The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Robust visual lock-on and simultaneous localization for an unmanned aerial vehicle

Jihong Min, Yekeun Jeong, In So Kweon

***Abstract*— We present a method for simultaneously locking on to a ground target and estimating the position of an unmanned aerial vehicle (UAV) under countermeasure (CM) conditions, where sensors are prevented from successfully tracking a target. Owing to the limited payload and power of the UAVs, we employ a monocular camera and a global positioning system (GPS) to carry out vision-based simultaneous localization and mapping (SLAM) using both an unscented Kalman filter and a Kalman filter. Since this approach estimates the state of the UAV and the location of the target, we can estimate the position of the target in the image, even in the presence of CMs.**

Our experiments show that the proposed method successfully locks on to the target and estimates the state of the UAV.

I. Introduction

In recent years, unmanned aerial vehicles (UAVs) have been extensively used in various applications including airborne surveillance and modern warfare, as well as in academic research. These vehicles can perform various tasks in areas that are otherwise inaccessible, including extremely dangerous environments.

In modern warfare, a UAV is used for conducting a variety of missions, including reconnaissance and invasion. One of the most important missions of a UAV is surface target bombing to destroy vital enemy sites. In order to accomplish such missions, the UAV should be able to track the target robustly, even in the presence of CMs.

To date, extensive research has been conducted on tracking a ground target with a UAV; tracking with fighter jets and missiles has been extensively studied and has shown good results. In these approaches, lasers, radars, vision systems, fused sensors, etc. have been widely used [1][2][3]. However, the CM conditions have not been addressed.

A variety of CMs such as laser and broadband jammers, smokes/aerosols, decoys, and directed infrared energy CMs can be deployed against UAVs. These CMs usually cause the sensors to miss the target or detect a number of undesired targets, and they may also jam or spoof the sensor signals.

In this study, we considered two types of CMs, smoke and jamming. Smoke may occlude the target from view; other CMs that cause visual interference can be treated as smoke. Jamming interferes with the operation of GPS sensors.

Because a UAV cannot be equipped with numerous sensors to track a target under CM conditions owing to its limited payload and power requirements, we choose a monocular camera and a global positioning system (GPS).

The authors are with the Electrical Engineering Department, Korea Advanced Institute of Technology (KAIST), 335 Gwahak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea. (e-mail: jhmin@rcv.kaist.ac.kr, ykjeong@rcv.kaist.ac.kr, iskweon@ee.kaist.ac.kr.)

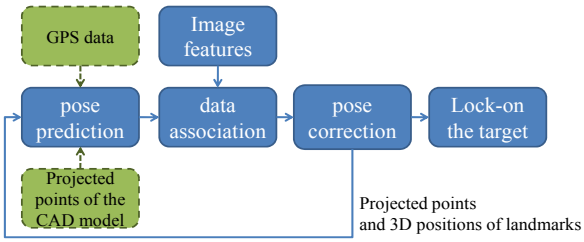


Figure 1. Overall procedure of our method. Dashed boxes represent data that are not always available

Since general vision-based tracking approaches are likely to fail when the target is partially or totally occluded, we apply a vision-based simultaneous localization and mapping (SLAM) algorithm to address this issue. Owing to its adaptive characteristics, this approach is particularly suited to our needs.

The vision-based SLAM [4][5][6] approach uses probabilistic filtering approaches to obtain geometric information about an entire image, which also includes regions other than the target. This approach allows us to simultaneously create a map and estimate the state of the UAV, even under severe CMs.

Several previous works have used approaches similar to our approach. Amidi et al. [7] proposed a visual odometer for an autonomous helicopter flight. This device can visually lock on to ground objects and estimate the position of the helicopter. However, they did not consider cases where objects were occluded. Deneault et al. [8] proposed an approach for tracking ground targets with a single camera, a GPS, and an inertial measurement unit (IMU). Their approach is similar to our approach, where the SLAM is used with an unscented Kalman filter (UKF) and a Kalman filter (KF). However, they did not consider cases where GPS data or IMU data was lost.

A. Approach

Fig. 1 shows the overall procedure of our method. The pose of the UAV is predicted from the geometric information, which is derived from the features in the images and updated by the KF when the GPS data or projected points of the target are available. The pose is then corrected by the UKF. For computational efficiency and filter consistency, we select and match the features using an active matching method [4] and then perform a statistical test.

Since the SLAM approach estimates the positions of the

UAV and the target, we can compute the relative distance and the angle between the UAV and the target and then estimate the position of the target in the image, even in the presence of CMs. The following sections explain each step in detail.

B. Assumptions

Before explaining the approach, we note the following assumptions made.

- ✓ We already know the computer-aided design(CAD) model of the target, which means actual distances between the corner points of the target are known.
- ✓ GPS data and the global pose of the CAD model with some errors are available for the first n views and periodically transmitted to the UAV when there are no CMs.
- ✓ We already know the intrinsic parameters of the camera.
- ✓ We regard jammed GPS data to be equivalent to their absence.
- ✓ The UAV does not change its heading direction very quickly.

