AUTONOMOUS TRAJECTORY TRACKING AND IMAGING VIA A QUADROTOR TYPE UAV: A SYSTEM INTEGRATION APPROACH

Mehmet Önder Efe[1], Mustafa Kaya[2], Munir Elfarra[3] and Rıdvan Özdemir[4]

[1] Univ. of the Turkish Aeronautical Association, Electrical and Electronics Engineering Department
Etimesgut, Ankara, Turkey, onderefe@ieee.org
[2,3] Univ. of the Turkish Aeronautical Association, Flight Training Department
Etimesgut, Ankara, Turkey, kaya.mustafa@thk.edu.tr
[4] Univ. of the Turkish Aeronautical Association, Ankara Aviation Vocational High School
Etimesgut, Ankara, Turkey, ozridvan@gmail.com

ABSTRACT

This paper presents an Unmanned Aerial Vehicle (UAV) based trajectory tracking and surveillance applications. Autonomy in a UAV system is a key property that is achieved via a number of sensors and well-studied software running in the behind. UAV system then becomes aware of its environment and the degree of awareness together with the capabilities determines the functional limits. This paper considers a quadrotor type UAV as its physics is straightforward to explain and the UAV flight is performed without human intervention. Open TLD system in integrated into the ground station and video frames acquired through the on-board camera module is processed by the ground operator. Any desired pattern can be marked and recognized by the UAV system. The whole system is tested experimentally and results have been found promising.

Keywords: Quadrotor uav, aerial surveillance, control

1. INTRODUCTION

The birth of the idea of UAVs dates back to the drawings of Leonardo da Vinci, who lived long before the success of manned flight. In the last century, the notable developments in this field can be summarized as follows: In 1910s, the tactical advantages and the associated importance provided by the use of UAVs are recognized particularly after the world War I. In 1940s, especially during the World War II, the V-1 UAVs of Nazi Germany was a great threat, and this was going to motivate laborious research efforts at the USA and in 60s, the UAVs was going to be used for surveillance purposes in Vietnam War. After 70s, other countries have initiated research in this field and new designs were developed by American researchers. The pioneering efforts were with Israel after 80s and several fleets of UAVs have been sold to some countries. The process since 1990s were mainly on the reduction of size in electronic circuitry, fast and reliable communication issues have all led to better command & control, and data communication performances.

Developing vision based functions has been a core issue in the contemporary UAV research and the topic is tightly blended with control and dynamical systems engineering. This is especially because of high level tasks in demanding environments and the low level solutions require robust controls schemes alleviating the disturbances. Many works have been reported in the literature and a descriptive subset can be summarized as follows: In [1], a rough map is detailed via camera observations and its resolution is adjusted using wavelet transformation. The updatable risk map is then used in planning the flight trajectory of a helicopter type UAV.

In [2], a micro aerial vehicle (MAV) is equipped with a camera detects the horizon and the developed MAV reduces the adverse effects of observation noise by using a Kalman filter. The controller is a PID and the major contribution is the use of a sole camera as the sensing hardware. In [3], Saripalli et al reports a system based on PC104 module and a PI type low level control scheme is utilized in a fuel powered UAV system. The target is assumed to carry a marker and the marker is recognized by using moment based techniques. In [4-6], Frew et al report a radio

controlled UAV system having a 2.9 m wingspan. In detection and positioning, techniques namely Bayesian pixel classification, Hough transform, composite component analysis are used. PI type control is used in [4] and receding horizon is exploited in [5-6]. In [7], Rathinam et al focus on patterns and regularities available in an image and exploit statistical information. The application is found to be successful even in 5 Hz sampling rate. In [8], synthesis of evolving trajectories is studied to avoid obstacles and it is highlighted that the proposed technique can be used in path planning. In [9], linear quadratic control scheme is used to achieve autonomous landing of SF40 type helicopter. The papers published recently consider genetic algorithms for path planning [10], laser and stereo vision odometry, which is fused with inertial measurement unit plus a Kalman filter, to enable seamless indoor and outdoor navigation [11]. In [12], vision based autonomous landing is studied and low cost nature of the proposed solution is emphasized. An uncalibrated camera and an airspeed sensor are utilized to develop the landing scheme. A virtual autonomous flight with a quadrotor UAV is studied in [13], where a dynamic model is developed in Matlab/Simulink environment and the model is linked to Google Earth application. Optical flow based virtual flight is performed in New York streets.
This paper is organized as follows: The second section introduces the test equipment; the third section presents the target tracking experiments. Next section is devoted to the trajectory tracking experiments, where the autonomous flight mode is discussed in detail. The conclusions are given at the end of the paper.

