment. It uses WiFi to transfer data. The video stream channel provides images from the frontal and/or bottom cameras.

B. *Vision based Pose Estimation*

The onboard webcam provides quadrotor a full autonomy and allows it to operate in almost any environment, without the need to setup a set of cameras. As long as the camera is previously calibrated (just needed once) and the set of markers are perfectly known (each marker size and their position, relative to each other), this system is ready to operate. The usage of markers allows an easy and fast computation so it can be used in real time.

The proposed method tracks pre-specified oriented roundel object continuously while maintaining a fixed distance from the roundel and also simultaneously keeping it approximately in the center of the image plane. During the following task, software utilizes the forward-facing camera images and part of the IMU (Inertial Measurement Unit) data to calculate the references for the four on-board low-level control loops. To obtain a stronger wind disturbance rejection and an improved navigation performance, a yaw heading reference based on the IMU data is internally kept and updated by our control algorithm.

C. *Pinhole Camera model*

Starting with the device that is able to capture the real world to a 2D structure, one of the most used simplified camera model is used the pinhole camera [17] [18]. This model aims to translate point $\tilde{X} = (X, Y, Z, 1)$ in real world into image points $\tilde{x} = (x, y, 1)$. It is based on the assumption that all the light passes through a single, infinitesimal point and is projected on the sensor, forming the image that everyone is familiar with. The pinhole model approximation doesn't need to consider the focusing issue and it considers that enough light passes through the hole.

Whole camera structure is abstracted to an infinitely small hole named as pinhole, and an image plane. The projection described by the model is known as perspective projection. The light rays which get projected are the one which pass through the hole and intersect with the image plane. The location of the pinhole is the optical center C of the camera. Focal length is the distance between optical center and image plane.

In reality the image plane is located behind the optical center and not in front of it, in the model this can be neglected without loss of generality. Every 3D point, whose connection line to the optical center intersects the image plane gets represented in the image. The 3D coordinates (X, Y, Z) of the point X are related to its 2D image point x with coordinates (x, y) by the following equations:

$$sm' = A[R|T]M'$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where,

- (X,Y,Z) are the coordinates of a 3D point in the world coordinate space.
- (u,v) are the coordinates of the projection points in pixels.
- A is the camera matrix, or a matrix of the intrinsic parameters.
- (c_x,c_y) is a principal point that is usually on the image center.
- f_x, f_y are focal lengths expressed in pixel units.

D. Extrinsic Camera Parameters

The location and orientation of the camera reference frame with respect to a known world reference frame is defined by Extrinsic Camera parameters. The unique transformation between the unknown camera reference frame and the known world reference frame is found by these parameters, which includes the translation vector between the relative positions of the origins of the two reference frames and the rotation matrix that brings the corresponding axes of the two frames into alignment.

E. Intrinsic Camera Parameters

The link of pixel coordinates of an image point with the corresponding coordinates in the camera reference frame is determined by these parameters. It characterize digital, optical and geometrical characteristics of the camera.

IV. Control Problem

This paper describes the development of a control strategy for a smooth autonomous maneuvering of quadrotor with target tracking and autonomous landing. The control approach seeks for full autonomy of the robot, by considering only internal sensors and processing. Quadrotor's relative pose is estimated through dead reckoning, and then proceed to control it via PD controller. A method was developed to estimate the relative position and orientation of the landing pattern with respect to the quadrotor using planar markers, an oriented roundel and computer vision algorithms. In order to help the identification of the moving platform, an Oriented Roundel is used. The image captured by the camera is processed and the markers must be detected and identified; computer vision algorithms are applied for estimating the pose of the quadrotor. This pose estimation is then used for controlling the quadrotor's twist and performing the required maneuvers to achieve the desired goal: lead the quadrotor to land on the platform autonomously.

V. Control Architecture

A. Architecture of Software Control System

The software architecture for the controller is supported by an interface with the sensor modules of the quadrotor which is shown in Fig. 2. The sensor module in the on-board system directly accesses the flight data, and is then transmitted to the remote workstation. The localization module which is usually present as an integral part of software architecture developed by the user, performs the task of localization of the system. The localization module presents the displacement of the system from the origin (position where the quadrotor transitions from taking-off to hover state). The global position is then provided as the coordinates to be operated upon by the path planning and tracking control algorithm. The GUI stands above the control algorithm, allowing the user to modify the preset values for destination coordinates. The control signals generated by the controller are then transferred back to the on-board actuator system, which activates the robot's rotor hardware system.

