 ISBN: 978-979-1421-19-5

Simulation of Intelligent Unmanned Aerial Vehicle (UAV) For Military Surveillance

M. Anwar Ma'sum, M. Kholid Arrofi, Grafika Jati, Futuhal Arifin, M. Nanda Kurniawan, Petrus Mursanto, Wisnu Jatmiko
Faculty of Computer Science, Universitas Indonesia
Email: anwar.rejoso@gmail.com

***Abstract*—Nowadays, Unmanned Aerial Vehicle (UAV) is an important technology for military and security application. Various missions can be done using UAV such as surveillance in unknown areas, forestry conservation, and spying enemy territory. Application which is developed in this research has a purpose to simulate condition in war zone for spying the enemy. Platform used in the experiment is Parrot AR.Drone ver.2.0, an mini quadrotor which was developed by Parrot SA. This quadrotor controlled by Robot Operating System (ROS) framework. The quadrotor will search and recognize some objects and locate their location. Many algorithms were used to do the mission. To recognize object Adaboost Classifier and Pinhole Algorithm were used. The result shows that average error for all scenario is only 0.24 meters.**

I. Introduction

Unmanned Aerial Vehicle (UAV) is an aircraft that can maneuver without a human pilot on board. There has been tremendous progress on development of UAV for various purposes. One example is development of UAV for military operation which is operated in urban terrain [1]. Other example is development of UAV for surveillance purpose to gather civilian air traffic and national air space information [2].

In Indonesia development of UAV is carried out for surveillance purpose as well. UAV is chosen as the agent, because it is capable to maneuver freely on the air, while grounds robot will difficulty maneuver at extreme terrain such as forest. Technically, it is used for monitoring operation in borderland as defensive military patrol. UAV is also used for monitoring at natural conservation area, such as conservative forest, and wildlife cage. It is due to many cases of illegal action in Indonesia, such as illegal logging, illegal mining, and animal hunting. In other side, the number of officers charged usually is not proportional with the width of the area. Therefore, UAV is strongly needed to help military surveillance operation.

Quadrotor is one kind of UAV that using four rotor as its actuator to move and maneuver. Quadrotor is capable to move in any direction both horizontal, vertical and its combination. Furthermore, it is also capable to rotate horizontally with its center of mass as rotation axis. The more important, it is also capable to hover while flying on the air. Therefore, quadrotor can maintain its position while gathering data. Since its technical capabilities, quadrotor is used as operation agent in several researches in intelligent UAV.

The aim of this research is to develop simulation of intelligent UAV for military surveillance. In this research, we use famous quadrotor named AR.Drone as agent. As software platform, we use Robot Operating System (ROS) in this research. During surveillance operation, UAV is assigned to patrol in an area. During patrol, UAV will search suspicious objects in the area. Then UAV will calculate objects position. Objects location is calculated by UAV coordinate and objects relative coordinate to UAV. To detect suspicious object from UAV camera view, we use Haar Like Features, and Adaboost classifier. To estimate UAV position relative to start point, we use Extended Kalman Filter (EKF) algorithm. To determine target position, we use Pinhole algorithm. After completing patrol, UAV will generate map of the area, contains its trajectory and suspicious objects location found by UAV.

Several researches have already been done. Shafique et al develop simple algorithm for detection of elliptical objects in remotely sensed images for UAV applications [3]. However, it is limited for objects that can be segmented as elliptical curve. Other work done by Teutsch et al develop algorithm for detection, segmentation, and tracking of moving objects in UAV Videos [4]. However, it does provide objects location in real space. Other work done by Ibrahim et al develop moving objects detection and tracking framework for UAV-based surveillance [5]. However, the framework is not integrated in ROS, so that cannot be used directly by researcher which uses ROS platform. Engel et al develop accurate figure flying with a quadrocopter using onboard visual and inertial sensing. This research uses Kalman Filter adapted in this research to estimate UAV position [6]. As track record, in previous research we develop

ISBN: 978-979-1421-19-5

swarm quadrotors for telecommunication network coverage area expansion in disaster area [7]. In that research, we implement Self Deployment Algorithm proposed by Takahashi [8].