2. TEST EQUIPMENT AND SYSTEM INTEGRATION

2.1. Quadrotor System and Its Dynamics

Derivation of dynamic model of a quadrotor type UAV is the central part of the control implementation. In this part, derivation of the dynamic model of the quadrotor UAV is considered. The vehicle can be represented as a four rotor body as shown in Figs. 1-2.

The rotor rotation directions are opposite in order to balance torque produced by rotors therefore rotor 1 and rotor 3 are rotating in the counter-clockwise direction while rotor 2 and rotor 4 rotate in the clockwise direction. Altering the angular rotor speeds causes motion in the Cartesian space. Considering the hover state, a balanced increase/decrease in the thrusts causes change in the altitude. Varying the rotor 1 and rotor 3 angular speeds inversely proportional will result a motion in x direction because of the change in pitch angle. Varying the rotor 4 and rotor 2 angular speeds inversely proportional will result a motion in y direction because of the change in roll angle. Angular speed difference of four rotors will result drag torque and yaw motion. Quadrotor UAV is modeled under the following assumptions.

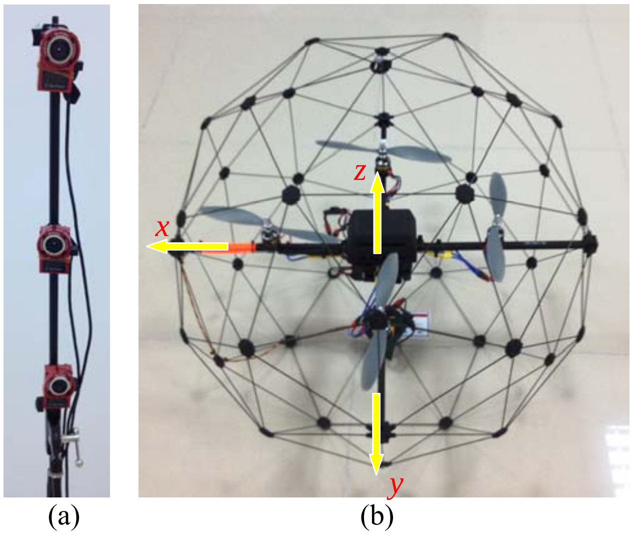

Figure 1. (a) Positioning cameras (b) Qball x4 quadrotor and its angle definitions

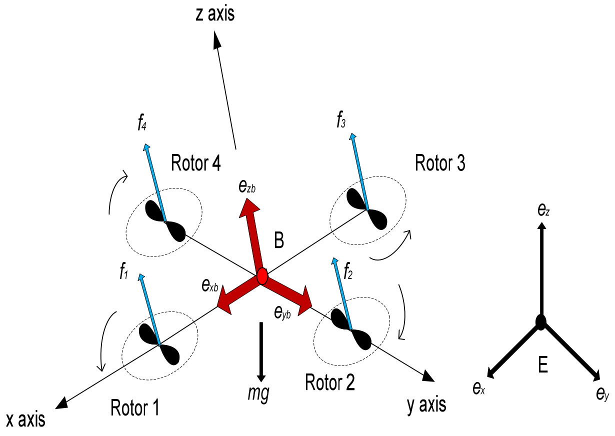

Figure 2. Quadrotor configuration and the choice of coordinate frames

- The quadrotor structure is rigid and symmetrical
- The quadrotor center of mass and body-fixed frame coincides

- Thrust and drag forces are proportional to the square of the propellers' speeds
- Ground effect is neglected
- The propellers are rigid

Suppose that E denotes earth fixed frame and B denotes body fixed frame which can be seen in Fig. 2, the airframe orientation is denoted by *R* matrix. *R* stands for the rotation from B to E. The dynamic model of quadrotor is derived from Newton-Euler approach. The dynamic equations of quadrotor can be written as follows [14].

$$\begin{bmatrix} mI_{3\times3} & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \dot{\vec{V}} \\ \dot{\vec{\omega}} \end{bmatrix} + \begin{bmatrix} \omega\times m\vec{V} \\ \omega\times I\vec{\omega} \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \qquad (1)$$