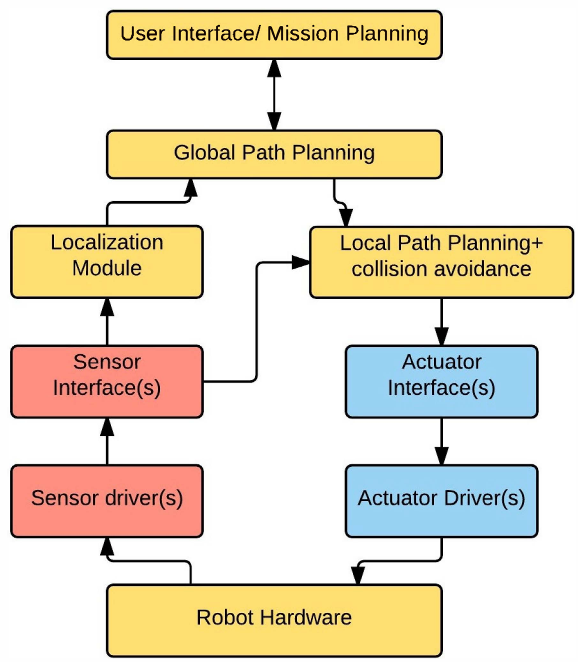


Fig. 2: Architecture of Software Control System

B. Path Planning Algorithm

The path-planning algorithm is modified from simple tracking controller to a tracking algorithm for the quadrotor to constantly hover over the roundel. The general flow of the algorithm is described in Fig. 3. After taking-off, the bottom camera of the quadrotor is used to search for the roundel in its area of coverage. If the roundel is within this area, the quadrotor moves by following the tracking control algorithm with the centroid of roundel as the destination coordinates. When the roundel is centered at the bottom-view camera, the quadrotor is hovering directly above the roundel and hence, landing command is executed simultaneously with searching. When the roundel escapes from the viewpoint, the default tracking control is in execution. When the landing command execution encounters the altitude below the threshold, the quadrotor is completely landed on the landing pad.

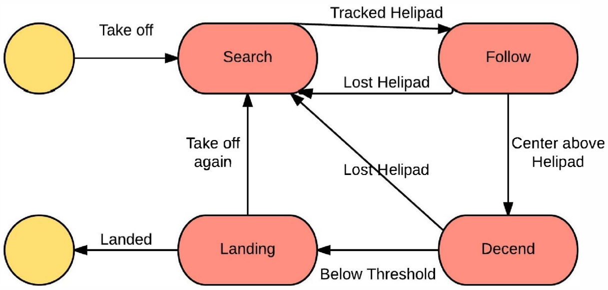


Fig. 3: Tracking State Diagram

C. Calibration of Camera

The task of determining the intrinsic parameters is called calibration of a camera. As the intrinsic parameters only depend on the camera itself (do not depend on its position and orientation), the calibration process in this paper is simplified by using a computational tool publicly available. The process consists in taking a set of pictures of a calibration board and let the algorithm calculate its intrinsic parameters. This tool also gives additional parameters such as lens distortions.

It is easy to calibrate both cameras using ROS (Robot Operating System) Camera Calibration package. After successful calibration, the commit button in the UI has to be pressed so that the driver will receive the data from the camera calibration node and save the information. From this point on, whenever driver is run on the same computer, saved file will be loaded automatically by the driver and information will be published to appropriate topic.

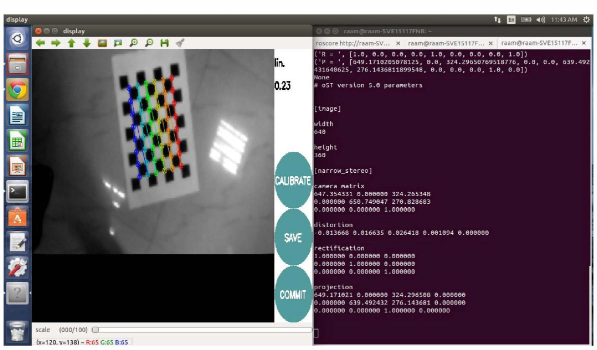


Fig. 4: Bottom Camera Calibration

D. Scale and Orientation Adaptive Mean Shift Tracking Algorithm (SOAMST)

for the scaled image, the so called SOAMST algorithm is used to identify the object [19]. The flow diagram of SOAMST is shown in Fig. 5.

