Obstacle Detection for Low Flying UAS Using Monocular Camera

F. Zhang[1], R. Goubran[2], P. Straznicky[3]
[1]Graduate Student, Department of Systems and Computer Engineering
Carleton University, Canada. fzhang@sce.carleton.ca
[2]Professor and Dean of Engineering, Faculty of Engineering and Design, Carleton University, Canada
SMIEEE. goubran@sce.carleton.ca
[3]Professor, Department of Mechanical and Aerospace Engineering
Carleton University, Canada. pstrazni@mae.carleton.ca

***Abstract* — This paper describes an obstacle detection algorithm for low flying unmanned aircraft system (UAS) using an inertial aided inverse depth Extended Kalman Filter (EKF) framework. The EKF framework fuses inertial measurements with monocular image sensor measurements to estimate the positions of a number of landmarks as well as the position and orientation of the UAS. A high resolution sparse terrain elevation map and UAS trajectory can then be computed from the filter state vector. An inverse depth parameterization is used to describe the positions of the landmarks so that features at all ranges can be tracked by the filter. The integration of inertial sensor increases the robustness of the algorithm under rough flying conditions. A test flight is conducted to test the algorithm in a realistic scenario. The result shows that the algorithm produces accurate terrain elevation model, and is capable of generating accurate high resolution terrain elevation map when image sensor with high resolution and dynamic range is used.**

Keywords – Obstacle Detection, Range Estimate, UAS, Motion Stereo

I. Introduction

Generating accurate high resolution elevation maps of the terrain to indicate hazard features is an important issue for low flying Unmanned Aircraft Systems (UAS). An autonomous system builds a model of the surrounding environment through range measurement making use of various technologies such as laser range finder [1], ultrasonic signals [2], image sensors [3] - [4], or 3D flash LIDAR [5]. However, laser, or ultrasonic sensors only capture range measurement at a point from a given position and orientation, and require scanning to obtain a complete range image. 3D flash LIDAR is able to simultaneously capture depth measurements across its entire field of view (FOV), but its high cost has limited its use in commercial applications. With recent advances of computer vision technology, cameras as passive sensors provide a low cost, light weight, yet feasible alternative.

In this paper, we describe an inertial aided inverse depth Extended Kalman Filter framework for sparse terrain map generation from monocular image sequence. The Extended Kalman Filter allows for incrementally integrating new measurements with existing measurements. Inverse depth parameterization was adopted so that the algorithm is robust in processing features at all distances. By fusing in inertial measurements, the algorithm is capable of handling rough flying conditions, such as excessive vibration, or sudden trajectory change of the aircraft.

II. Inertial Aided Inverse Depth Extended Kalman Filter for Monocular sequence

A. *Camera Centric Inverse Depth Parameterization*

The standard way of describing a feature's position is to use the Euclidean XYZ parameterization. In practical outdoor range estimation problem, the algorithm must deal with features located at near infinity. Inverse depth parameterization overcomes this problem by representing range d in its inverse form $\rho = 1/d$. In addition, features at infinity contribute in estimating the camera rotational motion even though they offer little information on camera translational motion. Furthermore, inverse depth parameterization allows us to initialize features into the EKF framework before it is safely triangulated.

The inverse depth parameterization used in this work was first introduced in [6]. All features and camera positions are referred to a world reference frame. When used in an Extended Kalman Filter framework, the system suffers decreasing linearity when camera is moving away from the world origin. A modified method which uses the camera center as the origin was proposed in [7]. Our work has adopted the camera centered approach with some modifications to integrate the inertial measurements.