With the diagonal inertia matrix $I \in R^{3\times3}$, ω is the body angular velocity, $\vec{V}$ is the body linear speed vector. The dynamics in (1) can be explained as follows:

$$\dot{\zeta} = v \qquad (2)$$

$$\dot{v} = -ge_z + \frac{1}{m}TRe_z \qquad (3)$$

$$sk(\omega) = R^T\dot{R} \qquad (4)$$

$$I\dot{\omega} = -\omega\times I\omega + \tau_a - \tau_g \qquad (5)$$

where the $z = [x \;\; y \;\; z]^T$ denotes the position of the center of mass of the quadrotor *B* relative to *E* and describing linear motion of vehicle, *v* is a vector including the linear velocities of the body frame. Translational dynamics derived from Newton's second law of motion and ge_z is gravitational acceleration in *z* axis where $e_z=[0 \;\; 0 \;\; 1]^T$ denotes unit vector in earth-fixed frame in (3). *T* is the total thrust force generated by four rotors and *m* is the mass of the vehicle. The matrix *R*, given in (6), is an orthogonal homogeneous transformation matrix defined using Euler angles ϕ (roll), θ (pitch) and Ψ (yaw). *R* is used to transfer forces acting on the vehicle into earth fixed frame. The derivative of the rotational matrix to the rotational matrix by a matrix known as the Skew Symmetric Matrix in (4) where Ω represents the angular velocity of the craft relative to the body frame axis and x denotes the cross product of the two vectors. *V* represents any real vector. In (5) controllable inputs are available based on Newtonian relationship. Torque is equal to inertia multiplied by rotational acceleration. Torques present in the quadrotor is formulated in (5).

$$R(\phi,\theta,\psi) = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - s_\phi c_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \qquad (6)$$

Expression in (3) can be written as in (7), where *m* stands for the mass of the vehicle.

$$F_b = -mge_z + Re_z\sum_{i=1}^{4} T_i \qquad (7)$$

In (7) F_b is the result of the forces generated by four rotors. T_i is thrust generated by each rotor. Denoting ω as the angular body rate of the airframe in body-fixed frame, angular body rate and Euler angle parameterization relationship can be given as in (8).

$$\omega = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \qquad (8)$$

The quadrotor displays low amplitude angular motions letting us consider the small angle approximation for the angular body rates small angle approximation (SAA) has been considered during dynamic modeling studies. Bouabdallah et al [15] find simulation test results reasonable by modeling with SAA, i.e. $\omega = [\dot{\phi}\;\dot{\theta}\;\dot{\psi}]^T$ and the rotational dynamics of the quadrotor are described in (5). The gyroscopic effect due to rigid body rotation is given by $-\omega\times I\omega$ which is given in (9).

$$-\omega\times I\omega = \begin{bmatrix} (I_{yy} - I_{zz})\dot{\theta}\dot{\psi} \\ (I_{xx} - I_{yy})\dot{\theta}\dot{\phi} \\ (I_{zz} - I_{xx})\dot{\phi}\dot{\psi} \end{bmatrix} \qquad (9)$$

τ_g is gyroscopic effect due to propeller orientation change that needs to be handled carefully, [16]. τ_g is defined as:

$$\tau_g = \sum_{i=1}^{4}(\Omega\times J_r)(-1)^{i+1}\omega_i e_z \qquad (10)$$

J_r is the propeller inertia, ω is propeller angular speeds. The gyroscopic effect caused by propellers only effects the dynamics of the vehicle during roll and pitch motion which can be seen from equations above. The control moments denoted by τ_a, which are produced by the actuation periphery, is given in (11).

$$\tau_a = \begin{bmatrix} \tau_{roll} \\ \tau_{pitch} \\ \tau_{yaw} \end{bmatrix} = \begin{bmatrix} l(T_4 - T_2) \\ l(T_1 - T_3) \\ -Q_1 + Q_2 - Q_3 + Q_4 \end{bmatrix} \qquad (11)$$

With these terms, the complete dynamics of the vehicle is described in (12).

