The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

UAV global pose estimation by matching forward-looking aerial images with satellite images

Kil-Ho Son, Youngbae Hwang, and In-So Kweon

***Abstract* —A global pose estimation method of an Unmanned Aerial Vehicle (UAV) by matching forward-looking aerial images from the UAV flying at low altitude with down-looking images from a satellite is proposed. To overcome the limitation of significantly different camera viewpoints and characteristics, we use buildings as a cue of matching. We extract buildings from aerial images and construct a 3D model of buildings, using the fundamental matrix. We estimate the global pose of the vehicle by matching 3D structure of buildings with satellite images, using a particle filter. Experimental results show that the proposed approach is a promising method to the global pose estimation of the UAV with forward-looking vision data.**

I. Introduction

POSE estimation of an aerial vehicle is a fundamental and challenging issue in the aerospace community. A global positioning system (GPS) is usually used for the localization of a vehicle. However, the GPS is unstable when there is jamming around the vehicle. Even though an inertial measurement unit (IMU) system can be applied to estimate the pose of the vehicle, The IMU system accumulates errors as time goes on. Also, the system can only estimate a local pose of a vehicle with reference to an initial pose. A camera system can be a complementary solution to overcome the above limitations. Although a camera system also has its limitations, such as an aerosol, a weather condition and an illumination, it is not affected by jamming around the vehicle or the accumulation of errors. In addition, it is the global localization method, if we can match aerial images with satellite images, which are geometrically referenced.

Various works have been presented for the UAV pose estimation using a camera. Sinopoli et. al. proposed a localization algorithm, using a GPS, an IMU and a camera system [1]. They estimated an initial pose of the vehicle with the GPS and IMU system and refined it with a camera. Although their results were good, the GPS and IMU have their limitations, as described above. A localization method matching digital elevation maps with aerial images has been suggested [2], and relative and absolute UAV localization methods by matching between satellite images and down-looking aerial images have been studied [3]. Caballero et. al. estimated a relative position of the UAV by calculating homographies among down-looking aerial scenes with the assumption that the ground is a plane[4][5]. All these methods showed good results with down-looking camera systems because matching between satellite images and down-looking aerial images is feasible as a result of using similar viewpoints. However, images taken by a down-looking camera tend to have blurred effects and little ground information when a UAV navigates at low altitude and high speed, especially as shown in Figure 1a and 1c. Although short exposure time can reduce the blurred effect, images can be too dark to obtain adequate information from them. A forward-looking camera system, however, can take well-focused and bright images with more ground information (Fig. 1b and 1c).

We propose a method of UAV global pose estimation by

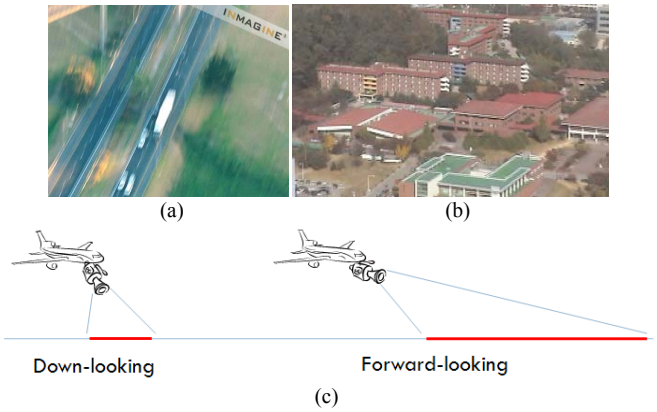


Fig. 1. Advantages of forward-looking camera: (a) down-looking aerial image; (b) forward-looking aerial image; and (c) Amount of ground information for each camera system.

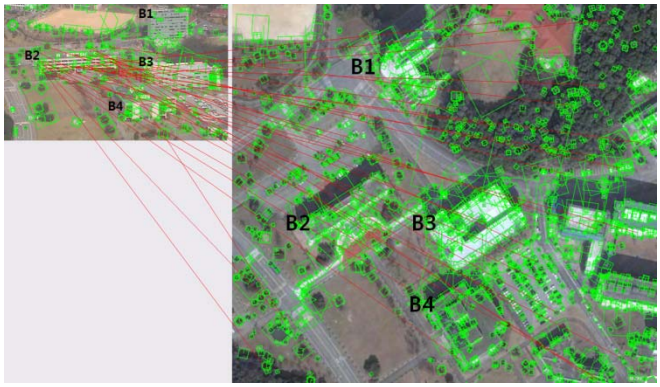


Fig. 2. SIFT matching results (green rectangles are the SIFT descriptors and red lines represent corresponding points).

This research was supported by ADD project, 'A Study on IR Image matching for estimating UAV position' (No. UD069009ED), and by the Defence Acquisition Program Administration and Agency for Defense Development, Korea, through the Image Information Research Center at KAIST under the contract UD070007AD

Kil-Ho Son is with the ADD, Daejeon, Korea (corresponding author to provide phone: 82-42-821-3674; e-mail: khson@add.re.kr).

Youngbae Hwang is with the KAIST, Deajeon, Korea (e-mail : unicorn@rcv.kaist.ac.kr).

In-So Kweon is with the KAIST, Daejeon, Korea (e-mail : iskweon@ee.kaist.ac.kr).