The rest of this paper is organized as follows : Section 2 discusses proposed framework in the research. Section 3 discusses the algorithm used to build artificial intelligence in UAV. Section 4 discusses the experiments and the results, and section 5 draws the conclusions. Last, section 6 discus about future work of this research.

II. Proposed Framework

A. System Design

The design of the system is shown as Figure 1. In this simulation study, we use real world environment consists of field, UAV and target object.

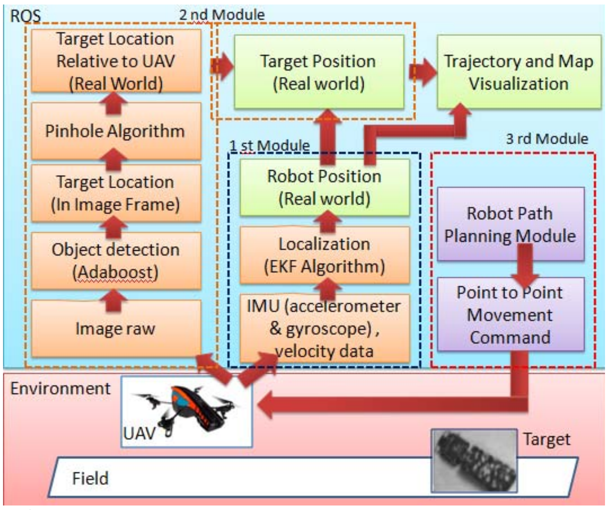


Figure 1. Simulation of intelligent UAV for military surveillance system architecture

The mission of UAV are explore whole field area, find and localize target position, and generate map of the area.

We build artificial intelligence in UAV system to carry out the mission. Artificial intelligence is implemented in personal computer (PC) using ROS platform, due to restriction that UAV cannot be programmed directly. There are three main modules in the UAV intelligence system. First module is estimation of robot position. This module is started by navigation data acquisition. Navigation data consists inertial measurement unit (IMU) which is represent of UAV orientation, and UAV velocity data. Then, navigation data is processed by Extended Kalman Filter (EKF) algorithm to determine robot position in real world space. Second module is estimation of target position. This module is started by image data acquisition from UAV camera sensor. Then, image data is processed using Adaboost classifier to detect target object appearance. If target object is detected in the image frame, then coordinate of the object is saved for next process. Object coordinate information in image frame, camera focus length data, and UAV altitude are used to determine target position relative to UAV using Pinhole algorithm. The target position data is in real world space. Using robot position data and target position relative to UAV data, target position can be determined. Then, robot position and target position are visualized in trajectory or map view. The third module is robot path planning module that manage robot movement during exploration.

B. Quadrotor and Target Object

Quadrotor used in this experiment is famous quadrotor named AR.Drone version 2.0 [9]. AR.Drone 2.0 is the next version of quadrotor developed by Parrot for entertainment purpose. AR.Drone 2.0 can be controlled directly via wifi connection from PC or using smart phone with Android or iOS platform. AR.Drone 2.0 is equipped with inertial measurement unit (IMU) sensor, sonar sensor and two camera sensor, front camera and bottom camera. IMU sensor is used to stabilize AR.Drone 2.0 when landing or take off, sonar sensor is used to determine altitude flight, and both camera is used to sensing from horizontal and vertical view. Technical specification of AR.Drone 2.0 used in this experiment is shown in Figure 2.

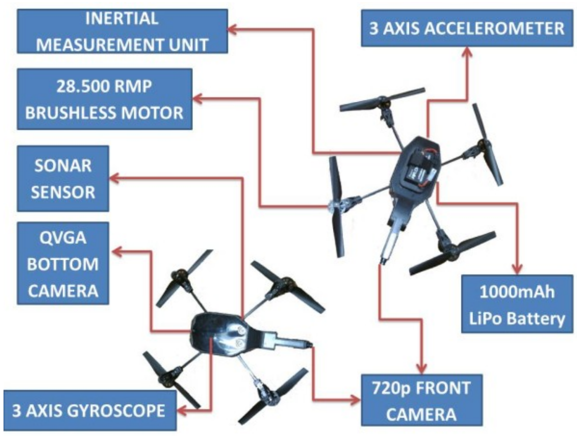


Figure 2. AR.Drone 2.