$$
\begin{aligned}
\ddot{x} &= (c_\phi s_\theta c_\psi + s_\phi s_\psi)\frac{1}{m}U_1 \\
\ddot{y} &= (c_\phi s_\theta s_\psi - s_\phi c_\psi)\frac{1}{m}U_1 \\
\ddot{z} &= -g + (c_\phi c_\theta)\frac{1}{m}U_1 \\
\ddot{\phi} &= \dot{\theta}\dot{\psi}\left[\frac{I_{yy}-I_{zz}}{I_{xx}}\right] + \frac{J_r}{I_{xx}}\dot{\theta}\Omega_d + \frac{l}{I_{xx}}U_2 \\
\ddot{\theta} &= \dot{\phi}\dot{\psi}\left[\frac{I_{zz}-I_{xx}}{I_{yy}}\right] - \frac{J_r}{I_{yy}}\dot{\phi}\Omega_d + \frac{l}{I_{yy}}U_3 \\
\ddot{\psi} &= \dot{\theta}\dot{\phi}\left[\frac{I_{xx}-I_{yy}}{I_{zz}}\right] + \frac{1}{I_{zz}}U_4
\end{aligned}
\tag{12}
$$

where m is the mass of the vehicle and the variable Ω_d seen in roll and pitch dynamics is defined as in (13). With this in mind, the control inputs U_1, U_2, U_3 and U_4 seen in (12) are defined in (14), where Ω_i is the angular speed (in radians per second) of the i-th rotor.

$$\Omega_d = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \tag{13}$$

$$
\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & b \\ -b & 0 & b & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{14}
$$

Defining $X = [\phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]^{\mathrm{T}}$ as the state of the attitude dynamics (i.e. $x_1 = \phi, x_2 = \dot{\phi}, x_3 = \theta, x_4 = \dot{\theta}, x_5 = \psi, x_6 = \dot{\psi}$), and $U = [U_1 U_2 U_3 U_4]^{\mathrm{T}}$ as the input vector, the state space representation of the dynamics can be given by $\dot{X} = f(X,U)$ where

$$
f(X,U) = \begin{bmatrix} x_2 \\ p_1 x_4 x_6 + p_2 x_4 \Omega_d + p_3 U_2 \\ x_4 \\ p_4 x_2 x_6 - p_5 x_2 \Omega_d + p_6 U_3 \\ x_6 \\ p_7 x_4 x_2 + p_8 U_4 \end{bmatrix} \tag{15}
$$

$p_1=(I_{yy}-I_{zz})/I_{xx}$, $p_2=J_r/I_{xx}$, $p_3= l/I_{xx}$, $p_4=(I_{zz}-I_{xx})/I_{yy}$, $p_5= J_r/I_{yy}$, $p_6=l/I_{yy}$, $p_7=(I_{xx}-I_{yy})/I_{zz}$, $p_8=1/I_{zz}$.

2.2. Control System

The quadrotor system used in this work has communication with the ground system via a wireless transceiver. The control system is developed in Matlab/Simulink® environment and downloaded to the vehicle over the wireless connection. The state of the system is read in a similar fashion and the control loop contains PID controllers to stabilize the attitude and the Cartesian positions and velocities.

2.3. Wireless Video Acquisition System

The Qball x4 system originally does not contain a camera system but we installed a lightweight wireless camera below the vehicle and a receiver is mounted to another computer that is devoted to monitor the vision based functions. The used camera is a COTS micro camera and the receiver is a radio receiver one shown in Fig. 3. With such a pair, the vehicle is able to provide video data for about 20 minutes with a 9V compact battery.

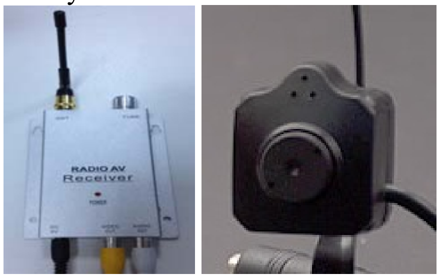

Figure 3. The receiver and the micro camera module

2.2. OpenCV Based Recognition System

Open Source Computer Vision Library (OpenCV) is a library of programming mainly focusing on real-time image processing. The library provides a common infrastructure for applications which require computer vision and machine learning. OpenCV was primarily built to accelerate the use of machine perception.

OpenCV was first developed by Intel®, and now supported by Itseez®. It is free for use under the open source BSD license [17]. Being a BSD-licensed product, OpenCV makes it easy for everyone to utilize and modify the code.