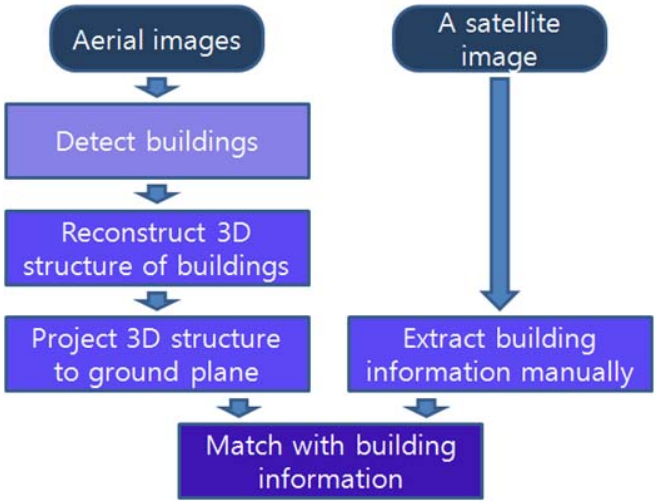
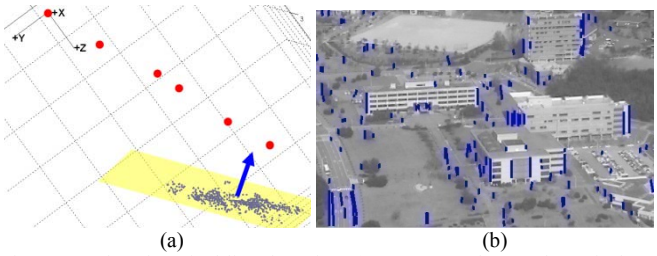


Fig. 3. An illustration of a matching system for global pose estimation of a UAV.

matching forward-looking aerial images with satellite images. The main problem of the matching is the significant difference in characteristics, such as type of cameras and a wide baseline between forward-looking images and down-looking satellite images. Conventional matching schemes by the scale invariant feature transform (SIFT) [6] algorithm using appearance information does not work well because of the wide baseline between two images, as shown in Figure 2. We detect SIFT descriptors in an aerial image and find the best corresponding points in a satellite image. It shows that most correspondence is false. Matching two images only using appearance information is difficult because forward-looking aerial images contain sides of buildings, whereas down-looking satellite images have only rooftops of buildings.

We suggest a semantic matching method between forward-looking aerial images and down-looking satellite images based on information of buildings to estimate a global pose of a UAV, as shown in Figure 3. First, we detect buildings in aerial images based on cues that columns of buildings are normal to the ground and buildings include man-made regions and repeated patterns. Next, we construct a 3D model of the detected buildings and find the relative pose of the camera, using structure from motion (SFM) [7]. Then, we project the 3D structure of buildings to make images, using the azimuth and elevation angles of the satellite. Finally, building information from the aerial images and satellite images is used to match the images, and we estimate the global pose of a UAV.

The paper is organized as follows: Section 2 explains the details of the building detection method in aerial images; Section 3 describes a method for 3D construction and reprojection of buildings; Section 4 discusses the matching method by considering errors of estimating intrinsic and extrinsic camera parameters; Section 5 presents the experimental results; and the conclusion is discussed in Section 6.



(a) (b)

Fig. 4. Results of vertical line detection: (a) constructed ground quasi-plane and its normal vector; and (b) vertical lines about the ground planes.

II. BUILDING DETECTION

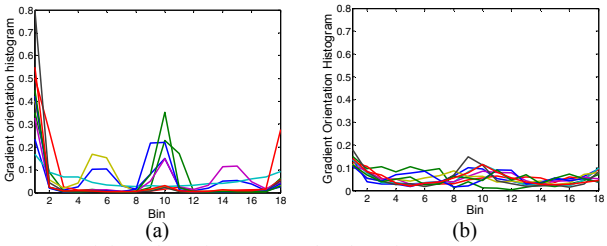
To detect buildings from aerial images, we generate candidates with a cue that columns of buildings are normal to the ground plane. Buildings are then detected based on the observation that buildings are parts of the man-made regions and have repeated patterns.

A. Detection of the ground plane and its normal vector

We cannot construct an accurate 3D model of buildings with a series of aerial images by using an SFM method because the distance between the camera and the building is too long compared with the size of the building. However, we can construct the 3D points of the ground as a quasi-plane through the SFM method, as shown in Figure 4a [7]. Let $ax+by+cz+d=0$ be the ground plane equation. Then, (a,b,c) is a normal vector of the ground plane. $P_1, P_2 \ldots P_k \ldots P_N (P_k = [x_k, y_k, z_k])$ are the N number of constructed 3D points of the ground plane using the SFM method. Then, all points have to satisfy the following equation:

$$A\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \bar{0} \quad , \quad A = \begin{bmatrix} P_1 & 1 \\ P_2 & 1 \\ \vdots & \vdots \\ P_N & 1 \end{bmatrix}. \tag{1}$$

Equation (1) is the over constrained case of simultaneous equations. We can solve the equation using singular value decomposition (SVD). Once we estimate the normal vector of the ground plane, we can find the direction of vertical lines to the ground plane in aerial images by projecting the normal vector to the image plane. Some 3D points from the buildings will result in erroneous results for detecting the normal vector of the ground plane, because the constructed 3D points of the ground plane are not plane but quasi-plane. We therefore include a $\pm$ 5-degree error when detecting vertical lines in aerial images. Figure 4a shows the SFM results with aerial images that contain 3D points of the ground (blue points) and pose of the camera (red points) and a normal vector of the ground plane. If we estimate how the normal vector of the ground plane projects to the aerial images, we can find the vertical lines about the ground plane in aerial images, as shown in Figure 4b. Our method detecting a direction of



(a) (b)

Fig. 5. Training data for man-made detection: (a) ten representative orientation histograms of training data for man-made regions; and (b) Ten representative orientation histograms of training data for the regions of the nature.

vertical lines to the ground plane in aerial images can be applied to the roll movement of the UAV because we estimate a direction of vertical lines through finding ground planes in aerial images

B. *Man-made region detection*

Given an input image, the man-made region detection can be considered as a classification problem for each patch (block, site). Each patch can be classified into the man-made region or the non-man-made ones. Let Y be the observed data as $Y = (y_m)_{m=1}^{M}$, where y_m is the data from the m^{th} site. Let the corresponding labels at the image site be given by $X = (x_m)_{m=1}^{M}$, where $x_m = \{0,1\}$. 0 means that the selected patch is a natural region, and 1 means the patch is a man-made region.