In this research we use military vehicle miniature as target object. We use two type of vehicle, they are truck and raft type. These objects are chosen to simulate real object in the field. Target objects used in this research is viewed in Figure 3.

Figure 3. Target Objects

C. Software Platform

Robot Operating System (ROS) was used as runtime environment for this AR.Drone application. All the programs in this research were managed by ROS. We can said that ROS has a role as mini operating system to operate robot. Some programs those run at top of ROS may run at the same time. It is happened because ROS can do multi-process communication. ROS also has a set of tools to develop robot system rapidly [10]. ROS platform is developed by familiar programming languages like C, C++ and Python. One of the most important terminology in ROS is node. The node in ROS is an application or package that do computation process. ROS node may run in parallel and communicate each other through ROS message called topic. Node that sends topic is called "publisher". On the other hand, node which receives message was named "subscriber". Open source community has developed many programs for AR.Drone quadrotor, such as ardrone_autonomy which have a role as AR.Drone driver. This was developed by Autonomy Laboratory [11].

III. ALGORITHM

This section discuss several concepts and algorithm used in this research. Sub section A discuss about dataset. Sub section B discusses Adaboost Algorithm used for object detection. Section C discusses Extended Kalman Filter algorithm used to estimate robot position.

A. Dataset

As mention before, that in order to detect object we used classification approach. This approach uses data set to train the system to be capable in detecting object. Data set used in this research is taken directly from quadrotor bottom camera. Since we use binary classification, dataset consists of two kind of data, positive data and negative data. Positive data is image which contains target objects. Positive images are taken by various angle (orientation), and objects location in the image. Positive images also taken from various quadrotor altitude from 80 cm to 120 cm. Positive images used consists of 250 sample images. Positive images used as dataset in this research is shown in Figure 4.

Beside positive images, negative images are used in as dataset. Negative images are background images where target objects taken. However, negative images do not contain target object as positive images. Negatives image samples used in this research are shown in Figure 5. In figure left, left image is floor image without floor line. Center image is floor image with floor line, and right image is floor image with floor line and shadow. These variation are used in order to gain better classification performance.

Before processing data to classifier, we need to calculate features value. We use Viola Jones Haar Like Feature used in previous research for face detection [12]. Haar Like Features basis are show in figure 6. In implementation, Haar Like Features can be adjusted to various sizes. The value of a feature is difference between number pixel in black rectangle and number pixel in white rectangle. Figure 6.a consists of feature to detect edge pattern, Figure 6.b consists of feature to detect line pattern, whereas Figure 6.c consists of feature to detect center surrounding pattern.

Figure 4. Positive images data

Figure 5. Negative images data

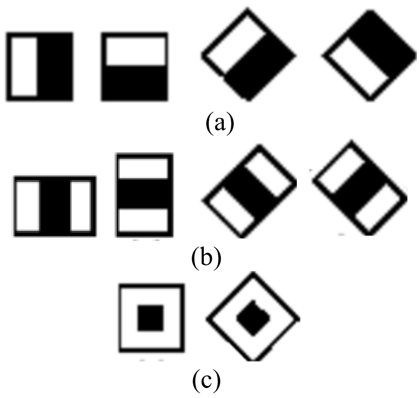


Figure 6. Viola Jones Haar Like Features

B. Adaboost Classifier

Adaboost is used in previous research for face detection [12]. The main idea of Adaboost Classifier is an ensemble of multiple weak classifier. Weak classifier is simple classifier such as threshold function. As mention before that we use Haar Like Features. From each Haar Like Features will be built a weak classifier with threshold function correspondent to it. Threshold value will be adjusted during training process. Several features (weak classifier) then were chosen to built strong classifier. Here are steps of Adaboost classifier :

1). Given N examples $(x_1,y_1),\ldots,(x_i,y_i),\ldots,(x_N,y_N)$, *where* $y_i \in \{-1,1\}$

2). Initialize samples weight

$$\omega_{1,i} = \begin{cases} \frac{1}{m}, if\ y_i = 0 \\ \frac{1}{l}, if\ y_i = 1 \end{cases} \quad (1)$$

where m and l are number of negatives and positives respectively.

