Journal of Aerospace Information Systems
Vol. 18, No. 12, December 2021

Particle Filter Approach to Vision-Based Navigation with Aerial Image Segmentation

Kyungwoo Hong,* Sungjoong Kim,* Junwoo Park,* and Hyochoong Bang†
Korea Advanced Institute of Science and Technology, Daejeon KS015, Republic of Korea

https://doi.org/10.2514/1.I010957

This study proposes a novel approach for a vision-based navigation problem using semantically segmented aerial images generated by a convolutional neural network. Vision-based navigation provides a position solution by matching an aerial image to a georeferenced database, and it has been increasingly studied for global navigation satellite system–denied environments. Aerial images include a vast amount of information that infers the position where they are located. However, it also includes features that disturb the estimation accuracy. The progress of convolutional neural network may provide a promising solution for extracting only helpful features for this purpose. Therefore, segmented images are modeled as a Gaussian mixture model, and the L_2 distance for a quantitative discrepancy between two images is established. This allows us to compare the two images quickly with improved accuracy. In addition, a framework of a particle filter is applied to estimate the position using an inertial navigation system. It employs the L_2 distance as a measurement, and the particles tend to converge to the true position. Flight test experiments were conducted to verify that the proposed approach achieved distance error of less than 10 m.

Nomenclature

H_{img}	=	height of the aerial image
i, j	=	Gaussian distributions index
i_{par}	=	particle index
N	=	number of total particles
N_P	=	number of Gaussian distributions of P
N_Q	=	number of Gaussian distributions of Q
$\hat{N}_{\mathrm{eff}}$	=	estimate of effective sample size
N_{thres}	=	threshold of effective sample size
P, Q	=	Gaussian mixture model
P_0	=	covariance of initial particles
p	=	probability density function of P
p_i	=	ith Gaussian distributions of P
q	=	probability density function of Q
q_i	=	ith Gaussian distributions of Q
u_k	=	relative movement at the kth time step
W_{img}	=	width of the aerial image
w_k	=	additive process noise at the kth time step
x_k	=	state vector at the kth time step
$x_k^{i_{\mathrm{par}}}$	=	state vector of the i_{par}th particle at the kth time step after correction
$\tilde{x}_k^{i_{\mathrm{par}}}$	=	state vector of the i_{par}th particle at the kth time step after propagation
z_k	=	measurement at the kth time step
α_i	=	proportional constant of p_i
β_i	=	proportional constant of q_i
μ	=	mean of Gaussian distributions
Σ	=	covariance of Gaussian distributions
σ_v	=	standard deviation of the likelihood function
σ_w	=	standard deviation of the process noise
ω_{sum}	=	sum of unnormalized weights
$\omega_k^{i_{\mathrm{par}}}$	=	unnormalized weight the i_{par}th particle at the kth time step
$\tilde{\omega}_k^{i_{\mathrm{par}}}$	=	unnormalized weight the i_{par}th particle at the kth time step

Subscripts

k	=	time index in a particle filter
i	=	ith Gaussian distributions

I. Introduction

Positioning systems of aerial vehicles usually consist of a global navigation satellite system (GNSS) and an inertial navigation system (INS) module. GNSS provides absolute positioning information with higher precision. However, GNSS suffers from vulnerabilities such as jamming and spoofing. In addition, GNSS signals can be blocked by taller buildings depending on the application and environment. INS errors should increase under a GNSS-denied environment over time. Therefore, navigation solutions to calibrate the INS in order to handle error sources are required.

Many studies have been conducted to overcome these vulnerabilities. In particular, navigation based on a database estimates the vehicle's position by matching an aerial image taken by an onboard camera with a georeferenced database during flight [1–3]. The georeferenced database is a set of orthogonal images that cover broad areas with location information such as latitude and longitude. It can be readily obtained from a geographic information system (GIS). The aerial image includes informative features to estimate the position of the aerial vehicle, but it also contains some features that disrupt the estimation performance. For example, seasonal variations and moving objects, such as cars, can result in differences between the aerial images taken and those of the database. Hence, comparing two images of the same scene taken at varying time instants using different sensors is a highly challenging issue in vision-based navigation problems.