We use a gradient orientation histogram classifier because man-made objects usually contain structures, such as lines, corners and Y junctions, which have a dominant edge. To capture these characteristics, the input image is convolved with the derivative of Gaussian filters to yield the gradient magnitude and orientation at each pixel. Then, for an image patch, the gradients are combined to yield a histogram over gradient orientations. We weight each count by the gradient magnitude at each pixel and normalize the histogram instead of incrementing the counts in the histogram [8]. Then, the histogram is shifted in order to make the dominant orientation direction be in the first bin. This shift procedure makes features be orientation invariant. We define C as:

$$C(y_i) = \begin{cases} 1 & (\| H_i - H_{M_\alpha} \| < \| H_i - H_{N_\beta} \|) \\ 0 & (\| H_i - H_{M_\alpha} \| \geq \| H_i - H_{N_\beta} \|) \end{cases} \quad (2)$$

$$\alpha = \arg\min_k \| H_i - H_{M_k} \| \quad (3)$$

$$\beta = \arg\min_k \| H_i - H_{N_k} \| \quad (4)$$

where H_i is a vector that contains the orientation histogram of the i-th patch. H_M is a set of representative histogram vectors extracted from training data corresponding to man-made regions where H_{M_k} is a k-th vector. H_N is a set of vectors and H_{N_k} is the k-th vector corresponding to the natural regions. To extract 10 representative histograms, we eliminate the outliers, using a cross-validation algorithm, and make 10 representative models of a histogram by the K-mean clustering algorithm. Equation (2) means that an input patch is classified into the man-made region if its orientation histogram is closer to 10 man-made representatives than natural ones.

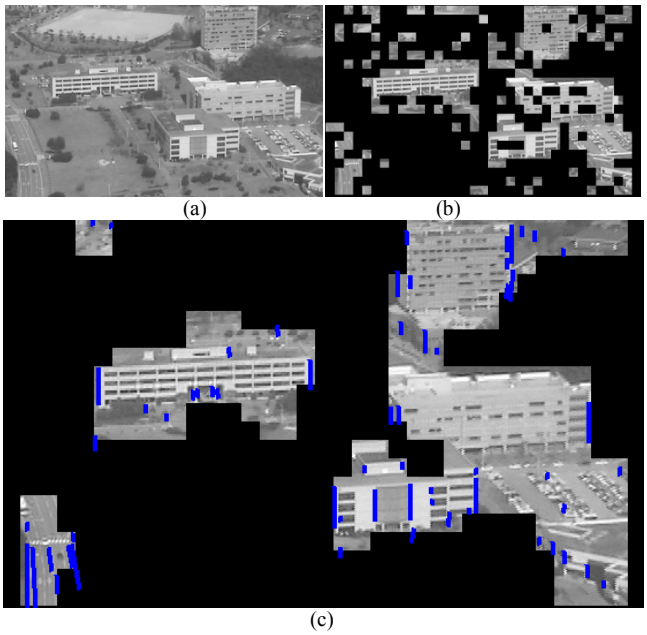

(a) (b) (c)

Fig. 6. Results of man-made region detection: (a) aerial input image; (b) man-made region detection before ICM; and (c) man-made region detection with ICM and normal lines about the ground plane in the man-made region.

Figure 5 shows the 10 representatives of training data. Note that the histograms of man-made regions have more than one significant peak in the histogram because of the structures in patches. In contrast, the results from natural patterns are distributed evenly. A significant difference in histograms between man-made and natural regions makes the classifier strong. We apply the iterated conditional modes (ICM) algorithm to use neighborhood information because man-made regions usually gather together [9]. Figure 6b is an input image and Figure 6b shows detected results whose black regions mean non-man-made regions before applying ICM. Figure 6c shows the final results of man-made region detection with ICM and normal lines about the ground plane in man-made regions, as explained in the previous chapter.

Because the sides of buildings are generally rectangles, we can generate candidates of buildings with two vertical lines. We construct all building candidates by combining two vertical lines in man-made regions, as shown in Figure7.

C. *Building decision algorithm among the remaining hypotheses*

We propose a building decision algorithm that selects true buildings among the remaining building hypotheses. By observing that the building texture has repeated patterns due to the windows or the bricks in the wall, we can use the main orientation around the corner to detect buildings. We rank all hypotheses that contain a common line because an extracted line can be a part of only one building. The score is estimated as below:

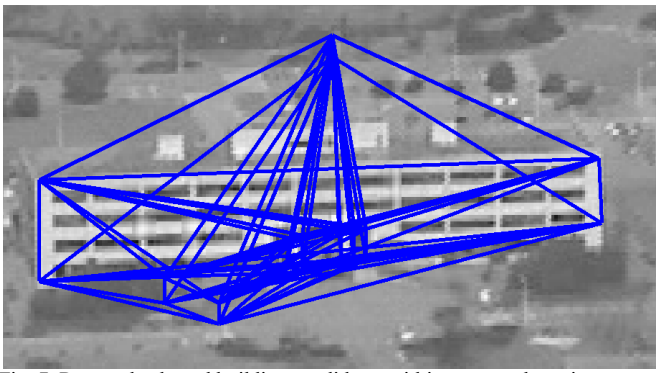


Fig. 7. Rectangle-shaped building candidates within man-made regions.