II. Probabilistic Filtering Scheme

A. State Vector

The state vector $X(k)$ at time k is composed of the UAV state and positions of the features on both the target and the non-target regions; the latter denotes all regions except the target in the image.

$$X(k) = [A(k) \quad T(k) \quad F(k)]^T \tag{1}$$

where $A(k)$ denotes the state of the UAV, and $T(k)$ and $F(k)$ denote a set of n features' 3D positions in the target and the non-target regions, respectively.

$$A(k) = [r(k) \quad \Psi(k)]^T \tag{2}$$

$$T(k) = [t_1(k) \quad \cdots \quad t_m(k)]^T \tag{3}$$

$$F(k) = [f_1(k) \quad \cdots \quad f_n(k)]^T \tag{4}$$

where $A(k)$ contains a position vector $r(k)$ and a rotation vector $\Psi(k)$. $t_i(k)$ is the 3D position of the i th corner point of the CAD model, and $f_i(k)$ is the 3D position of the i th feature. In this paper, we denote $t_i(k)$ and $f_i(k)$ as landmarks.

B. Pose Prediction

The vision literature extensively documents methods to determine the pose of a camera when the 3D positions of features and their projected positions in the image are known. In this study, we follow the approach of the three-point algorithm [9] to estimate the rotation and translation vectors of the camera when three correspondences are given between the image coordinates and their 3D locations in the state vector.

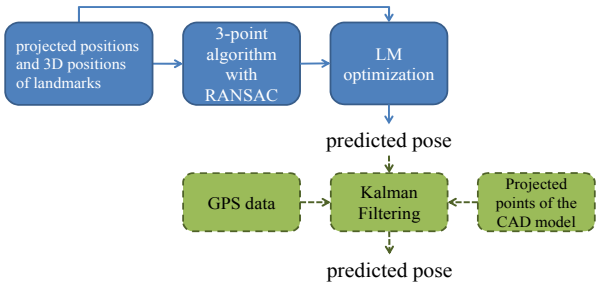


Figure 2. Procedure for UAV pose prediction

In addition, we combine the random sample consensus (RANSAC) algorithm [10] with the three-point algorithm for robust estimation of the model parameters in a manner similar to that used for the computation of geometric relations [11]. Fig. 2 shows the procedure for the UAV pose prediction.

After determining an initial estimate of the pose vector, we refine the estimate by minimizing the reprojection error using the Levenberg-Marquardt (LM) algorithm [12]. The refined pose vector is used as a predicted value of the UAV pose.

If the GPS data and the projected corner points of the CAD model are available, we can update the predicted pose using the Kalman filter as follows:

$$\tilde{X}(k) = \bar{X}(k) + K(k)(A_g(k) - A_f(k)) \tag{5}$$

$$\tilde{P}(k) = (I - K(k)H(k))\bar{P}(k) \tag{6}$$

$$K(k) = \bar{P}(k)H(k)^T (H(k)\bar{P}(k)H(k)^T + R(k))^{-1} \tag{7}$$

$$A_g(k) = [r_g(k) \quad \Psi_{cad}(k)]^T \tag{8}$$

$$A_f(k) = [r_f(k) \quad \Psi_f(k)]^T \tag{9}$$

$$\bar{X}(k) = [A_f(k) \quad T(k) \quad F(k)]^T \tag{10}$$

$$H(k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \tag{11}$$

where $A_g(k)$ is a pose vector containing the position vector $r_g(k)$ received from a GPS at time k and a rotation vector $\Psi_{cad}(k)$ determined from the projected points of the CAD model. $A_f(k)$ contains the position vector $r_f(k)$ and the rotation vector $\Psi_f(k)$ derived from the LM optimization.

$\bar{X}(k)$ is a predicted state vector that contains the predicted pose vector $A_f(k)$ and the position vectors $T(k)$ and $F(k)$ of the features in the target and the non-target regions, respectively. The predicted covariance matrix $\bar{P}(k)$ at time k is the same as the previous covariance matrix.