OpenCV has about 50,000 people of user community from all around the world. The estimated number of downloads exceeds 5 million since its first release. The library is used extensively in research groups, companies and by governmental bodies. OpenCV has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

OpenCV is written natively in C++ and has C++, C (therefore Matlab® through Matlab's MEX technology), Python and Java interfaces. It supports most of the commonly used operating systems. OpenCV leans mostly towards real-time vision applications and takes advantage of Intel's Integrated Performance Primitives when available. Main software developed in this study is based on Kalal et al.'s TLD Algorithm, [18]. TLD is an algorithm that simultaneously tracks, learns and detects an unknown object in a video stream. The algorithm makes minimal assumptions about the object, the scene or the camera's motion. It requires only initialization by a bounding box and operates in real-time. Fig. 4 illustrates a detailed block diagram of the TLD framework, [18].

The software used in this study and its graphical user interface (GUI) are both developed in Matlab®. TLD algorithm, abbreviated to represent Tracking, Learning and Detection, and OpenCV's optimized image processing routines (cross correlation of video frames and optical flow between two successive frames) are embedded into the Matlab code using MEX files, [19]. OpenCV v2.3.1 is used which was the latest release before initializing the study. In addition to the object tracking, the software developed may also be used for optical flow and edge detection problems.

3. FIRST EXPERIMENT: AERIAL SURVEILLANCE OF A STATIONARY TARGET

First experiment conducted in this paper is the recognition of a stationary target. The camera is mounted at the bottom of the vehicle and the flight is started. Due to the motion of the UAV, the target figure is sometimes in the picture and for some periods it is not seen. The goal is whenever the target is within the acquired image, software running at the ground station should mark it.

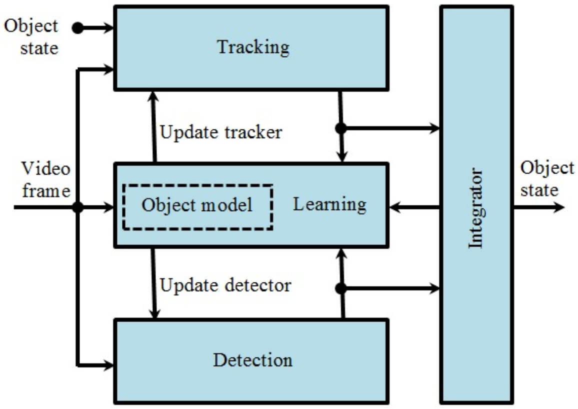

Figure 4. The macro level illustration of TLD framework

In fig.5, the trajectories followed in 3D space are shown. Top left plot is the followed trajectory that indicates a patrol flight over the target. The coordinates along *x*, *y* and *z* axes are shown in the remaining three subplots. The altitude (*z*) is about 35cm during the flight and no human intervention to vehicle occurred. In Fig.6, successive four different instants of autonomous video surveillance are shown. The target is marked and this behaviour is maintained till the ground station prescribes landing when *t*=70 sec.

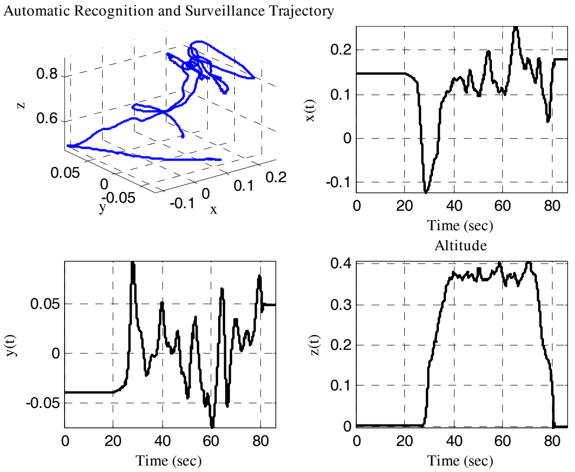

Figure 5. Results of the target tracking experiment.

