MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Vision-based UAV Navigation in Mountain Area

Jihwan Woo, Kilho Son, Teng Li, Gwansung Kim, In So Kweon
Dept of Electrical Engineering, KAIST
373-1 Guseong-dong, Yuseong-gu, Daejeon,
Korea
{jhwoo, khson, tengli, iskweon}@rcv.kaist.ac.kr

Abstract

Most vision-based UAV (Unmanned Aerial Vehicle) navigation algorithms extract manmade features such as buildings or roads, which are well structured in urban terrain, using the CCD camera. But in the mountain area, extracting, matching or tracking features is more difficult than doing the tasks in the urban terrain. And a CCD camera cannot carry out the computer vision algorithm that is required for the UAV navigation in the night or under dark situation. In this paper, we introduce a new approach for vision based UAV localization method. In our proposed system an UAV uses only DEM (Digital Elevation Map), IR (Infra-Red) image sequences and an altimeter. It can estimate its position and orientation with hypothesis and verification paradigm..

1 Instructions

The performance and autonomous on-board processing capabilities of UAVs have significantly improved in the last 10 years with respect to demands from environmental monitoring or traffic surveillance. Among the several indispensable technologies that an UAV must have, the reliable localization is an essential component of a successful autonomous flight. [3]

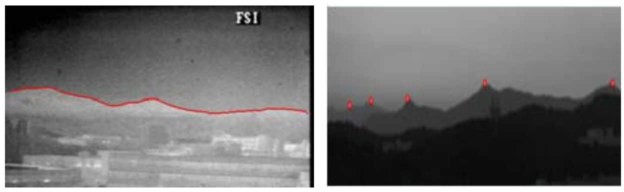
Most UAV autonomous navigation techniques are based on GPS(Global Positioning System) and the fusion of GPS with INS(Inertial Navigation System) information. However, GPS is sensitive to the signal dropout, hostile jamming and INS accumulates position error over time. When GPS and INS cannot work, the computer vision is an alternative for the navigation. This is a start of the visual odometer concept in UAV.[3,4,6]

Many researches on the visual odometer have been used in the urban area with a CCD color camera system. However, in the natural terrain environments such as mountain area, defining landmark or extracting feature set is not easy because the CCD color camera system cannot work in the night or under weak illuminated condition.[3] For solving these problems, we proposed a robust horizon and mountain peak extraction method under noisy images and bad weather, based on characteristics of human visual system such as binding, which is a main process of the visual perception. (See Fig 1)

In this paper, we estimate UAV position by matching extracted horizon and mountain peaks in the aerial images with those from DEM in the situation of knowing altitude. We suggest two stages for UAV localization. In the first stage, UAV estimates coarse location by matching reconstructed mountain peaks and mountain peaks extracted from DEM. For this stage mountain peaks extracted each frames are matched by curvatures and reconstructed in affine space by factorization. At the second stage, UAV can estimate its fine location by matching horizon in the aerial images and horizon generated from DEM. For generation of horizon from DEM, we use coarse UAV location estimated in the first stage as a virtual camera center. Virtually generated horizon is matched with horizon in the aerial image by MCMC(Monte Carlo Markov Chain) method.[9]

We analyze our algorithm with respect to several noise sources such as resolution of DEM and altimeter or the accuracy of mountain peaks extracted from IR image sequences. [5, 6]

In the following sections the brief of our system will be introduced. After this, an image matching method with two consecutive IR images taken in mountain area and matching method between image and DEM are summarized. Finally, the two stages of position estimation algorithm are explained with analysis of our system's robustness being presented.