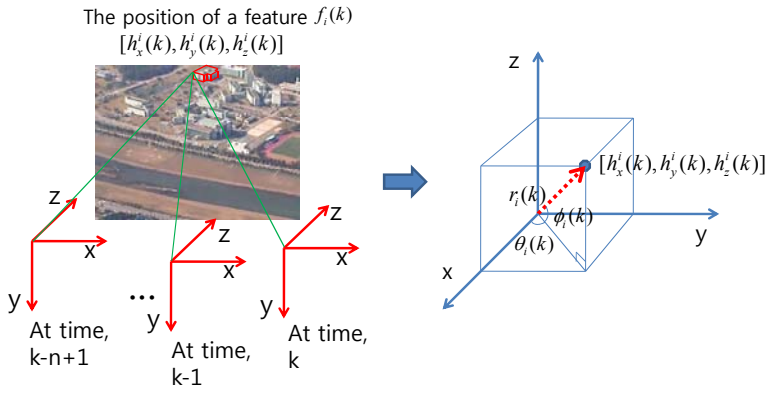


Figure 3. Initialization of features and measurement model

$$\bar{P}(k) = P(k-1) \tag{12}$$

However, if only the GPS data is available, we use equation (13) instead of equation (5) to update the state vector. Although the orientation cannot be updated, we can estimate the orientation by the pose correction step.

$$\tilde{X}(k) = \bar{X}(k) + K(k)(r_g(k) - r_f(k)) \tag{13}$$

C. Measurement Model

A monocular camera cannot give the 3D position of a feature using a single image. Hence, Davison et al. [4] proposed an initialization process involving the use of several images for determining the depth of the feature point by employing a particle-filter-type approach under the assumption that the feature is located close to the camera. However, most features are located far away from the camera in our case, and thus, this approach is infeasible.

In this study, we use several images to determine the 3D position of a feature by *n*-view triangulation. Fig. 3 shows the process of *n*-view triangulation [11] and the measurement model of the feature. However, the position of the feature obtained from *n*-view triangulation is not correct because the camera model is a weak perspective model.

Therefore, we use a monocular camera as a range sensor and measure the bearing and the range of the feature, because the ray direction of the feature offers better reliability than the position of the feature. After measuring the bearing and the range of the feature, we determine the position and the uncertainty of the feature by unscented transformation (UT) [13]. Equation (14) represents the measurement equation.

$z_i(k)$ is a measurement of the feature $f_i(k)$ at time *k* from the positions of the camera between time $k - n + 1$ ($X(k-n+1)$) and time k ($X(k)$). $v(k)$ is the measurement noise with zero mean uncorrelated Gaussian noise. $\left[h_x^i(k) \quad h_y^i(k) \quad h_z^i(k)\right]^T$ is the 3D position of the feature $f_i(k)$. $r_i(k)$ is the distance between a UAV and a feature. $\theta_i(k)$ and $\phi_i(k)$ are the azimuth and elevation angles from the position of the UAV to the feature, respectively.

$$z_i(k) = h(f_i(k), X(k-n+1), \ldots, X(k)) + v(k)$$
$$= \begin{bmatrix} \sqrt{h_x^i(k)^2 + h_y^i(k)^2 + h_z^i(k)^2} \\ \arctan\left(\dfrac{h_y^i(k)}{h_x^i(k)}\right) \\ \arctan\left(\dfrac{h_z^i(k)}{\sqrt{h_x^i(k)^2 + h_y^i(k)^2}}\right) \end{bmatrix} + v(k) = \begin{bmatrix} r_i(k) \\ \theta_i(k) \\ \phi_i(k) \end{bmatrix} + v(k) \tag{14}$$

D. Pose Correction

The UKF is used to update the poses of both the UAV and the landmarks. Instead of approximating the nonlinear measurement function h by a Taylor series expansion in the extended Kalman filter (EKF), the UKF linearizes the nonlinear function through UT and predicts the mean and covariance to the second order. In contrast, the EKF can only approximate up to the first order [14].

The updated equations are

$$X(k) = \tilde{X}(k) + K(k)(z(k) - \hat{Z}(k)) \tag{15}$$
$$P(k) = \tilde{P}(k) - K(k)S(k)K(k)^T \tag{16}$$

where

$$K(k) = P^{xz}(k)S(k)^{-1} \tag{17}$$
$$S(k) = \sum_{i=0}^{2n} \omega_c^{[i]}(Z^{[i]}(k) - \hat{Z}(k))(Z^{[i]}(k) - \hat{Z}(k))^T + Q(k) \tag{18}$$
$$P^{xz}(k) = \sum_{i=0}^{2n} \omega_c^{[i]}(\chi^{[i]}(k) - \tilde{X}(k))(Z^{[i]}(k) - \hat{Z}(k))^T \tag{19}$$