3). For t = 1,....., T do

(a). For each feature j, train a classifier $h_j()$

(b). Evaluate the error of the classifier

$$\varepsilon_i = \sum_{i=1}^{N} \omega_{t,i}\ b_i \quad (2)$$

(c). Choose a classifier $h_t()$ with lowest error ε_t

(d) Update weights

$$\omega_{t+1,i} = \omega_{t,i\ t,i}\ \beta_t^{1-bi} \quad (3)$$

Where,

$$b_i = \begin{cases} 0, if\ \ \mathrm{ht}(x\ \mathrm{i}) = yi \\ 1, otherwise \end{cases} \quad (4)$$

With

$$\beta_t = \varepsilon_t / (1 - \varepsilon_t) \quad (5)$$

4). Output strong classifier:

$$H(x) = \begin{cases} 1,\ if\ \sum_{t=1}^{T} \alpha_t . h_t \geq 0.5 \sum_{t=1}^{T} \alpha_t \\ -1,\ otherwise \end{cases} \quad (6)$$

With

$$\alpha_t = \log(1/\beta_t) \quad (7)$$

C. Extended Kalman Filter (EKF) Algorithm

Extended Kalman Filter (EKF) algorithm is used to estimate robots position. EKF estimation in quadrotor originally proposed by Engel et al [6]. In this section, the state space, the observation models and the motion model used in the EKF is described. The state space consist of ten state variables.

$$x_t := (x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t, \alpha_t, \beta_t, \gamma_t, \dot{\gamma}_t)^T \in \mathbb{R}^{10} \quad (8)$$

Where (x_t, y_t, z_t) denotes the position of the quadrotor in m and $(\dot{x}_t, \dot{y}_t, \dot{z}_t)$ denotes the velocity in $^m/_s$, both in world coordinates. Next, the state contains the roll α_t, pitch β_t and yaw γ_t angle of the quadrotor in deg, as well as the yaw-rotational speed $\dot{\gamma}_t$ in $^{deg}/_s$.

The prediction model describes how the state vector x_t changes from one time step to the next. Particularly, the quadrotor's horizontal acceleration $\ddot{x}$ and $\ddot{y}$ is estimated based on its current state x_t. In other side, $\ddot{x}$ and $\ddot{y}$ estimate quadrotor vertical acceleration $\ddot{z}$, yaw-rotational acceleration $\ddot{\gamma}$ and roll/pitch rotational speed $\dot{\alpha}, \dot{\beta}$ based on the state of x_t and the active command control u_t

The horizontal acceleration is proportional to the horizontal force acting upon the quadrotor, which is given by

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} \propto \boldsymbol{f}_{acc} - \boldsymbol{f}_{drag} \quad (9)$$

Where $\boldsymbol{f}_{drag}$ indicates the drag and $\boldsymbol{f}_{acc}$ indicates the accelerating force. The accelerating force depends on the tilt angle (α_t, β_t), then affect horizontal velocity of the quadrotor. It is approximated by projecting the quadrotor's z-axis onto the horizontal plane, which is can be written as :

$$\ddot{x}(x_t) = c_1(\cos\gamma_t \sin\alpha_t \cos\beta_t - sin\,\gamma_t sin\beta_t) - c_2\dot{x}_t \quad (10)$$

$$\ddot{y}(x_t) = c_1(-\sin\gamma_t \sin\alpha_t \cos\beta_t - cos\,\gamma_t sin\beta_t) - c_2\dot{y}_t \quad (11)$$

Where c_1 and c_2 are constant calibrated in test flights. This model assumes that four rotors generated constant thrust. Additionally, we describe the influence of sent control commands $u_t = (\bar{\alpha}_t, \bar{\beta}_t, \bar{z}_t, \bar{\gamma}_t)$ by a linear model:

$$\dot{\alpha}(x_t, u_t) = c_3\bar{\alpha}_t - c_4\bar{\alpha}_t \quad (12)$$
$$\dot{\beta}(x_t, u_t) = c_3\bar{\beta}_t - c_4\bar{\beta}_t \quad (13)$$
$$\ddot{\gamma}(x_t, u_t) = c_5\bar{\gamma}_t - c_4\bar{\gamma}_t \quad (14)$$
$$\ddot{z}(x_t, u_t) = c_7\bar{z}_t - c_8\bar{z}_t \quad (15)$$