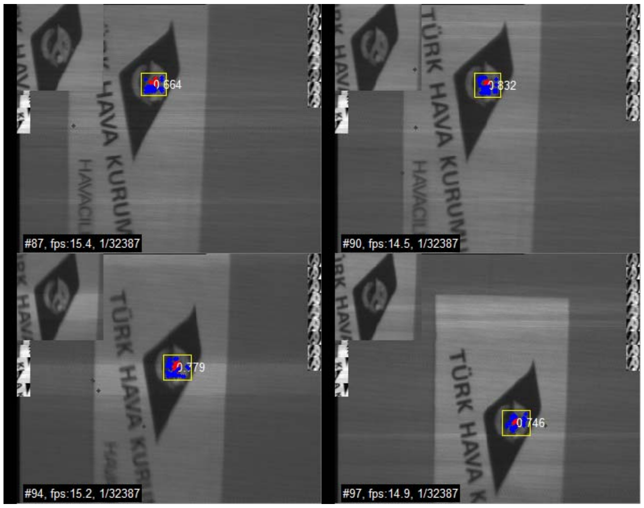

Figure 6. Successive four different instants of the experiment. UAV tracks the chosen logo.

4. SECOND EXPERIMENT: AERIAL SURVEILLANCE OF A MOVING TARGET

In the second experiment, we consider a moving target scenario. Since the ultrasonic sensor located at the bottom of the UAV is sensitive to objects moving on the ground, we consider an ellipsoid laser image as the target. The UAV is not fixed and the target is not fixed either. The problem under such conditions is considered as a challenging scenario. According to the results shown in Fig. 7, the UAV performs a surveillance flight for about 90 seconds and it traverses a trajectory above the feature being tracked.

Four different instants of flight and the acquired video images are shown in Fig. 8, where we see a background image that is not smooth and an ellipsoid image being tracked over such a background. The laser image is marked by a rectangle whenever it falls within the camera range. This experiment simply shows us that an autonomous UAV flight with the goal of detecting a moving target can be achieved and the experiment is considered as a proof of concept.

5. CONCLUSIONS

In this paper, we report an autonomous UAV flight for surveillance. The UAV motion was not intervened by a human operator and the two experiments, namely, the aerial surveillance of a stationary target and a moving target have yielded promising results. The internal controls have been achieved by PID loops and the UAV system has performed successful wireless data transfer to the ground station.

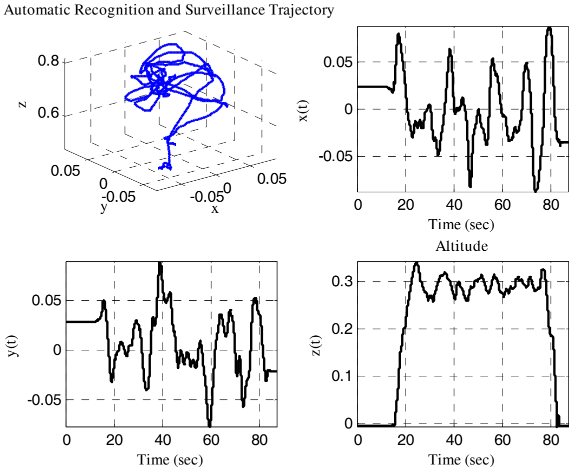

Figure 7. Results of the target tracking experiment.

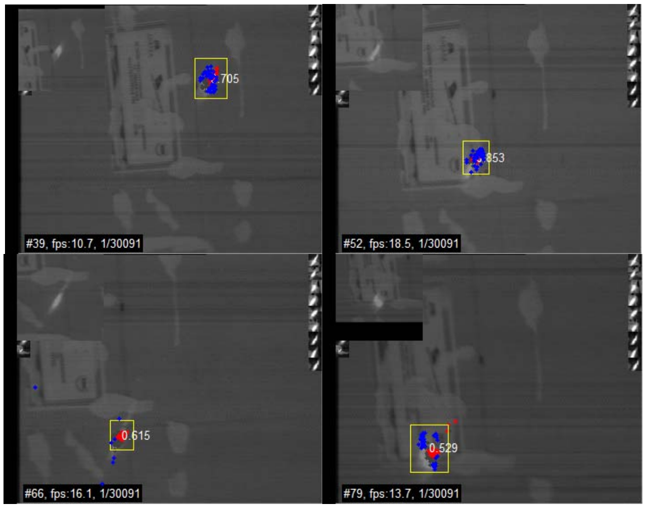

Figure 8. Successive two different instants of the experiment. UAV tracks the ellipsoid laser light moving on the ground.

The contribution of the current study is to demonstrate successful system integration for aerial surveillance and the future aim of the authors is to perform such an aerial surveillance for outdoor scenarios.