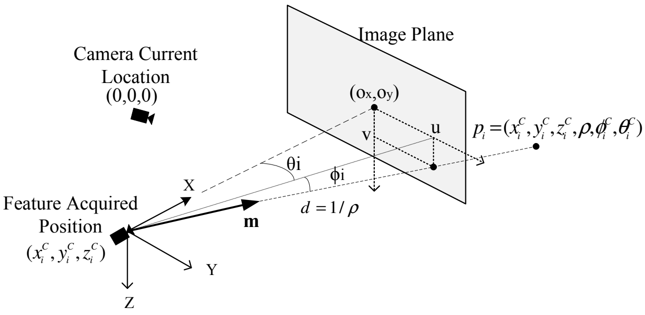

Figure 1 Inverse Parameterization

This work has been made possible by the funding from NSERC, and is supported by Sander Geophysics Ltd. who has provided the test flight, all hardware equipments, and ground truth data

F. Zhang is with Department of Systems and Computer Engineering, Carleton University, as well as Sander Geophysics Ltd.

A scene point $\boldsymbol{p}_i^C$ can be defined by 6 parameters, with the superscript C representing a camera reference frame (Figure 1).

$$\boldsymbol{p}_i^C = [x_i^C \ \ y_i^C \ \ z_i^C \ \rho_i \ \ \phi_i^C \ \ \theta_i^C\ ]^T \tag{1}$$

The first three parameters $[x_i^C, y_i^C, z_i^C]$ represent the initialization position where the feature is first observed. ρ_i is the inverse distance from the initialization position to the feature. The elevation-azimuth pair $[\phi_i^C, \theta_i^C]$ encodes a unit vector pointing from initialization point to the feature. The vector is given by

$$\boldsymbol{m}(\phi_i^C, \theta_i^C) = \begin{bmatrix} cos\phi_i^C cos\theta_i^C \\ cos\phi_i^C sin\theta_i^C \\ \sin\phi_i^C \end{bmatrix} \tag{2}$$

B. *Full State Vector*

The EKF state vector is defined as

$$\boldsymbol{x} = [\boldsymbol{OX}_{\boldsymbol{W}}^{\boldsymbol{C}} \quad \boldsymbol{c}^{\boldsymbol{C}} \quad \boldsymbol{r}^{\boldsymbol{C}} \quad \boldsymbol{p}_{\boldsymbol{1}}^{\boldsymbol{C}} \quad \boldsymbol{p}_{\boldsymbol{2}}^{\boldsymbol{C}} \quad ...]^{\boldsymbol{T}} \tag{3}$$

where $\boldsymbol{OX}_{\boldsymbol{W}}^{\boldsymbol{C}} = \left[O_x^C \quad O_y^C \quad O_z^C \ W_x^C \quad W_y^C \quad W_z^C\right]^T$ contains translation parameters $O_{x,y,z}^C$ and rotation parameters $W_{x,y,z}^C$ to transform the camera reference frame to the world reference frame. $[\boldsymbol{c}^{\boldsymbol{C}}, \boldsymbol{r}^{\boldsymbol{C}}]^{\boldsymbol{T}}$ represents the camera translation and rotation motion frame by frame in Euclidean coordinates, and $\boldsymbol{p}_{\boldsymbol{i}}^{\boldsymbol{C}}$ contains the feature parameters as described in the previous section.

C. *Prediction*

For a prediction step at time k, the world frame and features parameters are kept unchanged from time $k-1$. The camera parameters are updated using the new inertial measurements: velocity $\boldsymbol{v}^{\boldsymbol{C}}$, acceleration $\boldsymbol{a}^{\boldsymbol{C}}$, and rate of change in roll/pitch/yaw $\boldsymbol{w}^{\boldsymbol{C}}$. The camera motion parameters at time k are then

$$\begin{gathered} \boldsymbol{c}_{\boldsymbol{k}}^{\boldsymbol{C}} = \boldsymbol{v}^{\boldsymbol{C}}\Delta t + \frac{1}{2}\boldsymbol{a}^{\boldsymbol{C}}\Delta t^2 \\ \boldsymbol{r}_{\boldsymbol{k}}^{\boldsymbol{C}} = \boldsymbol{r}_{\boldsymbol{k-1}}^{\boldsymbol{C}} + \boldsymbol{w}^{\boldsymbol{C}} \end{gathered} \tag{4}$$

D. *Measurement Model*

Each observed feature is related to the camera motion through the measurement model (Figure 2). This relationship enables a correction on the camera motion and features parameters based on the features' location observed in the image.