(a) (b)

Figure 1 (a) Horizon and (b) peaks from IR images[1]

2 System Framework

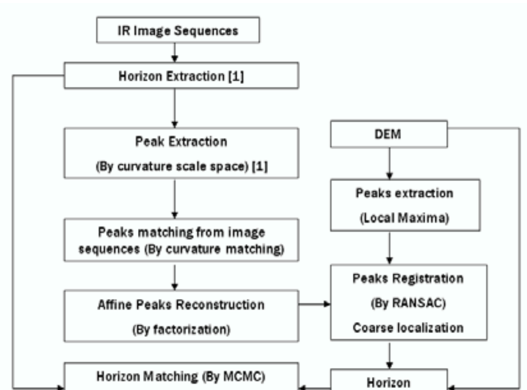


Figure 2 System Framework

Figure 2 shows the overall structure of our proposed

UAV position estimation system. First we can extract peak from DEM by finding local maxima. Using the curvature matching method, we can match the mountain peaks extracted from IR images sequence. Then, we can calculate the 3D structure of matched mountain peaks for the factorization method. The reconstructed 3D structure is an affine model because the distance from UAV to mountain is relatively longer than that between mountain peaks.

Finally, by matching the affine reconstructed peaks and DEM peaks, we can estimate the UAV position.

3 Affine Reconstruction of Mountain Peaks

3.1 Feature matching using curvature

For IR image, the intensity difference and complexity around horizon are not large. It is difficult to find correspondent peaks by using previous template-based matching method directly. We choose the curvature defined in [1] for another matching measure. Figure 3 shows the curvature of extracted peaks and their neighborhoods from two consecutive images are similar. If the location of a mountain peak is P, we can make a curvature vector using the N neighborhood pixels' curvature.

$$CV_P = [C_{P1}, C_{P2}, \cdots C_{PN}]^T \tag{1}$$

With this curvature vector and distance between the mountain peaks, we make a new matching model for the mountain peak. When the two features P and Q are given and if the value of equation (2) is smaller than a certain threshold, we define it as the true correspondence.

$$\alpha \left\| CV_P - CV_Q \right\| + \beta \left\| P_P - P_Q \right\| \tag{2}$$

$$P_I : \textit{Pixel location of } I^{th} \textit{ Peak}$$

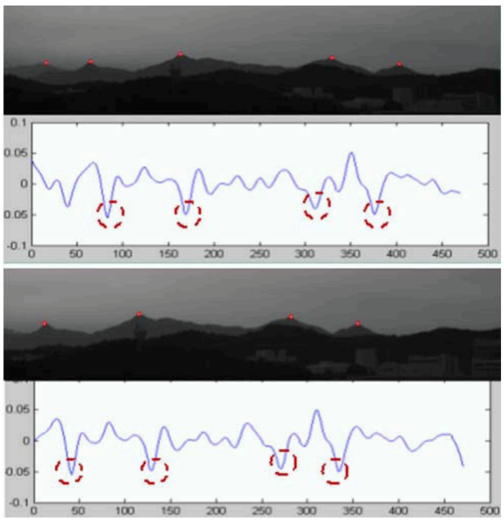


Figure 3 Curvature value due to the different frame

3.2 Factorization[7]

We can reconstruct the mountain peak geometry from the matched feature set in the image sequences. Among the several 3D reconstruction methods from images, factorization method is robust to noise and may be applied in finding the solution without any recursive calculation.

The depth between the mountain peaks is smaller than the distance between UAV and mountain peaks so the affine model is used for reconstruction. We can make a 2m by n matrix with the peak points which is obtained by n-peaks in m-frames, x and y direction. [2,7]

$$\hat{W} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^m & x_2^m & \cdots & x_n^m \\ y_1^1 & y_2^1 & \cdots & y_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ y_1^m & y_2^m & \cdots & y_n^m \end{bmatrix} = \begin{bmatrix} M^1 \\ M^2 \\ \vdots \\ M^m \end{bmatrix} [X_1\, X_2 \cdots X_n] \tag{3}$$

The matrix should be divided into M and X matrix by rank 3 condition. We can divide the W matrix by using SVD. The matrix X contains affine reconstructed information of the mountain peaks.

$$\hat{W} = U_{2m\times3} D_{3\times3} V_{n\times3}{}^T \tag{4}$$

So, the solution has two cases.