Score = # of corners whose main direction is similar to the main direction of the sum of the orientation histogram around corners

-

# of corners whose main direction is different to the main direction of the sum of the orientation histogram around corners.

Figure 8a shows corners and their main orientation after excluding the ground normal orientation direction within the rectangle candidates, and Figure 8b shows the sum of orientation histograms around the corners. The rectangle candidate is determined as a building because the main orientations of the corners are similar to the main direction of the sum of orientation histograms around the corners. Figure 9 shows the outlier case. Not only the sum of orientation histograms is randomly distributed but also the main orientations of the corners are almost random.

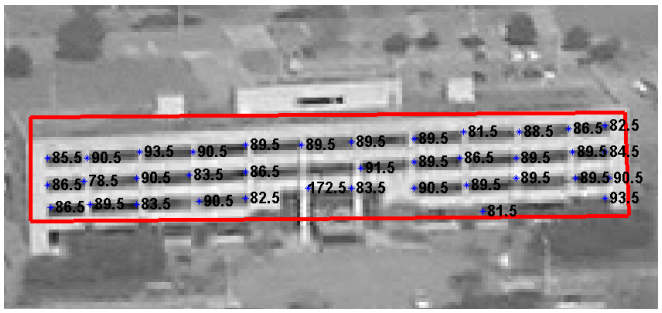
III. 3D CONSTRUCTION AND REPROJECTION OF BUILDINGS

A. Estimating intrinsic parameters in aerial images

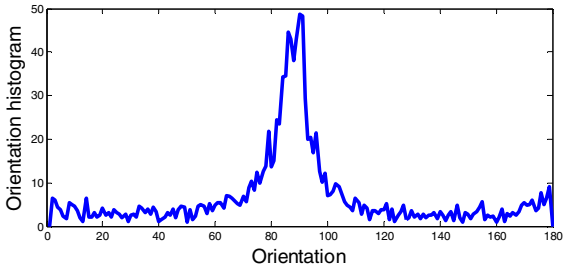
We find the intrinsic parameter using images that are collected from the UAV. If we assume that the intrinsic parameter of the camera is fixed and the UAV moves in various motions when it captures images, we can estimate the intrinsic parameter using the dual absolute quadric [9].

B. Estimating extrinsic parameters in aerial images

We assume that the UAV navigates at low altitude to allow robust computation of the fundamental matrix between consecutive aerial images. We find the corresponding points among a sequence of aerial images, using the KLT algorithm [10]. After estimating a fundamental matrix with the corresponding points, we calculate the extrinsic parameters of the camera by decomposing the fundamental matrix and estimate the 3D structure of the buildings by triangulation [7]. Although the corresponding points between aerial images have subpixel errors, there are some errors in the 3D structures of the building points because the points are triangulated using only the fundamental matrix. We therefore consider these errors when defining a cost-function matching between extracted building information in aerial and satellite images. The definition of cost function is explained in section 4.

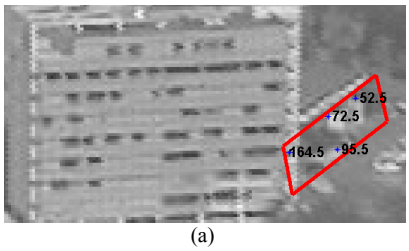


(a)

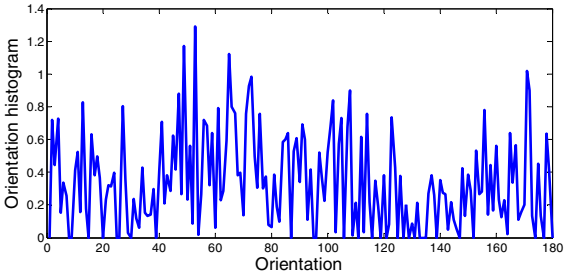


(b)

Fig. 8. Building detection result: (a) corner points and their main orientation except for the ground normal directions; and (b) sum of orientation histogram around the corner points.



(a)



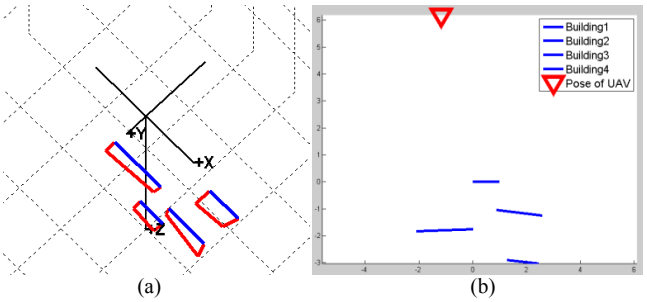
(b)

Fig. 9. Outlier regions: (a) corner points and their main orientation except for the ground normal directions; and (b) sum of orientation histogram around the corner points.

We construct the 3D structure of extracted buildings and relative aerial camera poses, as shown in Figure 10a. The origin of the coordinate is the position of the camera. The direction of the positive z-axis is the principal axis of the camera. Rectangles show the 3D position of buildings.

C. Projection of constructed buildings and cameras

We can find azimuth and elevation values of the satellite camera from the meta data of satellite images. We project the constructed 3D structure of buildings to the ground plane with the azimuth and elevation values of the satellite camera so that we can extract 2D information of buildings whose viewpoints are the same as satellite images, as shown in



(a) (b)

Fig. 10. Projection of constructed buildings and cameras: (a) constructed 3D model of buildings and relative pose of an aerial camera; and (b) image of buildings and a camera projected to the ground plane with azimuth and elevation of the satellite camera

Figure 10b. The triangle is the pose of an aerial camera and lines show the building information extracted from the aerial images.