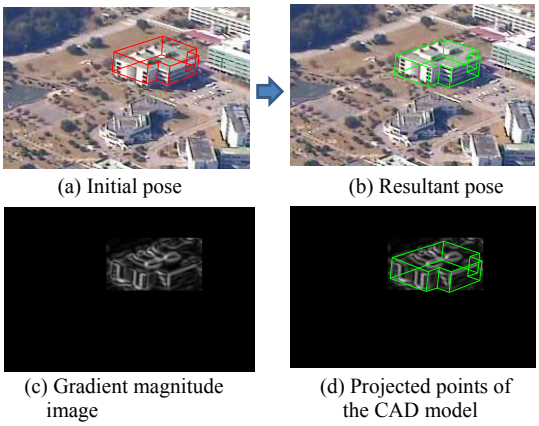


Figure 4. Results of CAD model matching

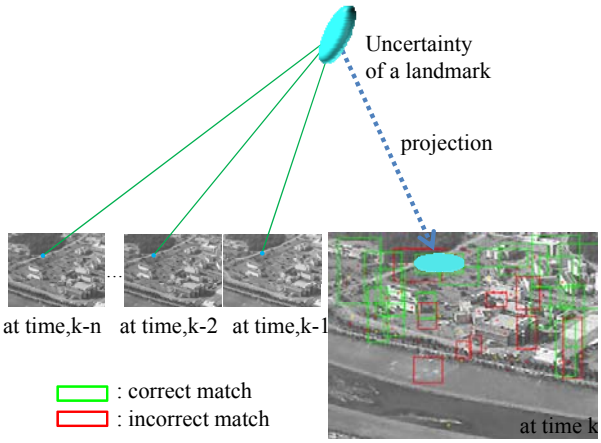


Figure 5. Active matching process and results of data association

$\hat{Z}(k)$ is a weighted sample mean of the sigma points that are passed through the measurement function h. $\omega_c^{[i]}$ is the weight of the i th sigma point, and $Z^{[i]}(k)$ is the propagated sigma point. $Q(k)$ is the zero mean uncorrelated Gaussian noise. For a complete explanation of the parameters of the UKF, refer to [13].

III. Feature Management

A. Initialization and Matching

We use SURF [17] as image features and track them in subsequent frames by image-to-image matching. Since we assume that the initial poses of the UAV are known, we can easily determine the initial 3D positions of the features. In the following frames, we can also determine the 3D positions of new features even if the GPS data is not available, because the pose of the UAV can be estimated by the filtering scheme.

If new features are tracked during some consecutive frames (in this paper, we use three frames), we measure the range and the bearing of the features and augment them to the state vector by using UT.

In the target case, since we assume that the CAD model and the initial global pose of the target with errors are known, we can predict the projected points of the CAD model in the image. As shown in Fig. 4(a), because of the errors in the UAV and the CAD model poses, the projected points of the CAD model in the image are not consistent with the measurements. In order to determine the correct projected points of the CAD model, we use the RANSAC algorithm and find the rotation and the translation vectors that maximize the energy, as shown in equation (20). Random values of the rotation and the translation vectors are limited to the uncertainties of each value.

$$E = \max_{r,t} \sum_{(x,y)\in(x_{cad},y_{cad})} I_{mag}(x,y) \qquad (20)$$

$$where\ [x_{cad}, y_{cad}, 1]^T = \lambda K \cdot RT(r,t) \cdot [X_{cad}, Y_{cad}, Z_{cad}, 1]^T$$

$I_{mag}(x,y)$ is the gradient magnitude value of the point (x,y) in the image, and $[x_{cad}, y_{cad}]$ is one of the projected points of the CAD model. Some example images are shown in Figs. 4(c) and (d). K is the intrinsic parameter of the camera, and $RT(r,t)$ is the extrinsic matrix determined by the rotation vector r and the translation vector t. $[X_{cad}, Y_{cad}, Z_{cad}]$ is one of the global positions of the CAD model. In order to reduce false matching, we only use some parts of the magnitude image, which are determined by the projected points of the CAD model with certain margins.

The projected corner points of the CAD model are tested by following the data association method. Then, if data is available, we use the rotation result in the filtering scheme.

B. Data Association

Associating the observations with incorrect landmarks in the filter can make the system inconsistent. This usually happens when matches of features are incorrect or features are on a moving object.

First, we use matching constraints of the active matching method [4] to find correct matches between landmarks in the state vector and observed features from the current image. In Fig. 5, the searching region is obtained by projecting the uncertainty of a landmark onto the image. In this manner, we can reduce the computational time required for finding a correct match.

If a landmark is on a moving object, the probability of the candidate feature falling in the searching region may be low, and the feature may be filtered out.

After active matching, we perform a statistical test using the filtered features. If we know the true solution for the state, we can perform a statistical test, i.e., a normalized estimation error squared (NEES) test, for filter consistency. Since the true solution for the state is not available, we perform another statistical test, i.e., a normalized innovation squared (NIS) test [14].