Case 1: $\hat{M} = U_{2m\times3} D_{3\times3}$, $\hat{X} = V_{n\times3}{}^T$. (5)

Case 2: $\hat{M} = U_{2m\times3}$, $\hat{X} = D_{3\times3} V_{n\times3}{}^T$ (6)

We make simulation tool. When the 3D peak data is given, we can generate several camera views. From these views, we can make affine reconstructed environment. (see Figure 4)

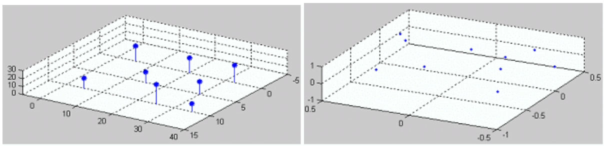


Figure 4 Real 3D peaks data set (left), affine reconstructed peaks from factorization (right)

4 Registration of Mountain Peaks

4.1 Mountain Peak Extraction from DEM

We extract peaks from DEM by searching the local maxima points which are higher than threshold height value.

4.2 Registration

We carry out the affine reconstruction of the mountain peaks by factorization method. It is difficult to register directly this reconstruction to the DEM which is Euclidian space. So we need an affine transformer to register DEM with reconstructed peaks.

$$\mathbf{X}_{\text{euclidean}} = \mathbf{T}_{\text{affine}} \mathbf{X}_{\text{affine}}$$

$$\mathbf{T}_{\text{affine}} = \begin{bmatrix} \mathbf{M}_{3\times3} & \mathbf{m}_{3\times1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{7}$$

$X_{\text{euclidian}}$: peak's coordinate in DEM(Euclidian Space)
X_{affine} : Peak's coordinate in Affine reconstructed space

T_{affine}, has 12 dof(degree of freedom), so we can calculate T_{affine} by 4 correspondences between $X_{\text{euclidian}}$ and X_{affine}. After fixing 4 peaks points in X_{affine} (affine reconstructed space), we can estimate T_{affine} by selecting 4 points in X_{affine}. It is very similar with RANSAC. We can verify the estimated T_{affine} by the error measure in equation (8).

$$Error = \left\| \mathbf{X}_{euclidean} - \mathbf{T}_{affine}\mathbf{X}_{affine} \right\| \tag{8}$$

Figure 5 shows the registration results. The affine reconstructed mountain peaks in Figure 4 are registered with 50 DEM peaks.

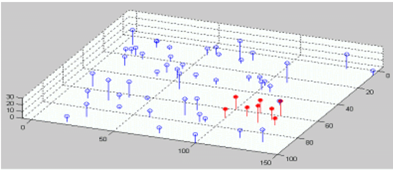


Figure 5 Registration results: with 50 peaks

5 UAV Localization

5.1 Coarse Pose Estimation

UAV's camera projection model is affine, so we can obtain the projection model P_A using matching results between peaks in image and peak in DEM through Gold Standard algorithm.[8]

$$x = \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} X = \begin{bmatrix} P^{1T} \\ P^{2T} \\ 0^T \; 1 \end{bmatrix} X = P_A X \tag{9}$$

A camera projection matrix is divided by an intrinsic matrix and an extrinsic parameter that contains camera motion with respect to the world coordinate. We know the intrinsic matrix because the initial one has not changed.

$$P_A = \begin{bmatrix} m_1^T & t_1^{'} \\ m_2^T & t_2^{'} \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1^T & t_1 \\ r_2^T & t_2 \\ 0 & 1 \end{bmatrix} \tag{10}$$

For the localization, we should know the extrinsic matrix. By multiplying the inverse of intrinsic matrix,

$$\begin{aligned} P_A &= K[R|T] \\ [R|T] &= K^{-1}P_A \end{aligned} \tag{11}$$

The vector T shows the UAV's position in the DEM. But in the affine model the translation T is valid up to scale factor.

$$[R\,|\,T] = K^{-1}P_A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \tag{12}$$