6. ACKNOWLEDGMENTS

Efforts of TLD system developers are gratefully appreciated. The authors thank Robit Tech. for hardware support. The funding from Ankara Development Agency via TR51/11/YEN0025 project is gratefully acknowledged.

REFERENCES

[1] B. Sinopolli, M. Micheli, G. Donato, T.J. Koo, “Vision Based Navigation for an Unmanned Aerial Vehicle,” Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA’01), Seoul, Korea, pp.1-8, 2001.

[2] S.M. Ettinger, M.C. Nechyba, P.G. Ifju, M. Waszak, “Vision-Guided Flight Stability and Control for Micro Air Vehicles,” Proc. IEEE Int. Conf. on Intelligent Robots and Systems, v.3, pp. 2134-40, 2002.

[3] S. Saripalli, J.F. Montgomery, G.S. Sukhatme, "Vision-based Autonomous Landing of an Unmanned Aerial Cehicle," Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'02), v.3, May 11-15, pp.2799-2804, 2002.

[4] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, “Vision-Based Road-Following Using a Small Autonomous Aircraft,” Prof. of the 2004 IEEE Aerospace Conference, v.5, March 6-13 , pp.3006 – 3015, 2004.

[5] E. Frew, “Receding Horizon Control Using Random Search for UAV Navigation with Passive, Non-cooperative Sensing,” Proc. of the 2005 AIAA Guidance, Navigation, and Control Conference, San Francisco, CA, August 2005.

[6] E. Frew, J. Langelaan, S. Joo, “Adaptive Receding Horizon Control for Vision-Based Navigation Small Unmanned Aircraft,” American Control Conference (ACC’06), 14-16 June 2006.

[7] S. Rathinam, Z. Kim, A. Soghikian, R. Sengupta, “Vision Based Following of Locally Linear Structures using an Unmanned Aerial Vehicle,” Decision and Control, 2005 and 2005 European Control Conference (CDC-ECC'05), December 12-15, pp. 6085- 6090, 2005.

[8] J.J. Kehoe, A.S. Watkins, R. Lind, “A Time Varying Hybrid Model for Dynamic Motion Planning of an Unmanned Vehicle,” AIAA Guidance, Navigation and Control Conference and Exhibit, August 21-24, Keystone CO, USA, 2006.

[9] Z. Yu, K. Nonami, J. Shin, D. Celestino, “3D Vision Based Landing Contorl of a Small Scale Autonomous Helicopter,” Int. J. Advanced Robotic Systems, v.4, n.1, pp.51-56, 2007.

[10] Y.V. Pehlivanoğlu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," Aerospace Science and Technology, vol.16, no.1, pp.47-55, 2012.

[11] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I.L. Grixa, F. Ruess, M. Suppa, D. Burschka, "Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue," IEEE Robotics & Automation Magazine, vol.19, no.3, pp.46-56, 2012.

[12] G. Anitha, R.N.G. Kumar, "Vision Based Autonomous Landing of an Unmanned Aerial Vehicle," Procedia Engineering, vol.38, pp.2250-2256, 2012.

[13] A. Eresen, N. İmamoğlu and M.Ö. Efe, "Autonomous Quadrotor Flight with Vision-based Obstacle Avoidance in Virtual Environment," Expert Systems with Applications, vol.39, pp.894-905, 2012.

[14] H. Bouadi, M. Bouchoucha and M. Tadjine, “Modelling and Stabilizing Control Laws Design Based on Sliding Mode for an UAV Type-Quadrotor”, IEEE Intelligent Vehicles Symposium, 2007.

[15] S. Bouabdallah, R. Siegwart, “Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor”, IEEE International Conference on Robotics and Automation, Page(s):2247 – 2252, 2005.

[16] M. Krstic, I. Kanellakopoulos, P. Kokotovic, “Nonlinear and Adaptive Control Design”, Page(s): 88-99, John Wiley& Sons,Inc., USA, 1995.

[17] http://opensource.org/licenses/BSD-2-Clause, Last accessed Jan. 24, 2012.

[18] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-Learning-Detection, “IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.34, no.7, pp:1409-1422, July 2012.

[19] http://www.mathworks.com/help/matlab/matlab_external/introducing-mex-files.html, Last accessed Jan. 24, 2012.