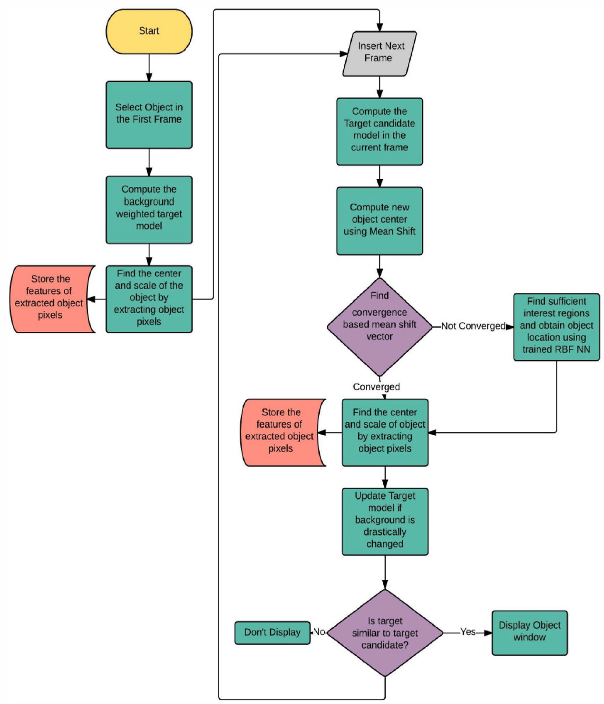


Fig. 5: Flow diagram of SOAMST Algorithm

VI. Experimental Results

For stable position estimation system, an oriented roundel is used as an external reference. The position estimation is done on the image pattern coordinates, allowing hovering over a fixed position, and takeoff and landing on the oriented roundel. For this whole process, the coordinates of roundel are calculated and fed to the controller.

For doing so, first of all, the quadrotor bottom camera field of view is measured and the relationship between image and real-world coordinates is maintained. This is done using the extrinsic and intrinsic parameters. Then, SOAMST algorithm for searching the desired oriented roundel is applied. Experimental results shows that the various maneuvers of the tracked object by SOAMST are flexible and robust.

The limitation of the present algorithm to track the moving object is limited to 1.5 meter altitude. This can be removed by changing the algorithms and a better camera.

The screenshots of the experimental video are shown in Fig. 7. The video of the experimental result is available in http://csrl.nitt.edu/quad_auto.mp4.

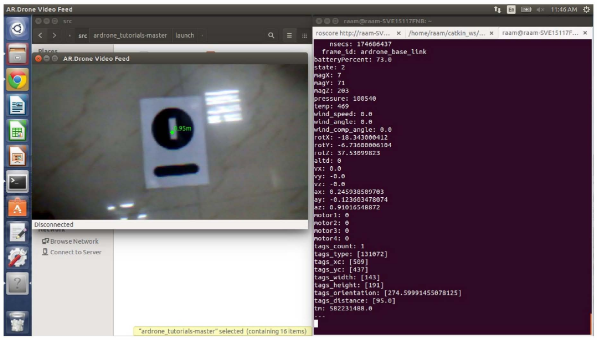
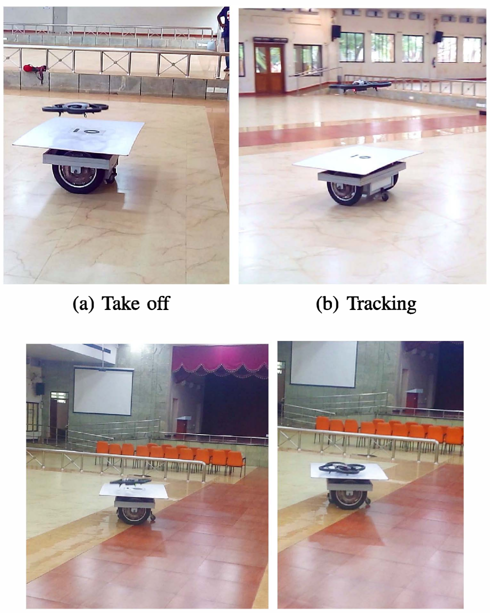