Recently, owing to the development in artificial intelligence technology, the performance of classifying objects in images using a convolutional neural network (CNN) has received significant attention [4,5]. In navigation problems, the difference between aerial images and those of the database is critical because it involves seasonal variation or moving objects. The progress of CNN allows us to segment meaningful and time-invariant information, such as buildings, roads, and fields in aerial images. Hence, many approaches based on artificial intelligence for vision-based localization have been studied [6]. In particular, a preliminary version of this study was presented in a conference paper, and the simulation results showed a horizontal error of 10 m regardless of the terrain [7]. Nassar

Presented as Paper 2020-1344 at the AIAA Scitech 2020 Forum, Orlando, FL, January 6–10, 2020; received 13 January 2021; revision received 7 June 2021; accepted for publication 16 September 2021; published online 19 October 2021. Copyright © 2021 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 2327-3097 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Ph.D. Candidate, Department of Aerospace Engineering.

†Professor, Department of Aerospace Engineering. Senior Member AIAA.

et al. [8] proposed a deep-CNN-based registration with the semantic shape matching method for the localization of an aerial vehicle, and achieved a positioning error of only 10 m. In [9], the authors also adopted a CNN to learn the features useful for aligning satellite images and aerial images. In [10], the authors presented building ratio features to match the segmented aerial images to the database.

In this study, we sought to establish a method for vision-based navigation with semantic segmentation using a CNN. The computational time, time-varying features, and fusion with the INS should be considered for the integrated navigation solution. The fully convolutional network (FCN) [4] is applied to extract salient features that can be observed irrespective of the time of day and seasonal changes. Each detected segment is modeled as a Gaussian distribution because of the difficulties in directly identifying the similarities of pixels between the semantically segmented images. Thus, the problem can be reformulated as a matching problem for two Gaussian mixture models (GMMs). The similarity of two GMMs can be defined through the L_2 norm between GMMs, L_2 distance. This makes comparison significantly faster while reducing the effects of segmentation errors. Finally, we propose a particle filter framework for GMM, which is taken as the measurement. The particle filter is usually addressed for nonlinear estimation problems [11], and the particles are propagated using the output from the INS.

The remainder of this paper is organized as follows. First, we briefly address the problem of vision-based navigation and why machine learning is required to solve the problem in Sec. II. The proposed algorithm is provided along with a detailed description of the semantic segmentation approach, the formulation of the L_2 distance measure between GMMs, and the particle filter in Sec. III. In Sec. IV, the results of the flight test experiment are presented to validate the proposed approach. Finally, the conclusions are provided in Sec. V.

II. Problem Descriptions

A metric for measuring similarity between images is necessary to search for the location, where the aerial image matches the database. For example, the template-based matching method seeks to find the most similar place using the sum of absolute differences (SAD) or cross-correlation of the pixels [12]. However, these methods are time-consuming when the size of the database is large and vulnerable to changes in scale and rotation. For feature-based matching, SIFT [13] is one of the most popular descriptors with its improved versions such as SURF [14], ORB [15]. Such methods seek to find a global affine transformation with feature points extracted from both images using a brute-force matcher or FLANN [16] matcher. The matching tools are based on distance measurements such as the Euclidean or Hamming distances of the descriptors as the similarity measures. However, these methods also suffer from issues in extracting the same features between the two images, making them difficult to apply to navigation. Figure 1 shows the matching results obtained using conventional feature-based matching methods: SURF and ORB. The aerial image on the left was obtained by an aerial vehicle, whereas the right-hand side is the database from GIS data, such as Google Maps or Open Street Map (OSM). In addition, it includes the area shown in the aerial images, which are marked with rectangles. As shown in Fig. 1, the match-rate performance did not meet the desired level. Because the time instances for flight and that of creating the database inevitably differ, both methods are not guaranteed to extract the same features from the two images.

In this study, we focus on two significant issues to be considered for navigation. The first issue is that the two images to be matched are taken from different viewpoints at different times by nonhomogeneous sensors. That is, the time instances to create a database and to take pictures are different. These images show the scene at the same location, but do not include the same features. For example, an aerial vehicle is hovering in one place, and the time-varying elements (e.g., cars and people) could make the aerial images look different. The aerial image can include moving objects that are not included in the database, which can contribute to inaccuracy. In addition, the sources of the images were not identical. The aerial images can be obtained by an aerial vehicle, but the database images usually consist of satellite images. Therefore, such a discrepancy