IV. Building Matching Based on Nonlinear Optimization

The building information from the satellite images is extracted manually. We match two types of images based on extracted building information, such as a scale, position and orientation of buildings. We also define the cost-function considering errors in the 3D construction of buildings and minimize it to match the buildings extracted from aerial images and satellite images, using nonlinear optimization methods.

A. Cost function, considering 3D construction error

Once we find the 2D building information from aerial images and satellite images, matching problems can be translated to 2D–2D registration problems (i.e., all we need to estimate is a homograhpy between two scenes that contain a scale, a rotation and a translation value). We define the cost function to find a homography.

Although corresponding points between the aerial images have a subpixel error, we have to consider that construction of 3D buildings from aerial images can have erroneous results because we do not know the exact intrinsic parameters and because the distance between aerial cameras and buildings is too long. This means that triangulation results can have erroneous results.

If we assume that there is no skew in an aerial camera, then there are two important factors affecting 3D construction in intrinsic parameters: a focal length in terms of pixel dimension in the x and y direction and a principal point offset [7]. Problems in focal length in terms of a pixel dimension result in the 3D construction results having a different scale error in the x and y direction. A principal point offset error affects all the results of constructed buildings. However, if scene depth is too large, principal point error does not have much of an effect in constructing 3D models of buildings [7]. The distance between aerial cameras and buildings is so great that we can neglect the error of principal points.

If depth of the scene is too large, results of the triangulation method are very sensitive to the error of corresponding points, even though the corresponding error is in the subpixel range. We must consider triangulation error because the distance between the aerial camera and the buildings is extremely long in a forward-looking camera system. Therefore, we define the cost function, considering the factors that severely affect 3D construction, to estimate a homography:

$$H = \arg\min_{H} \sum_{i}^{M} \begin{matrix} \alpha_i \,|1 - S(H \cdot l_i)/S(l_i^{close})| \\ +\beta_i D(H \cdot l_i - l_i^{close}) \\ +\gamma_i \,|A(H \cdot l_i) - A(l_i^{close})| \end{matrix} \tag{5}$$

$$H = \begin{bmatrix} b_{11} & b_{12} & T_x \\ b_{21} & b_{22} & T_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \tag{7}$$

$$B = R(-\phi)\,\Lambda R(\phi) \tag{8}$$

$$R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \tag{9}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{10}$$

$$l_i = \begin{bmatrix} x_1^i & x_2^i \\ y_1^i & y_2^i \\ 1 & 1 \end{bmatrix}. \tag{11}$$

l_i is the extracted building lines from aerial images, where x_1^i and x_2^i are the x coordinates of lines at endpoints and y_1^i and y_2^i are the y coordinates of lines at endpoints. A homography consists of a scale (Λ),rotation (R) and translation (T_x , T_y) factors, as shown in Equation (6). ϕ is a relative angle of principal axis, and λ_1 and λ_2 are scale factors in the x and y direction, respectively. The scale factors complement error in focal length in the pixel range. M is the total number of the constructed buildings from aerial images. l_i^{close} is a boundary line of buildings manually extracted in satellite images and is the closest one to the i-th line of the buildings constructed from aerial images. We place the weight terms α_i, β_i and γ_i to be dependent on the depth of the i-th building so that the cost function compensates for the error of the triangulation method. α_i β_i , γ_i and the others are defined as follows:

$$\alpha_i = \beta_i = \gamma_i = \frac{1}{Z_i} \tag{12}$$

$$S(l_i) = \| (x_1^i, y_1^i) - (x_2^i, y_2^i) \| \tag{13}$$

$$D(l_i) = \| (x_1^i, y_1^i) \| + \| (x_2^i, y_2^i) \| \tag{14}$$

$$A(l_i) = \tan^{-1}\left(\frac{y_1^i - y_2^i}{x_1^i - x_2^i}\right). \tag{15}$$

Z_i is the depth of the i-th building. Function S is used to calculate the length of lines and function D calculates the distance between lines. Function A is used to compute the relative orientation angle of buildings. Equation (5) consists of three similarity measures of two types of buildings, which consider the scale, position and orientation of buildings. The first term in equation (5) decreases if the size of the buildings is similar to each other. The second and third terms also decrease if the position and orientation of buildings are analogous to each other.

B. *Nonlinear optimization*

Our optimization problem has six degrees of freedom. We cannot apply gradient methods to find the minimum value of the proposed cost function because it is not a convex. Therefore, we apply the particle filter method, which is an algorithm for sampling the probability distributions to find the global minimum of the cost function [12].

We initialize the particles according to the buildings, which are manually extracted from the satellite image (i.e., we construct the same number of particles as the number of buildings from a satellite image). Each particle is initialized by aligning the constructed building line that is closest to the aerial camera with a line of buildings in the satellite image. For each particle, we calculate a local minimum value of the cost with the gradient method. We then make new particles in the form of a Gaussian distribution, where the number of particles defined according to the cost and variation of the particles is dependent on the number of the iteration of the gradient method. We iteratively do this procedure several times. We do this iteration five times empirically. Minimum value of the cost function in this procedure determines final matching results. The uncertainty of in position estimation is defined as follow:

$$U = 1 - \frac{\#\text{of particles when generating minimum cost value}}{\text{total}\#\text{of particles}}. \quad (16)$$

V. Experiments and Analyses

We tested our proposed method with QuickBird satellite images, which are rectified with shuttle radar topography mission (SRTM) as a digital elevation model (DEM) and ground control points (GCP) for geo-referencing [13]. We applied our system to a set of aerial images that has 561 by 341 resolutions. Figure 11 shows the results of building detection. Rectangles in the image are the detection results of buildings from the aerial images. The proposed method detects the buildings robustly as long as the buildings satisfy the conditions.