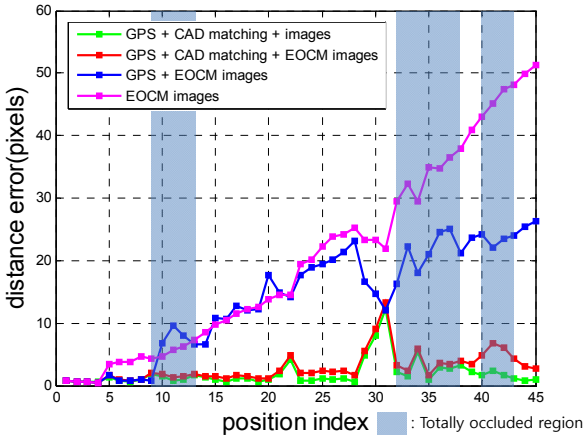


Figure 6. Lock-on results obtained with different combinations of sensor data

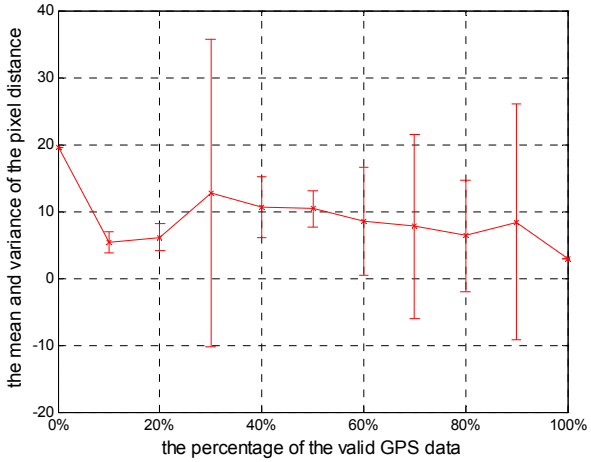


Figure 7. Monte Carlo experiments of the lock-on performance with randomly selected GPS data

$$v(k)^T S(k)^{-1} v(k) \le \chi^2_{r,1-\alpha} \tag{21}$$

where $v(k) = z(k) - \hat{Z}(k)$ is the innovation, $S(k)$ is given by equation (18), $\chi^2_{r,1-\alpha}$ is a threshold obtained from a χ^2 distribution with $r = \dim(v(k))$, and α is the desired significance level (in our case, 0.05).

Fig. 5 shows the process and the results of data association. The red rectangular region represents a match between a landmark and one of the features in the searching region that satisfies the NIS test. The green region shows that no features are detected that satisfy the NIS test.

C. Deletion

The state vector increases with an increase in the number of landmarks, and in turn, the computational time required for updating the map with *n* features increases drastically. Since the UAV moves forward at a high speed, most landmarks are rarely re-observed. Therefore, it is inefficient to keep all landmarks in the state vector. We followed the approach of Davison et al. [4], who removed the landmarks from both the state vector and the covariance matrix that were not observed in a predefined period. In our experiments, we use eight frames and do not remove the landmarks of the CAD model.

IV. Lock-on The Target

Given the estimates of the 3D positions of the UAV and the target, the position $(u_i(k), v_i(k))$ at which the CAD model is expected to be found in the image is determined as follows.

$$\begin{bmatrix} u_i(k) \\ v_i(k) \end{bmatrix} = CR(\Psi(k))(t_i(k) - r(k)) \tag{22}$$

where C is an intrinsic parameter of a camera, $R(\Psi(k))$ is a rotation matrix, and $\Psi(k)$ is a rotation vector of the UAV. $t_i(k)$ is the 3D position of the i th corner point of the CAD model, and $r(k)$ is the 3D position of the UAV. Using a set of projected features in the target, we can lock on to the target in the image.

V. Experiments

A. Aerial Space

1) Experimental Setup

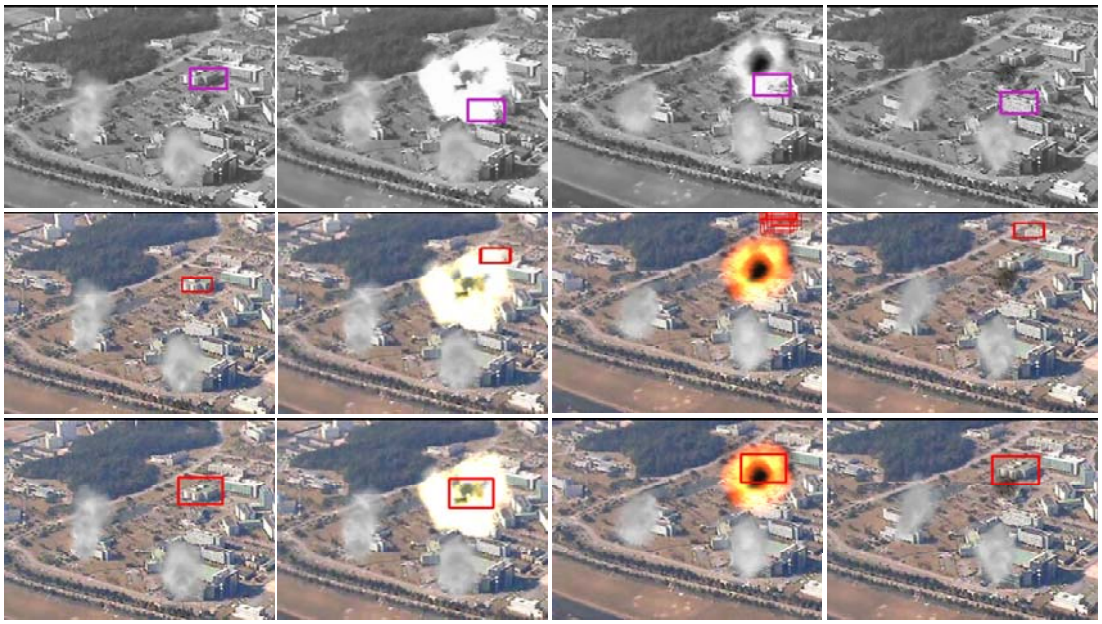
We performed experiments with a commercial GPS receiver and a monocular camera mounted on a helicopter. The helicopter moved toward the target from approximately 2 km away to 1 km away. Images with a resolution of 320 × 240 pixels were captured at a rate of 30 Hz, and the position data was received from the GPS at a rate of 1 Hz.