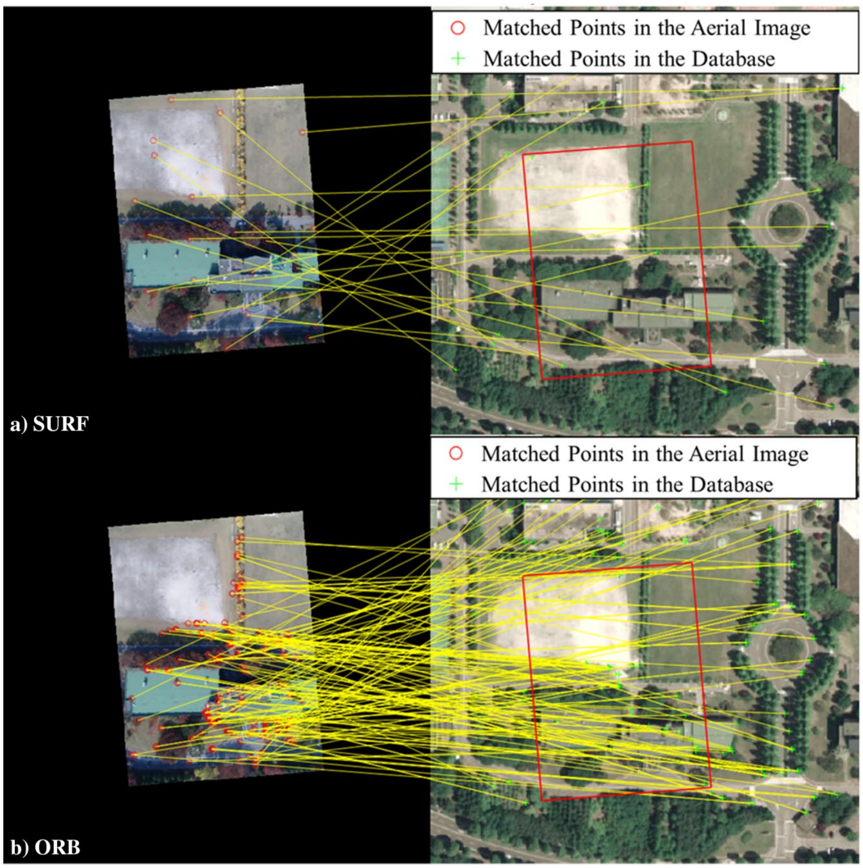
○ Matched Points in the Aerial Image
+ Matched Points in the Database
a) SURF
○ Matched Points in the Aerial Image
+ Matched Points in the Database
b) ORB

Fig. 1 Matching between an aerial image from an aerial vehicle and the database from satellite images using conventional feature matching methods.

would cause a failure in matching. Moreover, the matching should be based on high-level features such as object detection and classification rather than low-level features such as edge and orientated gradients. The second issue is the speed of the algorithm. The solutions should be obtained in real time because time-consuming methods such as template matching may not meet the requirements of control of the vehicle.

For the first reason, we chose image segmentation using the CNN technique, to obtain the image description in order to detect the time-invariant features. Nevertheless, the method of matching the two segmented images remains unsolved. Because the segmented image contains many errors, it is difficult to use existing matching techniques. Figure 2 presents examples of aerial images and their segmentation results. One can find some errors in recognizing non-buildings as a building and vice versa. In particular when small buildings were densely distributed, the results could be worse. Of course, conventional feature-based matching methods cannot detect the same feature owing to errors. Therefore, to use a CNN to detect time-invariant features, a new matching method for segmented images needs to be considered.

III. Technical Approach for the Proposed Method

A. Feature Extraction

As addressed in Sec. II, the most crucial point for navigation lies in the difference between the database and the aerial images caused by the image sources and the time instance taken for the pictures. To overcome these problems, a fully convolutional network (FCN) [4] has been used, where useful features are extracted in a time-varying situation. FCN is considered as one of the most popular and widely used methods for solving semantic segmentation problems. It relies upon the so-called upsampling to obtain a dense heat map, and its performance has been verified in many applications.

Training a network usually takes a long time and requires plenty of training data. The network can be trained offline. Therefore, the training time should not be a critical issue for navigation. However, the amount of training data is related to performance. Because FCN is a supervised learning technique, a reference answer image is necessary. The answer image was used to mark the features to be extracted. Because acquiring vast aerial images through a real flight and labeling the features by hand requires significant effort, GIS data were used for this study. GIS includes geographic information such as buildings, roads, and fields, along with orthogonal aerial images. Thus, it is used to generate training data efficiently.

The FCN network was trained using Caffe API, and approximately 12,000 training images were involved. First, we referred to the network pretrained using the PASCAL VOC 2012 training data. Hereafter, the last fully connected layer of the network was modified for the number of outputs. Finally, the network was trained using a dataset in the form of aerial images from the GIS. Figure 2 presents examples of an FCN. Note that the aerial images are acquired through real flight, and the network is trained using the images from the GIS data. Although the source of the train images and the test images are not the same, the buildings are well recognized. However, this is not considered a sophisticated method. In particular, at the edges of the building, the accuracy is unsatisfactory. Therefore, the proposed method can be considered to reduce the effects of such errors.