A number of UAV localization papers proposed verification methods using GPS or IMU systems [3]. However, it is very difficult to obtain various sensor data simultaneously, and synchronization of the sensors is not easy. Therefore, we manually verified our methods. Figure 12a and 12c show automatically detected buildings and their labeled numbers for each building in aerial images. Rectangles in Figure 12b and 12d are the manually detected building data from satellite images, and numbers in the image are also the

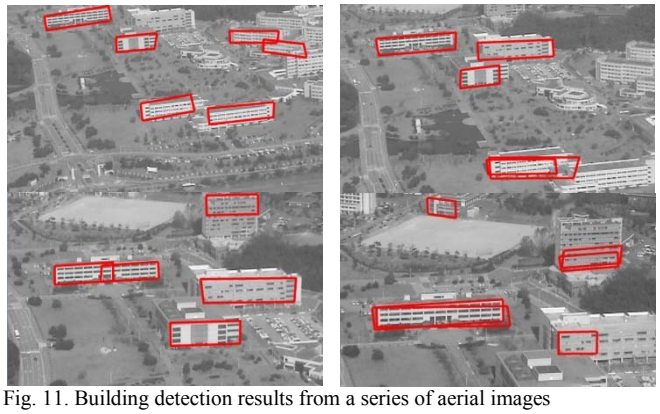


Fig. 11. Building detection results from a series of aerial images

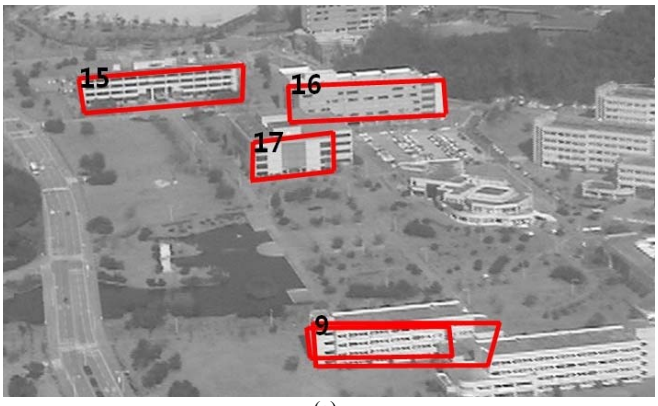
labeling numbers of each building. Buildings labeled as 9, 15, 16 and 17 in aerial images are matched with buildings labeled as 9, 15, 16, and 17 in the satellite image respectively, as shown in Figure12a and 12b. It is apparent that the automatically extracted buildings in an aerial image and the manually extracted buildings in a satellite image are well matched. Red circles in Figure 12b and 12d are the estimated pose of the camera and its uncertainties. The bigger circles mean the higher uncertainties.

We applied our algorithms to key frames that are selected when corresponding points are firstly below 150 between aerial images. Figure 13 shows the final estimated results of the camera poses and its uncertainties. Figure 13b–i shows a set of aerial images, with numbers in the circles of Figure 13j being corresponding camera poses. The closer the buildings are to the camera, the more the pose of the camera become stable because the triangulation error is reduced as the scene depth decreases. In Figure 13(a), uncertainties are decreased as the pose of the camera is close to the buildings except for the mismatching case in aerial image 7 (Fig. 13). If the building detection method does not work well because the columns of buildings are not detected, our algorithm fails, as shown in aerial image 7 (Fig. 13). Little information on the buildings from aerial images also results in method failure.

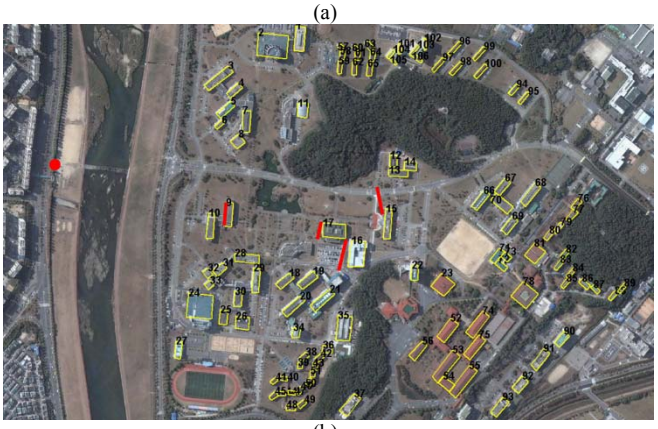
Our method compared with the GPS using other aerial images that has 320 by 240 resolutions, as shown in Figure 14. The UAV flied to the buildings that are detected in aerial images and collected key frames from 1 to 10 sequentially. Although buildings in satellite and aerial images are matched well, a proposed pose estimation method is unstable compared with GPS and has offset errors. This is mainly because the triangulation errors according to the scene depth. We can observe that the closer the UAV is to the buildings, the more the pose of the UAV become stable and also the errors are reduced.

VI. Conclusions

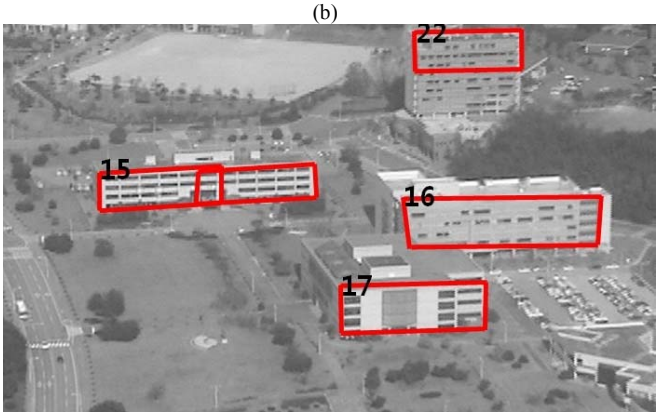
We designed a semantic matching algorithm between forward-looking aerial images and down-looking satellite images based on relational information of buildings to estimate a global pose of a UAV. We detect buildings in forward-looking aerial images by observing that the columns of buildings are normal to the ground plane and the buildings are part of man-made regions and have a repeated pattern.