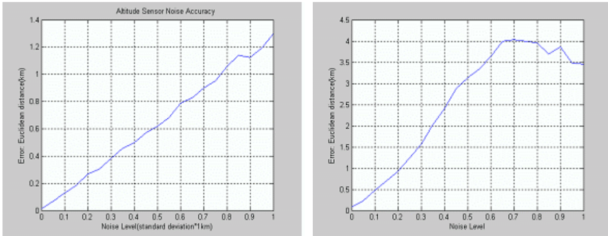
$$t_1 = kX, t_2 = kY, t_3 = kZ$$

When we know the true altitude Z measured by an altimeter which is usually equipped in most UAVs, we can estimate the real position parameter X, Y.

$$k = \frac{t_3}{Z},\quad X = \frac{t_1}{k} = \frac{t_1}{t_3}Z,\quad Y = \frac{t_2}{k} = \frac{t_2}{t_3}Z \tag{13}$$

5.2 Gaussian Noise Test

Figure 6 shows an error when Gaussian noise is added to altitude and image pixel location. The noise level means standard deviation of Gaussian noise. In equation (13), the estimation of position X and Y is in proportion to the altitude Z, linearly. When UAV flies at high altitude, the altimeter error can be ignored.



(a) (b)

Figure 6 Noise test for altitude(a) peak's location(b)

5.3 Fine Pose Estimation

There exists noise in extracting mountain peaks in the aerial images and DEM. The estimated result does not guarantee that it is the optimal solution. For finding optimal solution we add hypothesis and verification procedure. The solution in equation (13) is the initial position. From the initial position, we can generate synthesized horizon with DEM. This is a hypothesis step. If the hypothesis is correct, generated horizon and extracted horizon in the aerial images should be aligned in many pixels. This is a verification step. We implement this procedure using MCMC[9] method.

(a) Image generation Step (Hypothesis)
(X,Y,Z): UAV location, Z is known by altimeter
(x,y,z,): DEM coordinate, z is known by DEM height

We use OpenGL for generating horizon from DEM. We design this problem as that with 4-degree of freedom: two coordinate parameters (X,Y) in UAV's position and two coordinate parameters (x,y) in DEM. The vector from (X,Y,Z) to (x,y,z) means looking direction of camera in UAV, so we can generate the synthetic image from (X,Y,Z).

(b) Image alignment Step (Verification)

For verifying the alignment, we check the number of overlapped horizon pixel between aerial image and synthesized image. Table 1 shows the proposed verification algorithm. The number of pixels is scoring function value in UAV position Θ_t. For jumping distribution, we simply use uniform distribution which randomly moves to next step in the boundary. If we select large searching boundary, it takes long time for computation.

Step1. Initialize Θ_0 ($f(\Theta_0)$> 0(scoring function))
Step2. Using current sample value and jumping distribution Jt ($\Theta_{t\text{-}1}|\Theta^*$) set next candidate sample Θ^*
Step3. Calculate ratio follow

$$r = \frac{f(\theta^*)J_t(\theta_{t-1}|\theta^*)}{f(\theta_{t-1})J_t(\theta^*|\theta_{t-1})}$$

Step4. If u<min(r,1) $\Theta_t = \Theta^*$ else $\Theta_t = \Theta_{t-1}$
Step5. Go to setp2

.

Table 1 Proposed Verification Algorithm

6 Experimental Result

Figure 7 display DEM in OpenGL. The area is the West Sea of Korea. Latitude ranges from 36.5895° to 36.6086° and longitude is from 126.2454° to 126.2335°. Figure 8 shows the alignment result. For hypothesis, we set searching boundaries at 200m in each direction of X, Y, x, and y. The maximum iteration number allowed is 250. We can not obtain a real aerial image on this site, so we add several levels of Gaussian noise in the synthesized image.