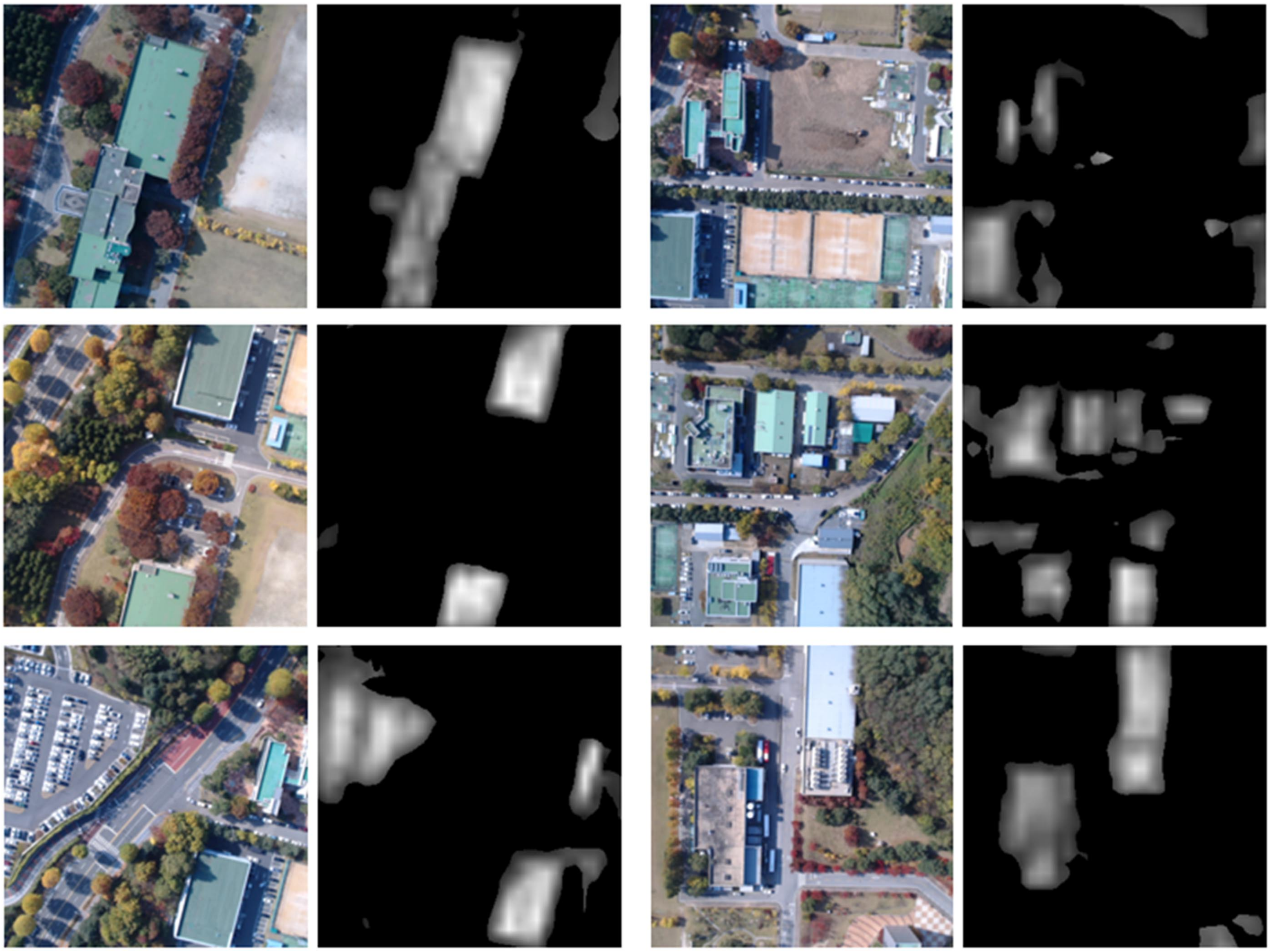

Fig. 2 Examples of the images from an aerial vehicle during flight and the semantic segmentation of the aerial images for which buildings are features to detect.