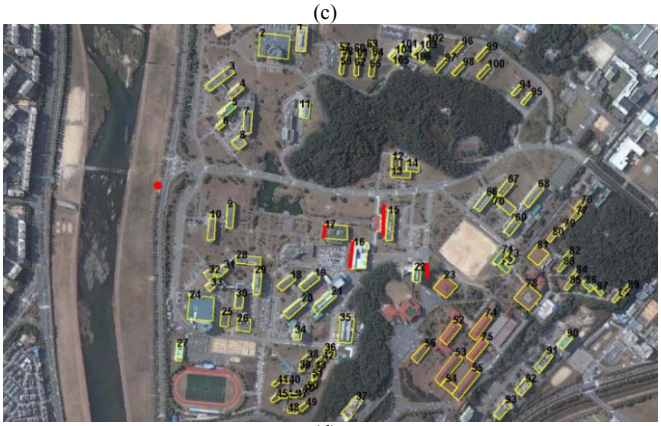
(a)



(b)



(c)



(d)

Fig. 12. Matching results with building information: (a) detected buildings in aerial scene 1; (b) matched results between buildings in aerial scene 1 and buildings in satellite images; (c) detected buildings in aerial scene 2; and (d) matched results between buildings in aerial scene 2 and buildings in satellite images.

Finally, we proposed a cost function to match the building information from aerial images and manually detected connections of buildings from satellite images, considering errors in 3D construction of aerial images. Our system works well with assumptions that the UAV flies around buildings and the ground plane is flat. Also, our system is not real time. Even though there are limitations to our algorithm, our proposed approach shows a promising solution to pose estimation of a UAV with only forward-looking camera data. For more accurate and robust pose estimation, we will carry out additional studies on how to combine road information with our current work.

REFERENCES

[1] Bruno Sinopoli, Mario Micheli, Gianluca Donato, T. John Koo, "Vision Based Navigation for an Unmanned Aerial Vehicle", *International Conference on Robotics and Automation*, 2001.

[2] Kumar. R., Sawhney. H. S., Asmuth. J. C. Pope. A. Hsu. S., "Registration of video to geo-referenced imagery", Fourteenth International Conference on, 1998.

[3] D.-G. Sim, R.-H. Park, R. –C. Kim, S. U. Lee, and I. –C. Kim, " Integrated position estimation using aerial image sequences.", IEEE Trans, Pattern Anal. Machine Intell., Vol. PAMI-24 no. 1, pp. 1-18, Jan.2002.

[4] Fernando Caballero, Luis Merino, Joaquin Ferruz and Anibal Ollero, "Improving Vision-based Planar Motion Estimation for Unmanned Aerial Vehicles through Online Mosaicing", *International Conference on Robotics and Automation*, 2006.

[5] Fernando Caballero, Luis Merino, Joaquin Ferruz and Anibal Ollero, "Homography Based Kalman Filter for Mosaic building. Application to UAV pose estimation", *International Conference on Robotics and Automation*, 2007.

[6] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features", *in Proceedings of the IEEE International Conference on Computer Vision,* 1999.

[7] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2004. Second Edition

[8] S. Kumar and Martial Hebert, "Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field", *in Proceedings of the IEEE International Conference on Computer Vision,* 2003.

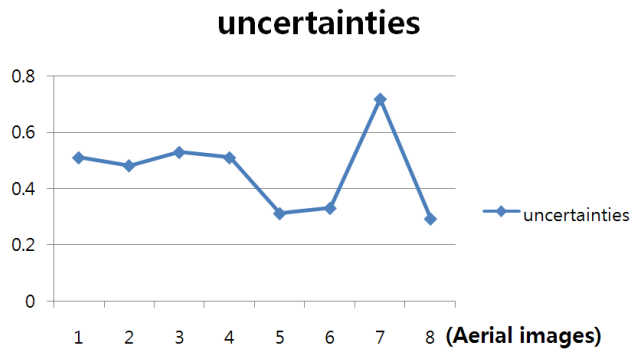
[9] J. Besag, "On the statistical analysis of dirty pictures", *Journal of the Royal Statistical Society,* 1986.

[10] Jianbo Shi, Carlo Tomasi, "Good features to track", *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, pp. 593 - 600.