For a feature $\boldsymbol{p}_{\boldsymbol{i}}^{\boldsymbol{C}}$, the vector $\boldsymbol{h}^{\boldsymbol{R}}$ pointing from the predicted camera location to the feature initialization position is

$$\boldsymbol{h}_{\boldsymbol{k}}^{\boldsymbol{R}} = \begin{bmatrix} x_i^C \\ y_i^C \\ z_i^C \end{bmatrix}_k - \begin{bmatrix} c_x^C \\ c_y^C \\ c_z^C \end{bmatrix}_k \tag{5}$$

The normalized vector pointing from the predicted camera position to the feature at time k is then

$$\boldsymbol{h}_{\boldsymbol{k}}^{\boldsymbol{C}} = Q^{-1}\left(\boldsymbol{r}_{\boldsymbol{k}}^{\boldsymbol{C}}\right)\left(\rho_k \boldsymbol{h}_{\boldsymbol{k}}^{\boldsymbol{R}} + \boldsymbol{m}(\phi_k^C, \theta_k^C)\right) \tag{6}$$

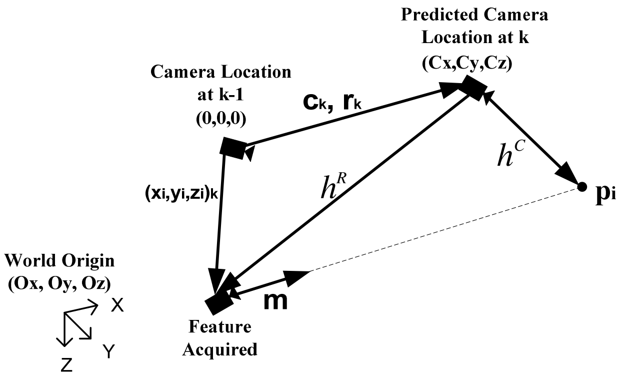

Figure 2 Measurement Model

where $Q^{-1}\left(\boldsymbol{r}_{\boldsymbol{k}}^{\boldsymbol{C}}\right)$ is the inverse rotation matrix from the camera frame at time $k-1$ to camera frame at time k. From vector $\boldsymbol{h}_{\boldsymbol{k}}^{\boldsymbol{C}}$, the feature location on image plane can be found by

$$\boldsymbol{h}_{\boldsymbol{k}}^{\boldsymbol{U}} = \begin{bmatrix} u_k \\ v_k \end{bmatrix} = \begin{bmatrix} \dfrac{s_x h_{y,k}^C}{h_{x,k}^C} \\ \dfrac{s_y h_{z,k}^C}{h_{x,k}^C} \end{bmatrix} \tag{7}$$

where s_x and s_y is the scaling factor obtained through camera calibration.

E. *Composition Step*

In order to apply the correction and update the camera reference frame to the new camera position, an additional composition step is necessary. The world reference frame parameters and features parameters are updated by applying reference frame transformation from the camera location at time k-1 to camera location at time k. The EKF covariance matrix P_k is also updated through

$$P_k = JP_kJ^T \tag{8}$$

where J is the Jacobian matrix of the composition equations.

III. EXPERIMENT AND RESULTS

In order to examine the algorithm's performance in a realistic low-altitude flying condition, a test flight was conducted with the support of Sander Geophysics Ltd. To achieve the true flying altitude of the UAS as much as possible (<100m), a simulated UAS (SUAS) was used to carry all sensors and was towed by a helicopter via a tow rope of 33 meters long (Figure 3).