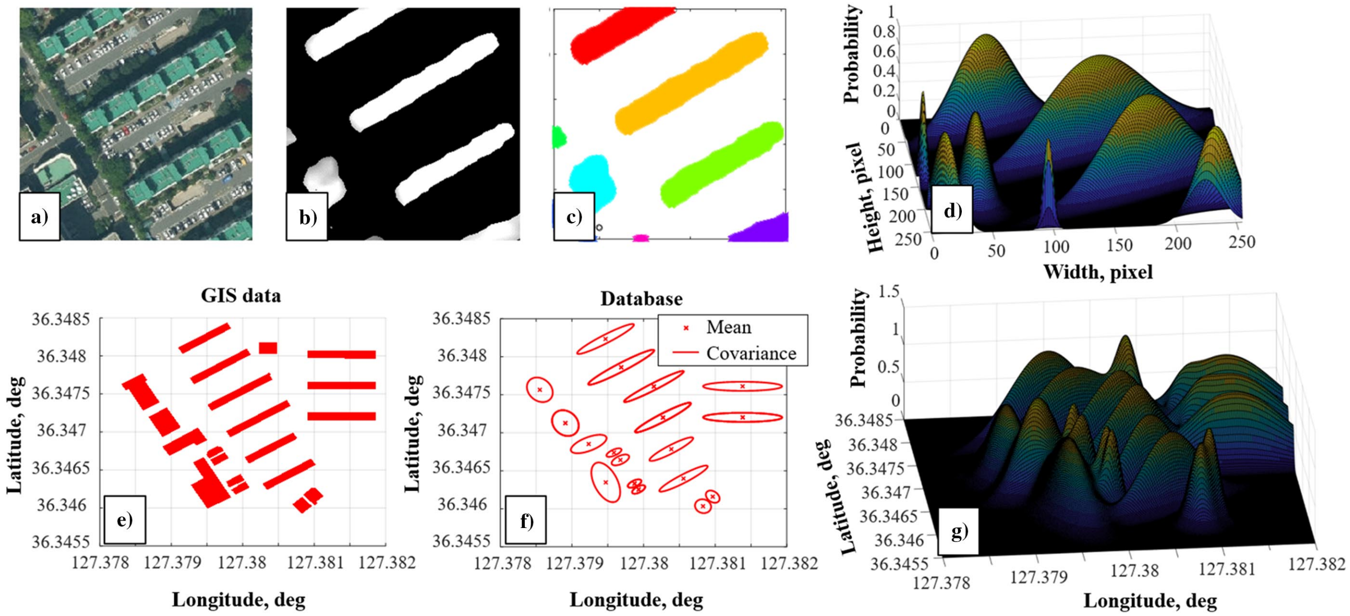

Fig. 3 Process of generating GMM from aerial images (a–d) and from the database (e–g).

B. Similarity Between an Aerial Image and a Database

To match two images, one should be able to compare the two images quantitatively. That is, numerical criteria should be established to assess the similarity between the segmented aerial images and those of the database. For this goal, the matching problem is defined by assuming that the segmented image is a two-dimensional (2D) GMM. The extracted features are converted to a Gaussian distribution. Therefore, the images should contain multiple Gaussian distributions. The parameters of the Gaussian distribution are represented in terms of the mean and covariance. The mean value is the center position of the feature, and the covariance is proportional to the size of the features. This approach offers two advantages. First, the effects of the errors of the FCN can be adjusted. The central area of the object is accurately recognized, but its accuracy significantly degrades as it approaches the edge of the object. Modeling it as a GMM can help in handling errors. For example, the false-negative cases, which usually occur on the edge of an object, do not affect the results because the Gaussian distribution exhibits the highest probability at the center. In the case of false positives, it also does not have much effect because the covariance is smaller than the true-positive result. Therefore, it is an effective method that considers the characteristics of FCN errors. Second, the L2 distance between the two GMMs can be introduced quickly because there is an analytical solution. The L_2 distance for the similarity of the two GMMs can be adopted between the aerial and database images. Therefore, the image matching problem is reconstructed into a problem by comparing the two GMMs.