Fig. 6: Oriented Roundel Tracking



(a) Take off (b) Tracking

(c) Landing (d) Landing Finished

Fig. 7: Process of Autonomous Visual Tracking and Landing of a Quadrotor on a Helipad: a. Taking off, b. Tracking, c. Landing, d. Landing Finished

VII. Conclusion

A computer vision approach is proposed in this paper which tracks the pre-specified oriented roundel object continuously while maintaining a fixed distance from the roundel and also simultaneously keeping it approximately in the center of the image plane. Quadrotor's relative pose is estimated through dead reckoning and control approach is implemented which seeks for full autonomy of the robot, by considering only internal sensors and processing unit. Experimental results are presented.

References

[1] Z. Sarris, "Survey of uav applications in civil markets," 9th Mediterranean Conference on Control and Automations, Dubrovnik, Croatia, 27-29 June 2001.

[2] G. M. Hoffmann, H.Huang, S. waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: theory and experiment," AIAA Guidance, Navigation and Control Conference ad Exhibit, August 2007.

[3] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," vol. 3, IEEE International Conference on Robotics and Automation, ICRA, 11-15 May 2002, pp. 2799 – 2804.

[4] S. Saripalli and G. S. Sukhatme, "Landing on a moving target using an autonomous helicopter," IEEE International Conference on Robotics and Automation, 10-14 April 2007, pp. 2030–2035.

[5] H. Voos and H. Bou-Ammar, "Nonlinear tracking and landing controller for quadrotor aerial robots," 2010 IEEE International Conference on Control Applications Part of 2010 IEEE Multi-Conference on Systems and Control Yokohama, Japan, September 8-10 2010, pp. 2036–2041.

[6] J. Kim, Y. Jung, D. Lee, and D. H. Shim, "Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera," International Conference on Unmanned Aircraft Systems (ICUAS), May 27-30 2014, pp. 1243–1252.

[7] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a vtol uav on a moving platform using image-based visual servoing," IEEE International Conference on Robotics and Automation RiverCentre, Saint Paul, Minnesota, USA, May 14-18 2012, pp. 971–976.

[8] R. Mahony and V. Kumar, "Aerial robotics and the quadrotor [from the guest editors]," *Robotics and Automation Magazine, IEEE*, vol. 19, no. 3, p. 19, 2012.

[9] H. Lim, J. Park, D. Lee, , and H. J. Kim, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *Robotics and Automation Magazine, IEEE*, vol. 19, no. 3, pp. 33–35, 2012.

[10] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *Robotics and Automation Magazine, IEEE*, vol. 19, no. 3, pp. 46–56, 2012.

[11] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control : Balancing autonomy and human assistance with a group of quadrotor uavs," *Robotics and Automation Magazine, IEEE*, vol. 19, no. 3, pp. 57 – 68, 2012.

[12] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles, modelling, estimation and control of quadrotor," *IEEE Robotics and Automation Magazine*, pp. 20–32, September 2012.

[13] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "Backstepping/nonlinear H_∞ control for path tracking of a quadrotor unmanned aerial vehicle," 2008 American Control Conference Westin Seattle Hotel, Seattle, Washington, USA, June 11-13 2008, pp. 3356–3361.

[14] T. Lee, M. leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav for extreme maneuverability," 18th IFAC World Congress Milano(Italy), August 28-September 2 2011, pp. 6337–6342.

[15] M.Belkheiri, A. Rabhi, A. E. Hajjaji, and C. Pégard, "Different linearization control techniques for a quadrotor system," 2nd International Conference on Communications, Computing and Control Applications (CCCA), December 6-8 2012.

[16] A. Ataka, H. Tnunay, R. Inovan, M. Abdurrohman, H. Preastianto, A. Cahyadi, and Y. Yamamoto, "Controllability and observability analysis of the gain scheduling based linearization for uav quadrotor," IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (ROBIONETICS), November 25-27 2013, pp. 212 – 218.

[17] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 11 2000.

[18] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[19] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Scale and orientation adaptive mean shift tracking," *Computer Vision, IET*, vol. 6, pp. 52 – 61, 1 2012.