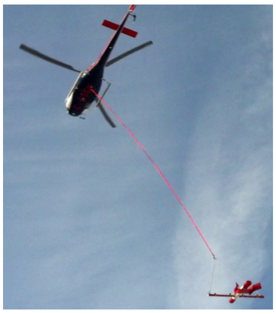

Figure 3 Simulated UAS towed by helicopter

Sensors mounted on the SUAS included one wide angle CCD camera with 6 mm focal length capturing monocular image sequence at 30 fps, a pair of narrow angle CCD cameras for binocular images, one GPS antenna, and one INS/GPS navigation unit Athena GS-111m (see Figure 4). Analog video captured by the cameras was sent to the helicopter, digitized with 720x480 resolution using a PC/104+ MPEG4 video encoder from Parvus, and time-stamped on the image screen for post-flight synchronization with the inertial measurements. A snapshot of the digitized video is shown in Figure 5. In addition, the helicopter carried radar altimeter and GPS antenna.

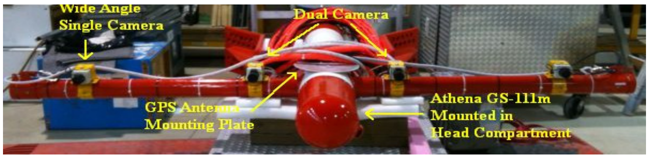

Figure 4 Sensors mounted on Simulated UAS

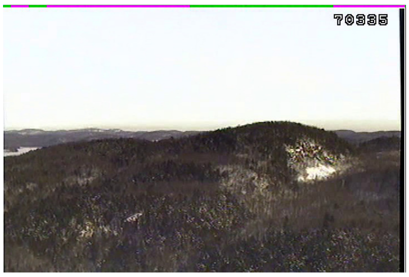

Figure 5 Image from monocular camera with GPS second timestamp

The algorithm described in section II was implemented using Python programming language. An open source machine vision library OpenCV [8] was utilized to perform feature extraction and tracking. The feature extraction method used was the Shi-Tomasi corner detector [9]. Feature tracking was accomplished through the pyramid implementation of Lucas-Kanade optical flow method [10].

As a preliminary test, an image sequence of 200 frames was processed with the algorithm. The camera position at the 1st frame was defined as the world frame origin. 40 features points were initialized at the 1st frame, added to the Extended Kalman Filter state vector, and updated on every subsequent frame. At the end of the sequence, the inverse depths of most features converged to a stable value except for a few outliers (Figure 6).

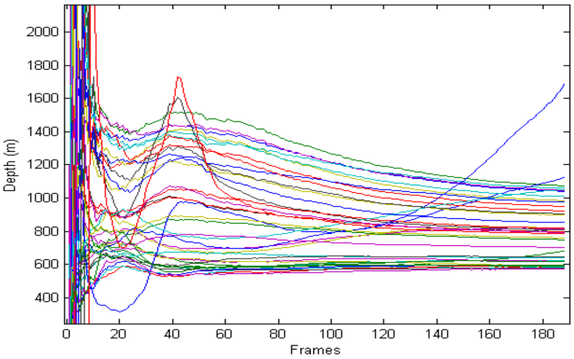

Figure 6 Features Range over Iterations

The accuracy of the result was analyzed by comparing to the digital elevation models (DEM) [10], which is referred as ground truth in the remaining of this paper. First, the features' positions were converted to Euclidean representation in world reference frame.

$$\begin{bmatrix} X_i^W \\ Y_i^W \\ Z_i^W \end{bmatrix} = Q^{-1}(O_{XYZ}^c, W_{XYZ}^c)\left(\begin{bmatrix} x_i^C \\ y_i^C \\ z_i^C \end{bmatrix} + \frac{1}{\rho_i}\boldsymbol{m}(\phi_i^C, \theta_i^C)\right) \qquad (9)$$

where $Q^{-1}(O_{XYZ}^c, W_{XYZ}^c)$ is the inverse transformation matrix from camera reference frame to world reference frame. Then the feature coordinate were converted into UTM representation using the GPS coordinates and SUAS orientation in the 1st frame. The resulting terrain map is shown in Figure 7, and the ground truth map is shown in Figure 8. Due to lack of texture in part of the image, there were very few feature points in the left side of the map. However, for area where sufficient number of features can be extracted (front and right side), the camera-generated terrain map accurately mirrors the ground truth map. In addition, the algorithm is able to produce a higher resolution map than the ground truth map in the rich texture region. This suggests that with a high resolution high dynamic range camera, the algorithm would be able to produce high resolution terrain map and detect obstacles that are not shown in DEM, such as tall trees, steep hills, electrical power line towers, etc.