Given two 2D GMMs, denoted as P and Q, the L_2 distance between P and Q is defined as follows:

$$L_2(P,Q)=\int_0^{H_{\mathrm{img}}}\int_0^{W_{\mathrm{img}}}(p(x,y)-q(x,y))^2\,\mathrm{d}x\,\mathrm{d}y \\ =\int_0^{H_{\mathrm{img}}}\int_0^{W_{\mathrm{img}}}\left(\sum_{i=1}^{N_P}\alpha_i p_i(x,y)-\sum_{i=1}^{N_Q}\beta_i q_i(x,y)\right)^2\mathrm{d}x\,\mathrm{d}y \qquad (1)$$

where $p(x,y)$ and $q(x,y)$ are the probability density functions of P and Q, respectively. The W_{img} and H_{img} correspond to the width and height of the aerial image, respectively. P and Q consist of the N_P and N_Q Gaussian distributions, $p_i(x,y)$ and $q_i(x,y)$ with proportional constants, α_i and β_i, respectively. The proportional constant makes the probability of the centers of each distribution equal, so that Eq. (1) becomes

$$L_2(P,Q)=\sum_i^{N_P}\sum_j^{N_P}\alpha_i\alpha_j\int_0^{H_{\mathrm{img}}}\int_0^{W_{\mathrm{img}}}p_i p_j\,\mathrm{d}x\,\mathrm{d}y \\ -2\sum_i^{N_P}\sum_j^{N_Q}\alpha_i\beta_j\int_0^{H_{\mathrm{img}}}\int_0^{W_{\mathrm{img}}}p_i q_j\,\mathrm{d}x\,\mathrm{d}y \\ +\sum_i^{N_Q}\sum_j^{N_Q}\beta_i\beta_j\int_0^{H_{\mathrm{img}}}\int_0^{W_{\mathrm{img}}}q_i q_j\,\mathrm{d}x\,\mathrm{d}y \qquad (2)$$

As shown in Eq. (2), the product of each GMM mode is involved in the numerical evaluation of the L_2 distance. Naïve computation over the 2D image domain requires an excessive amount of computation. Fortunately, an analytical solution for the integral exists [17]. Therefore, we do not have to consider the integral, and it allows accelerating the computation of L_2 distance.

$$\int_0^{H_{\mathrm{img}}}\int_0^{W_{\mathrm{img}}}p_i(\mu_{p_i},\Sigma_{p_i})q_j(\mu_{q_j},\Sigma_{q_j})\,\mathrm{d}x\,\mathrm{d}y \\ =\frac{\exp(-0.5(\mu_{p_i}-\mu_{q_j})^T(\Sigma_{p_i}+\Sigma_{q_j})^{-1}(\mu_{p_i}-\mu_{q_j}))}{\sqrt{|2\pi(\Sigma_{p_i}+\Sigma_{q_j})|}} \qquad (3)$$

where μ and Σ represent the mean and covariance of the Gaussian distribution, respectively. Because the 2D plane is under consideration, μ is 2×1, and Σ is 2×2 matrix. Finally, the L_2 distance was formulated as follows:

$$L_2(P,Q) \\ =\sum_i^{N_P}\sum_j^{N_P}\alpha_i\alpha_j\frac{\exp(-0.5(\mu_{p_i}-\mu_{p_j})^T(\Sigma_{p_i}+\Sigma_{p_j})^{-1}(\mu_{p_i}-\mu_{p_j}))}{\sqrt{|2\pi(\Sigma_{p_i}+\Sigma_{p_j})|}} \\ -2\sum_i^{N_P}\sum_j^{N_Q}\alpha_i\beta_j\frac{\exp(-0.5(\mu_{p_i}-\mu_{q_j})^T(\Sigma_{p_i}+\Sigma_{q_j})^{-1}(\mu_{p_i}-\mu_{q_j}))}{\sqrt{|2\pi(\Sigma_{p_i}+\Sigma_{q_j})|}} \\ +\sum_i^{N_Q}\sum_j^{N_Q}\beta_i\beta_j\frac{\exp(-0.5(\mu_{q_i}-\mu_{q_j})^T(\Sigma_{q_i}+\Sigma_{q_j})^{-1}(\mu_{q_i}-\mu_{q_j}))}{\sqrt{|2\pi(\Sigma_{q_i}+\Sigma_{q_j})|}} \qquad (4)$$

C. Particle Filter with Image Measurements

In this subsection, a brief introduction to the particle filter is presented along with the image measurements. The particle filter approach is employed to fuse the INS and image measurements to take advantage of the handling of nonlinear systems. In addition, the particle distribution can properly address the INS information through the process model of the filter. Therefore, it may be possible to search a database efficiently, rather than comparing it globally or randomly.