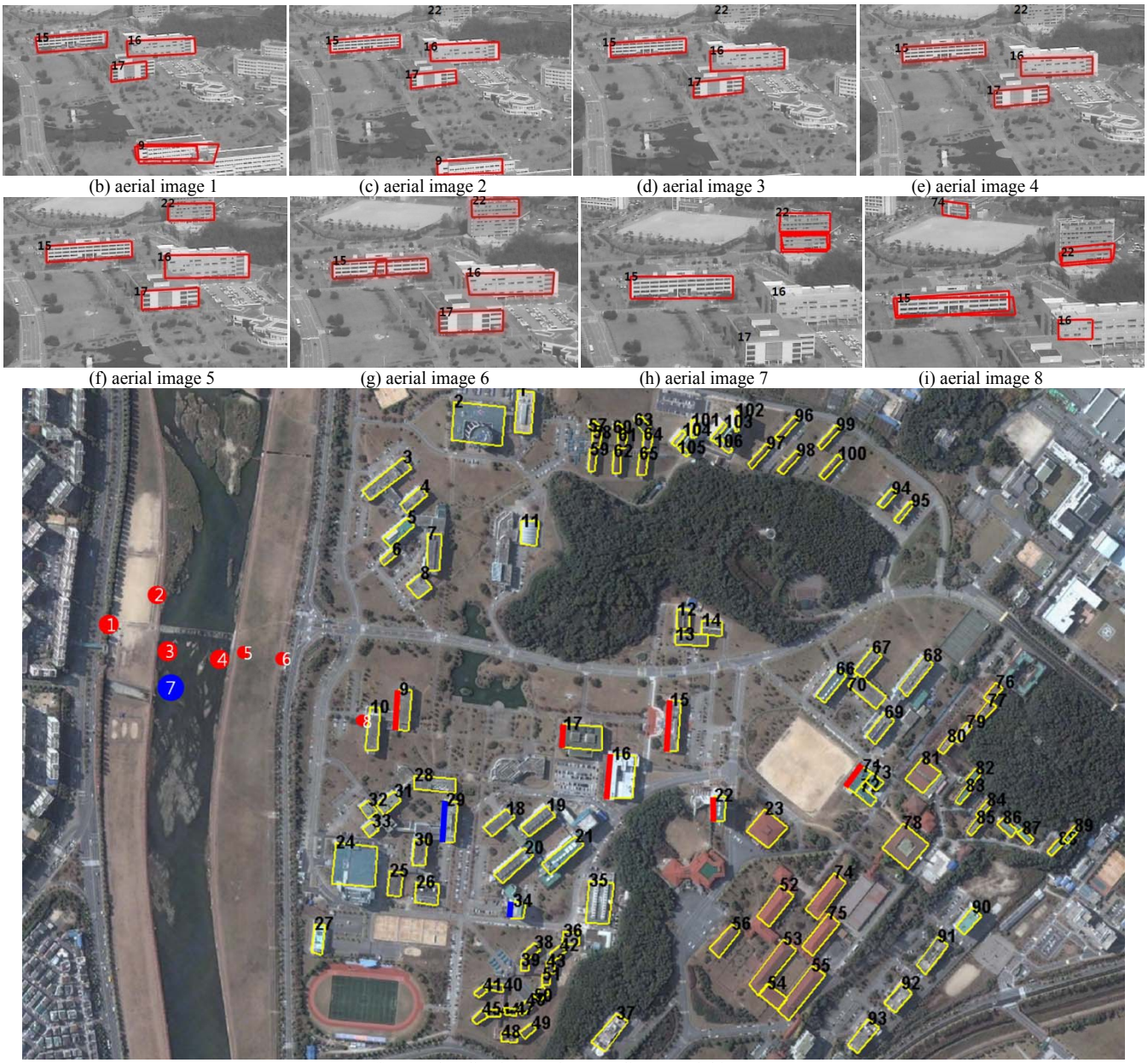
[11] B. Triggs. Autocalibration and the absolute quadric. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 609–614, 1997.

[12] J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, 2001

[13] http://en.wikipedia.org/wiki/QuickBird



(a) uncertainties of pose for each aerial image

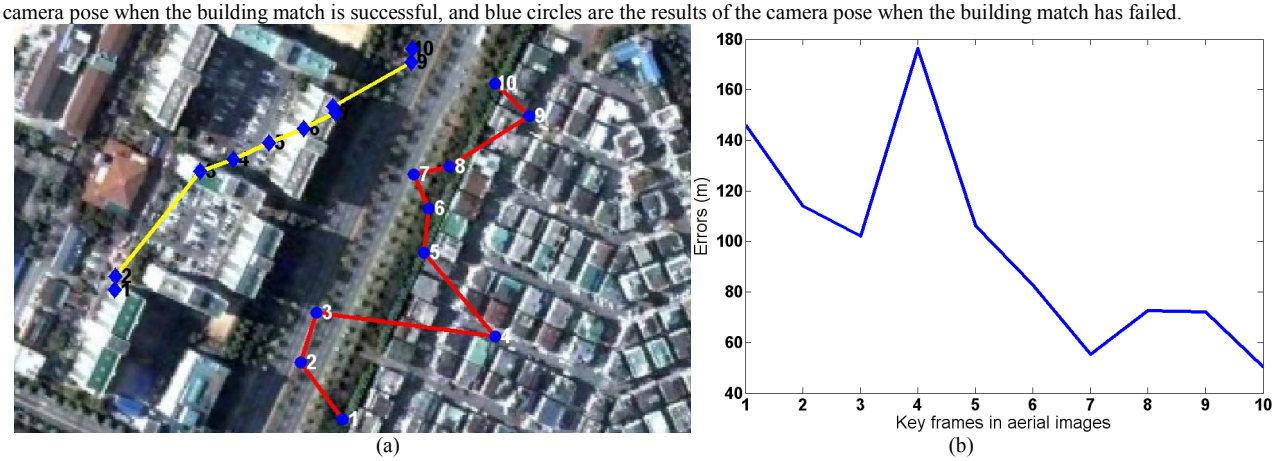


(b) aerial image 1 (c) aerial image 2 (d) aerial image 3 (e) aerial image 4

(f) aerial image 5 (g) aerial image 6 (h) aerial image 7 (i) aerial image 8

(j) UAV localization results according to aerial images

Fig. 13. Final results; (a) uncertainties of UAV pose for each aerial image; (b)~(i) red rectangles are detected buildings and numbers are building label numbers; (j) numbers in circles are the corresponding aerial image number and circles show the pose of the forward-looking aerial camera. Red circles are the results of the camera pose when the building match is successful, and blue circles are the results of the camera pose when the building match has failed.



(a) (b)

Fig. 14. GPS and proposed results; (a) yellow lines are GPS results and red lines are proposed results. Numbers are key frame number in aerial images; (b) errors between GPS and proposed results in Euclidian distance.