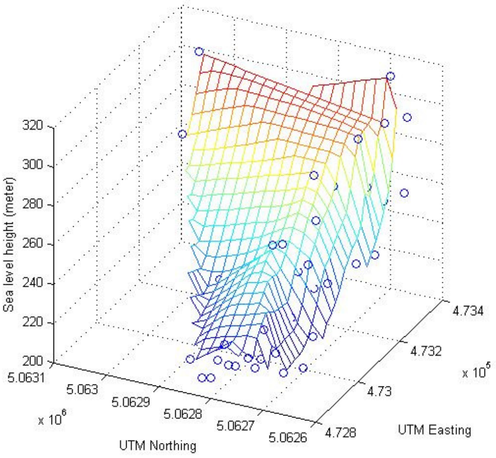

Figure 7 Terrain Map generated by monocular camera

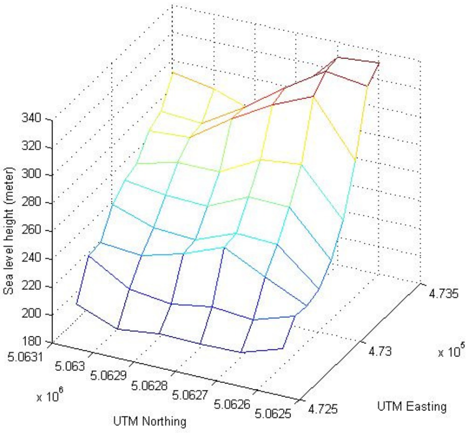

Figure 8 Digital Elevation Model (Ground Truth)

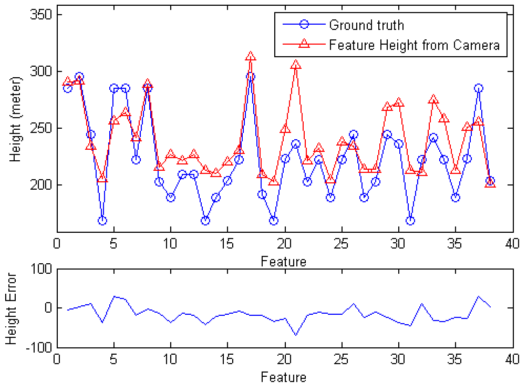

Figure 9 Point to point comparison of camera result and ground truth

A point-to-point comparison of the camera-generated features height and the closest point on ground truth map is shown in Figure 9. The algorithm results generally agree with the ground truth, considering the latter has a much lower resolution. To better understand the error contributed by the ground truth map resolution, the height difference between the estimated map and ground truth is compared to the values measured by radar altimeter (Table I). From the statistical comparison, the camera result is only slightly worse than the radar result. Therefore, we conclude that the result obtained through the algorithm described is accurate.

Table I Height Difference Comparison

	Height Difference between Camera Result and Ground Truth	Height Difference between Radar and Ground Truth
Mean	-15.87	-7.16
Maximum	69.93	43.95
Minimum	2.87	0.01
Standard Deviation	20.93	11.21

Finally, the SUAS position can be reconstructed from the world reference frame parameters. The reconstructed SUAS flight path is compared to the actual flight path recorded by GPS (Figure 10). Over the more than 250 meters of reconstructed trajectory, the camera result agrees with the GPS result very well for the first 200 meters, with slight difference in the last 50 meters. On the other hand, the camera result provides more detail SUAS position than the GPS.