The process model is expressed as

$$x_{k+1} = x_k + u_k + w_k \tag{5}$$

where x_k is a state vector that consists of the latitude and longitude at time step k, and u_k and w_k are the relative movement and additive process noise, respectively. The additive process noise is assumed to be a zero-mean normal distribution: $w_k \sim \mathcal{N}(0, \sigma_w^2)$, for which the grade of INS is usually characterized. An INS provides estimates of relative movement. This dynamic equation can specify the predictive conditional transition density $p(x_k|x_{k-1}^{i_{\text{par}}})$, by which particles are drawn. This equation is a simple form of the process model, and this simple model is considered realistic without the detail of INS integration if an independent attitude solution is available [18]. Instead of addressing the INS further, we focused on the correction step using image measurements.

The particle filter measurements correspond to the L_2 distance between the two GMMs from the aerial and database images, as mentioned above. The aerial images were transformed into what can be used as measurements. Figure 3 shows the process of obtaining the L_2 distance using aerial images. For instance, (a) is an example of an aerial image obtained by an aerial vehicle. Hereafter, the features can be extracted by the FCN network as shown in (b). Because it is not classified from each feature, the so-called clustering method, DBSCAN [19], is used to group that information. The reason for using DBSCAN is that it does not require the number of clusters. The clustering results are presented in (c). Finally, the GMM of the aerial images shown in (d) can be created by extracting the center and covariance from the DBSCAN results. Furthermore, each particle receives the database information based on its position. Given the database received (e), the center and covariance can be calculated as (f). Then, the GMM of each particle shown in (g) can be constructed. Eventually, the L_2 distance between the two GMMs is obtained from (d) and (g) using Eq. (4).

Algorithm 1: Particle filter with image measurements

```
- FOR $i_{\text{par}} = 1{:}N$
    Draw $\tilde{x}_k^{i_{\text{par}}} \sim p(x_k|x_{k-1}^{i_{\text{par}}})$
    Calculate $\tilde{\omega}_k^{i_{\text{par}}} = p(z_k|\tilde{x}_k^{i_{\text{par}}})$
- Calculate the sum of weights $\omega_{\text{sum}} = \sum_{i_{\text{par}}=1}^{N} \tilde{\omega}_k^{i_{\text{par}}}$
- FOR $i_{\text{par}} = 1{:}N$
    Normalize $\omega_k^{i_{\text{par}}} = \frac{\tilde{\omega}_k^{i_{\text{par}}}}{\omega_{\text{sum}}}$
- IF $\hat{N}_{\text{eff}} < N_{\text{thres}}$
    $\left(\left\{x_k^{i_{\text{par}}}\right\}_{i_{\text{par}}=1}^{N}, \left\{\omega_k^{i_{\text{par}}}\right\}_{i_{\text{par}}=1}^{N}\right) =$
Resampling$\left(\left\{\tilde{x}_k^{i_{\text{par}}}\right\}_{i_{\text{par}}=1}^{N}, \left\{\omega_k^{i_{\text{par}}}\right\}_{i_{\text{par}}=1}^{N}\right)$
- ELSE
    $\left\{x_k^{i_{\text{par}}}\right\}_{i_{\text{par}}=1}^{N} = \left\{\tilde{x}_k^{i_{\text{par}}}\right\}_{i_{\text{par}}=1}^{N}$
```

The particle weights are updated using the likelihood function. The L_2 distance results in a small size when the two GMMs are similar and become large when two GMMs are significantly separated. Based on this property, it is possible to determine the similarity between the two GMMs. In other words, one can see how close each particle is to the true position. Therefore, the likelihood function is modeled as a normal distribution such that

$$p(z_k|\tilde{x}_k^{i_{\text{par}}}) = \frac{1}{\sigma_v\sqrt{2\pi}}\exp\left(-\frac{L_2\text{distance}^2}{2\sigma_v}\right) \tag{6}$$

where z_k and $\tilde{x}_k^{i_{\text{par}}}$ are the measurement and i_{par}th predicted particle, respectively, and σ_v denotes the standard deviation of the likelihood function.

By repeating the propagation and correction process, a small number of particles would dominate the weights, reducing the filter performance, which is known to be a particle degeneracy problem. The resampling process is used to solve the degeneracy problem. In other words, it converts the high-weighted particles into normal-weighted multiparticles and eliminates the low-weighted particles. The effective sample size can be estimated, and resampling is performed when the estimate of the effective sample size is smaller than a threshold. A multinomial resampling method was used for the proposed algorithm [20].

A pseudocode description of the particle filter is presented in Algorithm 1. The multinomial resampling method is denoted as *Resampling* in Algorithm 1.

The entire navigation system is constructed by extracting time-invariant features, conversion to GMM, and data fusion with the particle filter. The camera onboard an aerial vehicle takes aerial images, and these images are transformed into GMMs using the trained network and DBSCAN. The particles are propagated by the relative movement obtained by the INS and updated in the form of the L_2 distance. A schematic diagram of the navigation system architecture is shown in Fig. 4.