2) Lock-on Results

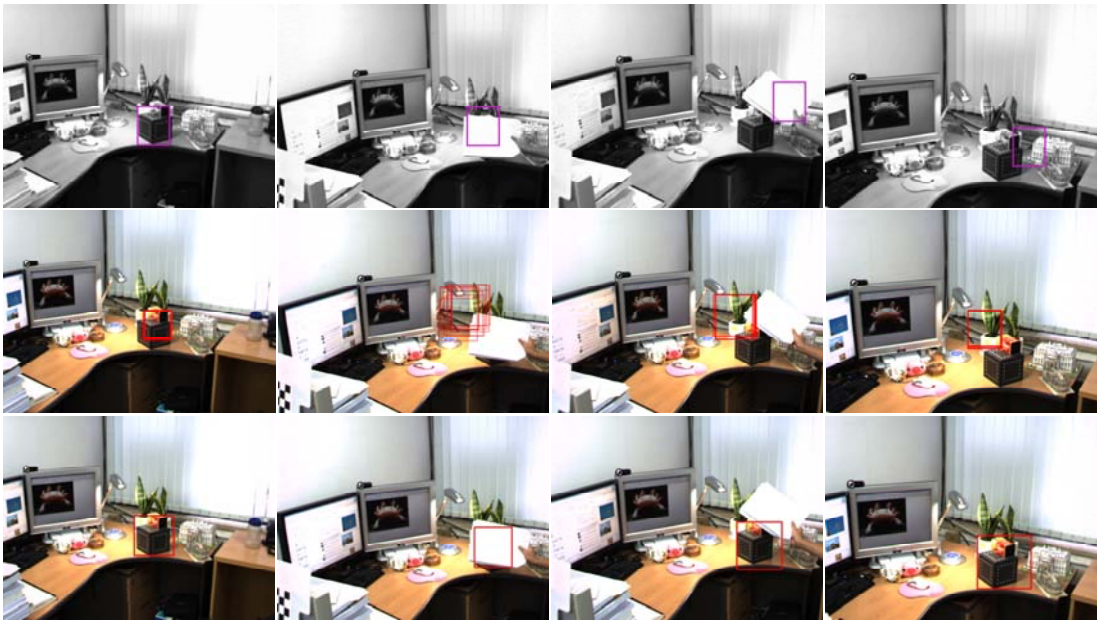
A fair evaluation of our approach is difficult given the absence of data captured under the real conditions of CMs. Therefore, we simulated the CM conditions by using a graphical tool and discarding some of the GPS data.

First, lock-on results with different combinations of sensor data are shown in Fig. 6. We evaluate the lock-on performance with the pixel distance between the original image center and the estimated image center of the target. For the first several frames, all sensor data is used for initialization. The lock-on error occurs even if CAD matching is used because the UAV still shows a position error after pose correction.

To evaluate the effect of the GPS data on the lock-on, we perform Monte Carlo experiments with the randomly eliminated GPS data. The mean and variance of the pixel distance between the original image center and the estimated image center of the target are shown in Fig. 7. Since the GPS data gives more accurate location information than visual sensors, we can obtain better lock-on performance.



(a) Aerial images



(b) Indoor images

Figure 8. Comparison results of the lock-on performance with aerial images and indoor images.
First rows: MIL tracking [15]; second rows: particle-filter-based tracking [16], last rows: proposed method

We compared our approach with the state-of-the-art tracking algorithms, the online multiple instance learning (MIL) algorithm [15] and the particle-filter-based tracking algorithm [16], both of which deal with the occlusion problem.

For a fairer comparison, we only use visual data except for the initialization step.

Fig. 8(a) shows the comparison results of the lock-on performance with aerial images. The first, second, and last rows show the results of MIL tracking, particle-filter-based tracking, and our approach, respectively, obtained using only visual data.