IV. Conclusion

An inertial aided inverse depth Extended Kalman Filter framework for sparse terrain map generation from monocular image sequence is presented. The algorithm was preliminarily tested with realistic outdoor low-altitude aerial video data collected using a SUAS towed by a helicopter. The outputs of the algorithm including terrain elevation map and SUAS position were compared to ground truth data. Despite the large scale of the measurement environment, the result obtained by the algorithm provides accurate estimate of features ranges and the sensor's own location. The key factor that limits the resolution of the generated terrain map is the quality of the captured video. Being a vision system, the algorithm requires rich texture in the image to extract sufficient number of features. With a high resolution high dynamic range camera, the algorithm would be able to produce high resolution terrain map and detect obstacles.

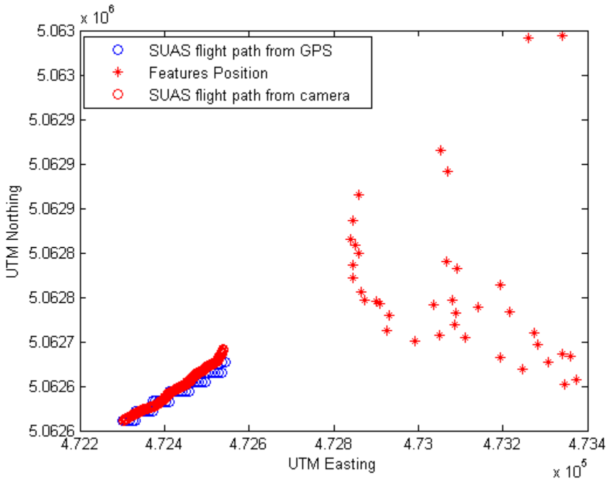

Figure 10 SUAS flight path

References

[1] Moufid Harb, Rami Abielmona, Kamal Naji, and Emil Petriu, "Neural Networks for Environmental Recognition and Navigation of a Mobile Robot," in *Proceedings of IEEE International Instrumentation and Measurement Technology Conference*, 2008, pp. 1123 - 1128.

[2] M. M. Saad, Chris J. Bleakley, and Simon Dobson, "Robust High-Accuracy Ultrasonic Range Measurement System," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 10, pp. 3334 - 3341, October 2011.

[3] Ming-Chih Lu, Chen-Chien Hsu, and Yin-Yu Lu, "Distance and Angle Measurement of Distant Objects on an Oblique Plane Based on Pixel Variation of CCD Image," in *Proceedings of IEEE International Instrumentation and Measurement Technology Conference*, 2010, pp. 318 - 322.

[4] E. Hanna, P. Straznicky, and R. Goubran, "Obstacle Detection for Low Flying Unmanned Aerial Vehicles Using Stereoscopic Imaging," in *Proceedings of IEEE International Instrumentation and Measurement Technology Conference*, 2008, pp. 113 - 118.

[5] Farzin Amzajerdian, Diego Pierrottet, Larry Petway, Glenn Hines, and Vincent Roback, "Lidar systems for precision navigation and safe landing on planetary bodies," in *Proceedings of International Symposium on Photoelectronic Detection and Imaging*, 2011, pp. 24-26.

[6] Javier Civera, Andrew J. Davison, and J.M.M Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, pp. 932-945, Oct. 2008.

[7] Javier Civera, Óscar G. Grasa, Andrew J. Davison, and J. M. M. Montiel, "1-Point RANSAC for EKF Filtering: Application to Real-Time Structure from Motion and Visual Odometry," *Journal of Field Robotics*, vol. 27(5), pp. 609-631, Oct. 2010.

[8] OpenCV dev team. OpenCV V2.3 Documentation. [Online]. http://opencv.itseez.com/

[9] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition*, June, 1994.

[10] J. Y. Bouguet, *Pyramidal implementation of the Lucas Kanade Feature Tracke.*: Intel Corporation Microprocessor Research Labs, 2003.

[11] The CGIAR Consortium for Spatial Information. [Online]. http://srtm.csi.cgiar.org/