Coefficients $c_3, \ldots, c_8$ are also estimated from test flight data. Overall, state transition is given as

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \\ \dot{x}_{t+1} \\ \dot{y}_{t+1} \\ \dot{z}_{t+1} \\ \alpha_{t+1} \\ \beta_{t+1} \\ \gamma_{t+1} \\ \dot{\gamma}_{t+1} \end{pmatrix} \leftarrow \begin{pmatrix} x_t \\ y_t \\ z_t \\ \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \\ \alpha_t \\ \beta_t \\ \gamma_t \\ \dot{\gamma}_t \end{pmatrix} + \delta_t \begin{pmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \\ \ddot{x}(\mathbf{x}_t) \\ \ddot{y}(\mathbf{x}_t) \\ \ddot{z}(\mathbf{x}_t, \mathbf{u}_t) \\ \dot{\alpha}(\mathbf{x}_t, \mathbf{u}_t) \\ \dot{\beta}(\mathbf{x}_t, \mathbf{u}_t) \\ \dot{\gamma}_t \\ \ddot{\gamma}(\mathbf{x}_t, \mathbf{u}_t) \end{pmatrix} \quad (16)$$

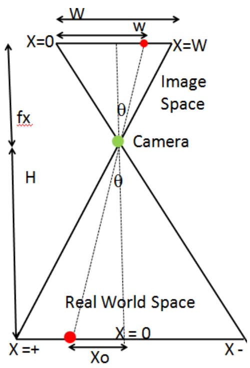


Figure 7. Pinhole Principle

D. Pinhole Algorithm

Pinhole algorithm is used to determine objet position relative to quadrotor. This is a simple algorithm that uses object coordinate in the image (w), image width

(W), camera focal length (fx), and distance between object and camera (H) to determine object coordinate in real world space. Illustration of pinhole principle is shown in Figure 7.

From illustration in figure 7, we can denote that :

$$tg(\theta) = \frac{(w-0.5\,W)}{fx} = \frac{(Xo)}{H} \quad (17)$$

Therefore,

$$Xo = \frac{H.(w-0.5\,W)}{fx} \quad (18)$$

Using the same analogy, we can compute object coordinate in real world Y axis as below :

$$Yo = \frac{H.(0.5\,Y-y)}{fy} \quad (19)$$

Where Yo is object coordinate in real world Y axis, y is object coordinate in pixel space, and fy is focal length for Y axis. Focal length (fx and fy) is obtained from camera calibration.

IV. EXPERIMENT AND RESULT

In this research, we conduct two main experiments. First experiment is conducted to measure object detection accuracy. In this experiment, we capture image from quadrotor camera sensor. Image contains target object in various position and orientation. Object detection result is shown in Figure 8.

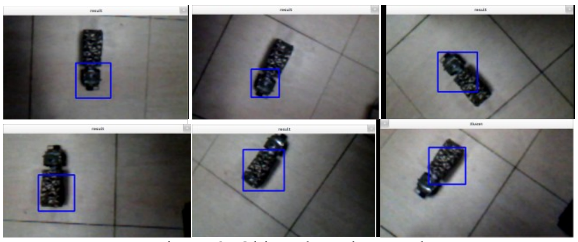


Figure 8. Object detection result

To evaluate object detection performance, we count hit, miss, and false number in the experiment. From 100 tests have been done, we have results as shown in Table I.

TABLE I. Object Detection Evaluation

Number of tests	Hit	Miss	False
100	71	29	21

Table I shows that hit rate of the object detection is 71 %. False rate of the system is 21%. Although the hit rate of the system object detection is not very high, the false rate of the system is low enough. Therefore, object detection module is reliable to use in surveillance simulation.