The lock-on performance of our approach with only visual information decreases gradually when the target is very far away from the camera because localization with only the camera is not accurate. However, it is more robust than the other vision-based algorithms under long-term occlusion.

3) Localization Results

Since we cannot precisely evaluate the localization accuracy of our approach because of the GPS errors, we use the position distance between the GPS data and the estimated position as a localization error. Since it is natural that more GPS data gives smaller localization errors, we created a scenario where several GPS data points are valid in the first,

middle, and last frames. Fig. 9 shows this scenario and the localization results. Even if the GPS data is not valid, we can estimate the location of the UAV with some errors by using the visual data.

B. Indoor Space

We performed experiments with a hand-held monocular camera in a room. The camera moved toward the target from 4 m away to 1 m away while capturing images at a resolution of 640 × 480 pixels and a rate of 10 Hz; the target was frequently occluded. We used only visual information because the GPS cannot be used in indoor environments. However, for initialization, we captured some images with known 3D positions.

As in the case of the aerial experiments, we compared our approach with other algorithms. Fig. 8(b) shows the results of this comparison.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we present a robust vision-based SLAM approach for simultaneously locking on to a ground target and estimating the position of a UAV. By using visual information about both the non-target region and the target model, we can lock on to the target and localize a UAV under CM conditions.

In addition, we combine the GPS data with the geometric information derived from the features in an image through the use of a KF and a UKF. This scheme helps us to lock on to the target and localize the UAV more robustly under CM conditions.

However, when the target is very far away from the UAV, the visual information is not sufficient to estimate the location of the UAV and lock on to the target.

Future work will deal with this situation and also include large map management for computational efficiency; further, experiments will be performed under various CM conditions.

VII. ACKNOWLEDGEMENTS

This research was supported by the Defense Acquisition Program Administration and the Agency for Defense Development, Korea through the Image Information Research Center at Korea Advanced Institute of Science & Technology under the contract UD100006CD.

We would like to express our sincere gratitude to the reviewers of this paper for their valuable comments and recommendations.

REFERENCES

[1] Y. Kang, D. S. Caveney, J. K. Hedrick. “Real-time Obstacle Map Building with Target Tracking”, Journal of Aerospace Computing, Information, and Communication (1542-9423),2008,vol. 5 no. 5

[2] P. Theodorakopoulos, S. Lacroix, “A Strategy for Tracking a Ground Target with a UAV”,IROS 2008

[3] M. Ridley, E. Nettleton, A. Goktogan, G. M. Brooker, S. Sukkarieh, H. Durrant-Whyte: “Decentralised Ground Target Tracking with Heterogeneous Sensing Nodes on Multiple UAVs”. Information processing in sensor networks 2003, pp. 545-565

[4] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, “MonoSLAM: Real-Time Single Camera SLAM”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 29(6), pp. 1052-1067, 2007

[5] E. Eade, T. Drummond, “Scalable Monocular SLAM”, IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 469?476, 2006.

[6] D. Chekhlov, M. Pupilli, W. Mayol, A. Calway, “Robust Real-Time Visual SLAM Using Scale Prediction and Exemplar Based Feature Description”, IEEE International Conference on Computer Vision and Pattern Recognition, 2007.

[7] O. Amidi, T. Kanade and K. Fujita, “A Visual Odometer for Autonomous Helicopter Flight”, Intelligent autonomous systems 1998, pp.123-130

[8] D. Deneault, D. Schinstock, C. Lewis, “Tracking Ground Targets with Measurements Obtained from a Single Monocular Camera Mounted on an Unmanned Aerial Vehicle”, IEEE International Conference on Robotics and Automation, ICRA 2008. pp. 65-72

[9] D. Nister, “A Minimal Solution to the Generalised 3-Point Pose Problem”, IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2004

[10] M. A. Fischler, R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, in ECCV’96, 1996, pp. 683-695

[11] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN 0-521-62304-9, 2000. pp. 116-131, 310-324

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes in C++*,Cambridge Univ. Press,ISBN 0-521-75033-4,2002