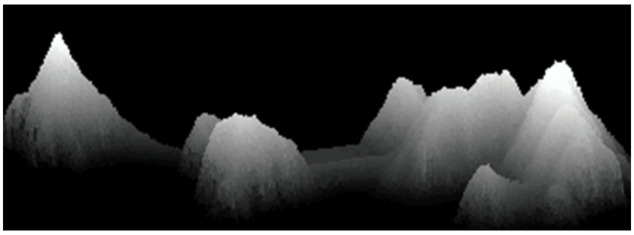


Figure 7 DEM in west sea of Korea

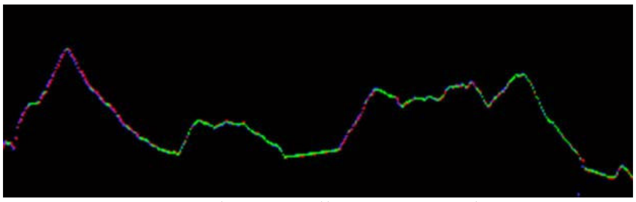


Figure 8 Alignment Result

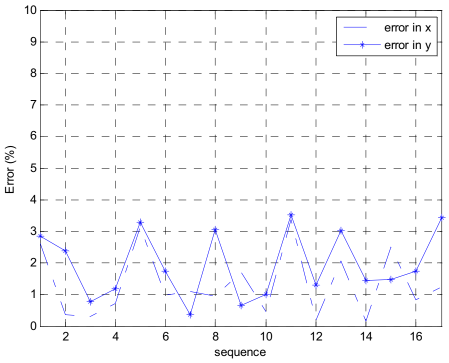


Figure 9 Simulation Result

Table 2 Pose Estimation Result

	Ground Truth	Estimated value	Error(%)
X	-2820m	-2,901m	2.86%
Y	2630m	2699m	2.63%

Table 2 shows estimation result in 1 pixel error in the extracted horizon. In the figure 6, at the coarse estimation stage, the estimated error is about 4km, and after fine localization the error is below 1km. In simulation set up, the error is under 4% in x and y direction. (See Figure 9) The error is considered small compared to UAV's altitude.

7 Conclusion

In this paper we have proposed a new system for practical vision-based UAV localization algorithm using an altimeter and DEM. We test robustness of our algorithm with respect to several noise sources. We use mountain peaks and horizon as new features for UAV localization. [1] MCMC method is used for finding solution efficiently. The initial solution is significantly important for finding an optimal solution. We divide our system into two stages of searching an initial solution and finding the optimal solution, to increase its robustness and efficiency. Our algorithm is tested only as a simulation set-up, which is a limitation of our work. We will expand our work to the realm of real situation.

The algorithm works when GPS is jammed and INS data has enormous errors. Our proposed algorithm will be used as the initial value of filter which estimates the UAV's location.

In the near future we will make a probabilistic model for managing a feature set for efficiently registering affine reconstructed map to DEM.

Acknowledgments

This research has been supported by from ADD (No. UD069009ED) and the Korean Ministry of Science and Technology for NRL Program (No. M1-0302-00-0064). Now, Gwansung Kim is working at ADD.

References

[1] Jihwan Woo, In So Kweon et al, "Robust Horizon and Peak Extraction for Vision-based Navigation", IAPR Machine Vision Applications, 2005

[2] Emanuele Trucco, Alessandro Verri, "Introductory Techniques for 3-D Computer Vision" Prentice Hall 1998

[3] Fernando Caballero et al, "A visual odometer without 3D reconstruction for aerial vehicles. Application to building inspection", IEEE ICRA, 2005

[4] F. Cozman, E. Krotkov, "Position Estimation from Outdoor Visual Landmarks for Teleoperation of Lunar Rovers." Third IEEE Workshop on Applications of Computer Vision, pp. 156-161, 1996.

[5] K. Rushant, L. Spacek, "An Autonomous Vehicle Navigation System using Panoramic Machine Vision Techniques" Technical report CSM-298, 1997.

[6] P. C. Naval Jr., M. Mukunoki, M. Minoh, and K. Ikeda, "Estimating Camera Position and Orientation from Geographical Map and Mountain Image." 38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers, pp.9-16, 1997.

[7] C. Tomasi, T. Kanade, "Shape and motion from image streams under orthography: A factorization approach", International Journal of Computer Vision, 9(2), pp. 137-154, 11. 1992.

[8] Richard Hartley, Andrew Zisserman, "Multiple View Geometry" 2nd edition, Cambridge

[9] Christian P. Robert, George Casella, "Markov Chain Monte Carlo in Practice", 2nd edition, Springer