In the second experiment, we simulate surveillance patrol using quadrotor. In this simulation, quadrotor is commanded to explore an area called field. Then, quadrotor must localize suspicious objects detected, based on its location in the field and objects locations relative to quadrotors. Experimental environment of the simulation is shown if Figure 9.

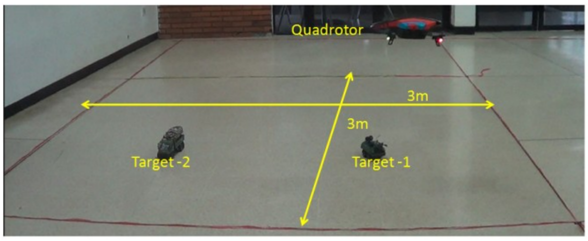


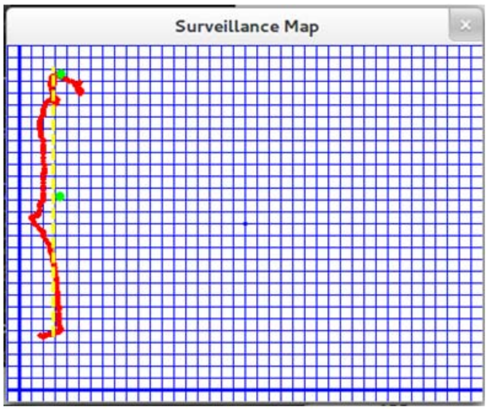
Figure 9. Experimental Environment

In this patrol simulation, we use four scenario. Each scenario has different objects position. In this experiment, we evaluate objects localization performance by calculate error (difference) between real position and estimated position. Object localization performance can be seen in Table II.

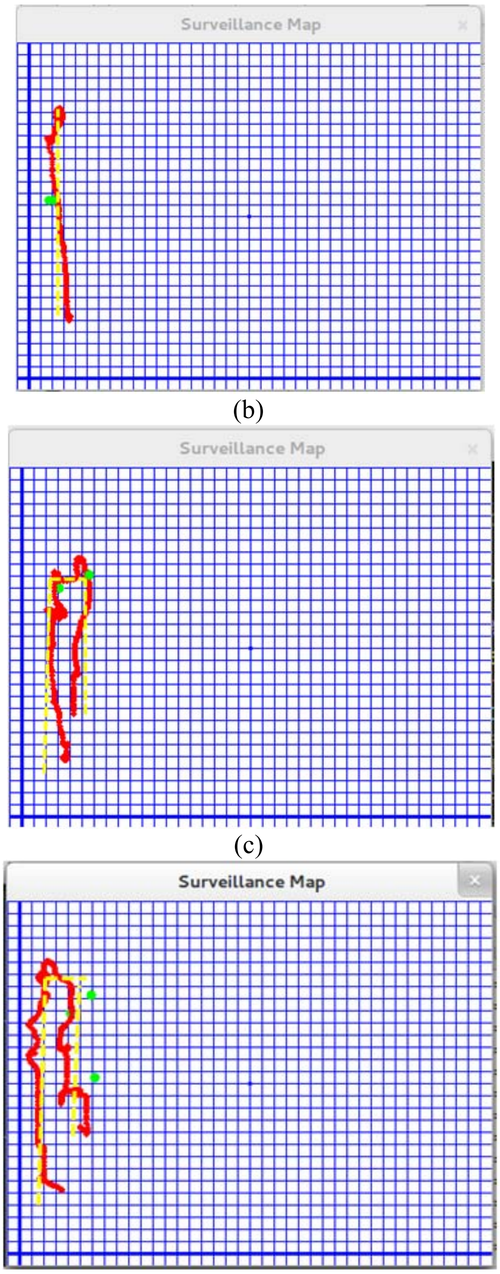
TABLE II. Scenario Result

Scenario	Object	Real position (m)		Estimated position (m)		Error (m)
		X	Y	X	Y	
1	1	0.30	1.20	0.30	1.43	0.23
	2	0.30	2.40	0.31	2.45	0.05
2	1	0.30	1.50	0.02	1.33	0.32
	2	*object not detected*				
3	1	0.30	1.50	0.23	1.69	0.21
	2	0.90	2.10	0.73	1.81	0.34
4	1	0.90	0.90	0.86	1.24	0.35
	2	0.90	2.10	0.80	1.93	0.20
Average						0.24

Table II shows that mean error of object position estimation is 0.24m (24 cm). After completing patrol, quadrotor will generate a map from its surveillance includes object position detected. Map generated from each scenario is shown in Figure 10.