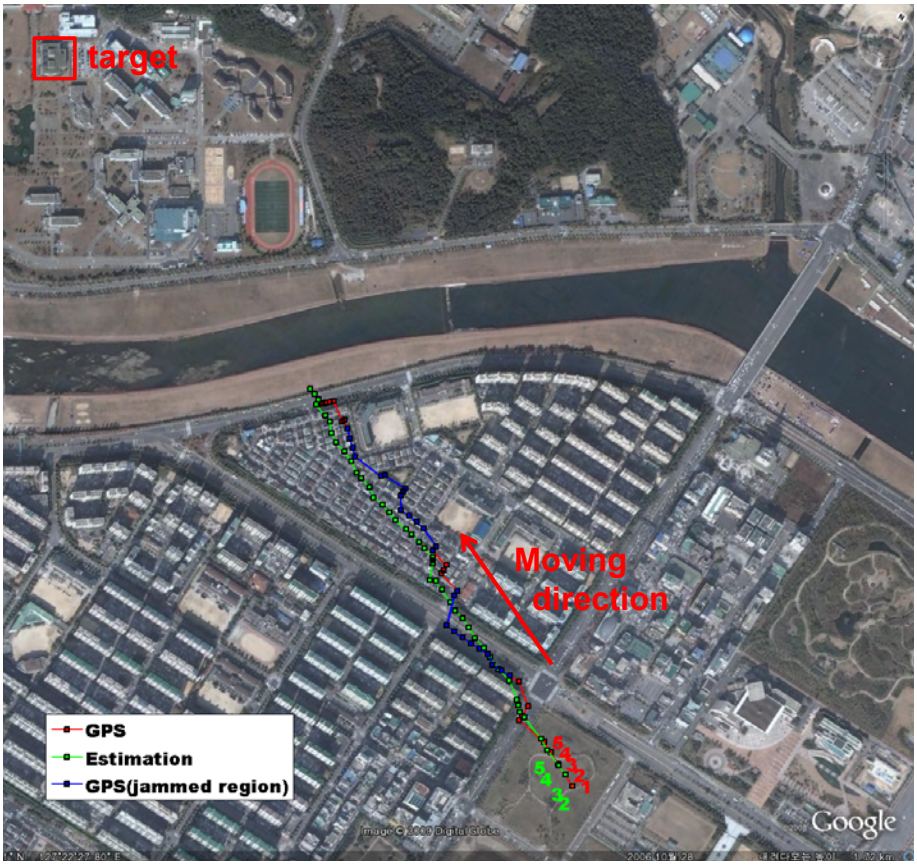
[13] S.J.Julier, J.K.Uhlmann, H.F. Durrant-Whyte. “A New Approach for Filtering Nonlinear Systems”, In Proceedings of the American Control Conference, pp. 1628-1632,1995

[14] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardos, J. Neira, “Robocentric Map Joining: Improving the Consistency of EKF-SLAM”, robotics and autonomous systems, Volume 55, Issue 1, 31 January 2007, pp. 21-29

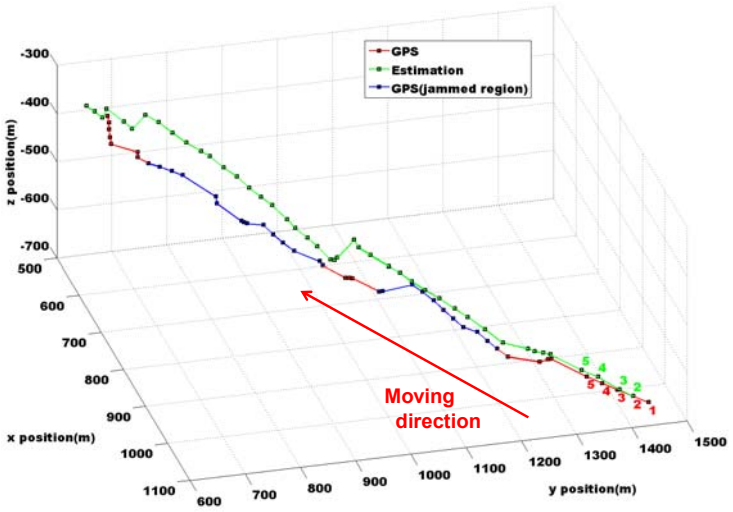
[15] B. Babenko, M. Yang, S. Belongie, “Visual Tracking with Online Multiple Instance Learning”, IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2009

[16] J. Kim, C. Park, I.S. Kweon, “Visual Tracking for Non-Rigid Objects using Rao-Blackwellized Particle Filter”, IEEE International Conference on Robotics and Automation, ICRA 2010

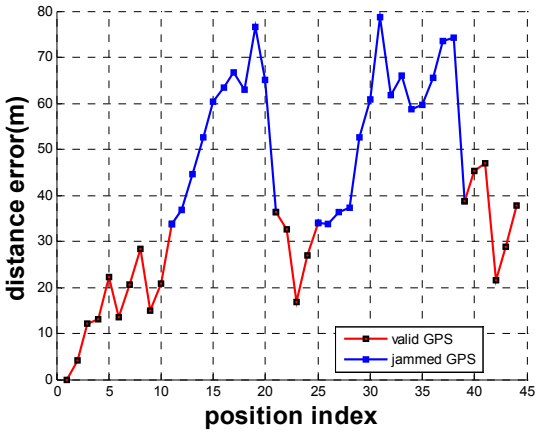
[17] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, “SURF: Speeded Up Robust Features”, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2008



(a) Localization results and GPS on a Google map



(b) 3D trajectories of localization results and GPS



(c) Distance errors between GPS and localization results in Euclidian distance

Figure 9. Localization results of a UAV