(a)



(b)

(c)

(d)

Figure 10. Map generated from each scenario

In Figure 10, red point represents quadrotors position during patrol. As seen in the figure, that quadrotors path is not smooth as designed (yellow color).

V. CONCLUSION

A simulation of intelligent UAV for military surveillance is successfully implemented. The system is implemented using AR.Drone quadrotos, and ROS platform. Haar Like Features and Adaboost Classifier are used to develop object detection system. Performance of the object detection is 71% hit rate and 21 % false rate. To estimate object position in the field area, we use Extended Kalman Filter and Pinhole algorithm. Experimental simulation shows that the estimation methods have error about 0.24 m (24 cm).

VI. FUTURE WORK

In the future work, we try to propose object recognition algorithm. Therefore, UAV can differentiate two or more object targets. Besides, we will use swarm quadrotor to enhance surveillance patrol to be more effective.

ACKNOWLEDGEMENT

This work is supported by Grant of National Innovation System Intensive Research No. 06/M/Kp/I/2012 by the Ministry of Research and Technology, Republic of Indonesia.

REFERENCES

[1] J. Dufrene, W.R., “Mobile military security with concentration on unmanned aerial vehicles,” in Digital Avionics Systems Conference, 2005. DASC 2005. The 24th, vol. 2, 2005, pp. 8 pp. Vol. 2.

[2] A. Pahsa, P. Kaya, G. Alat, and B. Baykal, “Integrating navigation amp; surveillance of unmanned air vehicles into the civilian national airspaces by using ads-b applications,” in Integrated Communications, Navigation and Surveilance Conference (ICNS), 2011, 2011, pp. J7–1–J7–7 H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[3] Shafique, S.; Sheikh, N.M., "Simple algorithm for detection of elliptical objects in remotely sensed images for UAV applications," *Applied Sciences and Technology (IBCAST), 2009 6th International Bhurban Conference on* , vol., no., pp.258,264, 19-22 Jan. 2009.

[4] Teutsch, M.; Kruger, W., "Detection, Segmentation, and Tracking of Moving Objects in UAV Videos," *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on* , vol., no., pp.313,318, 18-21 Sept. 2012.

[5] Ibrahim, A.W.N.; Pang Wee Ching; Seet, G.L.G.; Lau, W.S.M.; Czajewski, W., "Moving Objects Detection and Tracking Framework for UAV-based Surveillance," *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on* , vol., no., pp.456,461, 14-17 Nov. 2010.

[6] D. C. Jakob Engel, J urgen Sturm, “Accurate figure flying with a quadrocopter using onboard visual and inertial sensing,”http://vision.in.tum.de/media/spezial/bib/engel12vic-omor.pdf

[7] M. Alvissalim, B. Zaman, Z. Hafizh, M. Ma’sum, G. Jati, W. Jatmiko, and P. Mursanto, “Swarm quadrotor robots for telecommunication network coverage area expansion in disaster area,” in SICE Annual Conference (SICE), 2012 Proceedings of, 2012, pp. 2256–2261.

[8] J. Takahashi, K. Sekiyama, and T. Fukuda, “Self-deployment algorithm for mobile sensor network based on connection priority criteria with obstacle avoidance,” in Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on, 2007, pp. 1434–143

[9] Ar Drone 2.0 Spesification, http://ardrone2.parrot.com/AR.Drone-2/specifications/ accessed on June 12th 2013.

[10] S. Cousins, “Exponential Growth of ROS,” IEEE Robotics and Automa-tion Magazine, no. March, pp. 19–20, 2011

[11] AR.Drone Autonomy Research Group, “AR.Drone autonomy repository, ”https://github.com/AutonomyLab/AR.Drone autonomy. accessed on June 12th 2013.

[12] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, December 2001, pp. 511–518