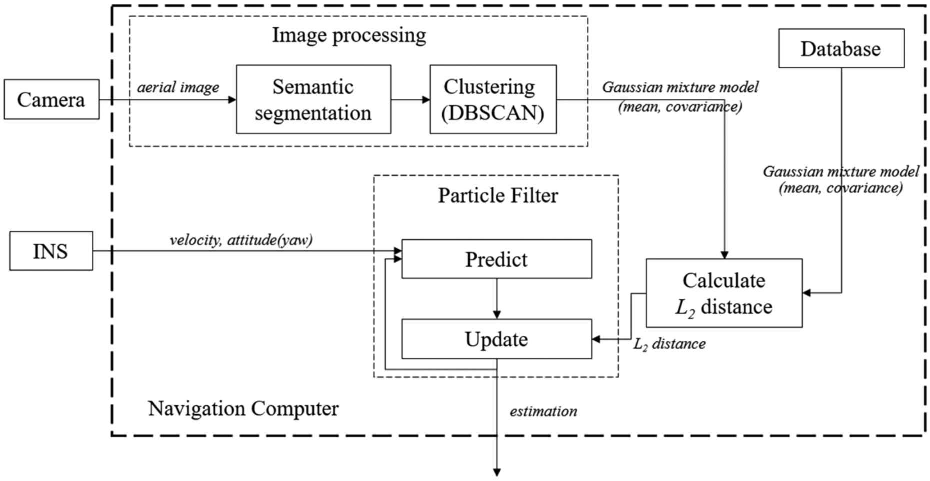

Fig. 4 Proposed navigation system.

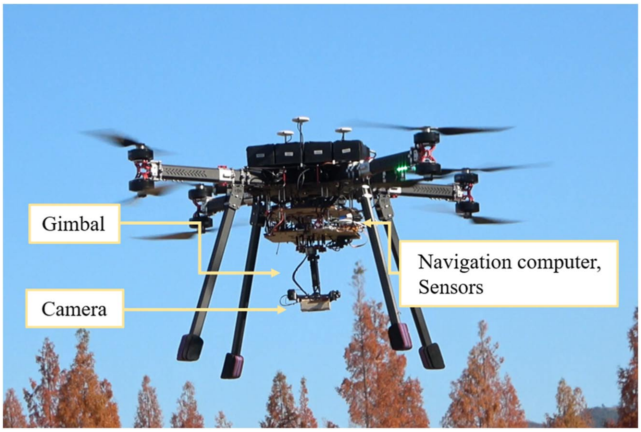

Fig. 5 The multirotor vehicle used in the flight experiment.

IV. Experiment by Flight Test

This section presents the setup of the flight test experiment and the navigation results through an experiment to demonstrate the performance of the proposed method. First, the flight test experimental conditions were overviewed. Then, the estimation results are assessed in terms of the position errors, which implies that the effectiveness of the proposed method is satisfactory.

A. Flight Test Experiment Conditions

Figure 5 shows the scene of the flight test experiment and its own multirotor UAV with a camera, gimbal, sensors, and a navigation computer. The camera was a Logitech C920 attached to the gimbal to obtain the stable aerial images. It is set to point downward to maintain the attitude of the camera, regardless of the attitude of the multirotor vehicle. The gimbal was mounted at the bottom of the main frame. The sensors and navigation computer were placed on the gimbal.

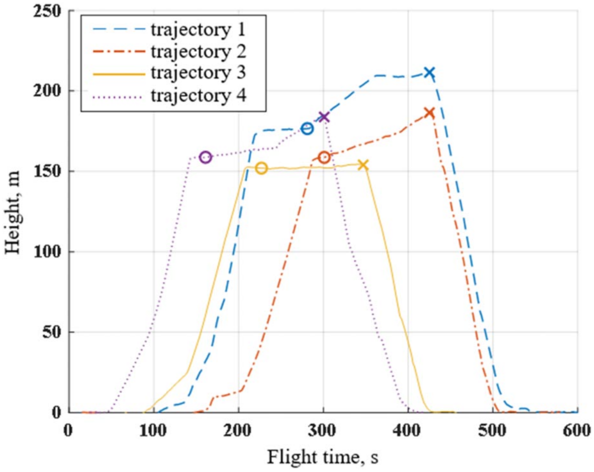

Fig. 6 Time history of the height of the multirotor vehicle during the flight.

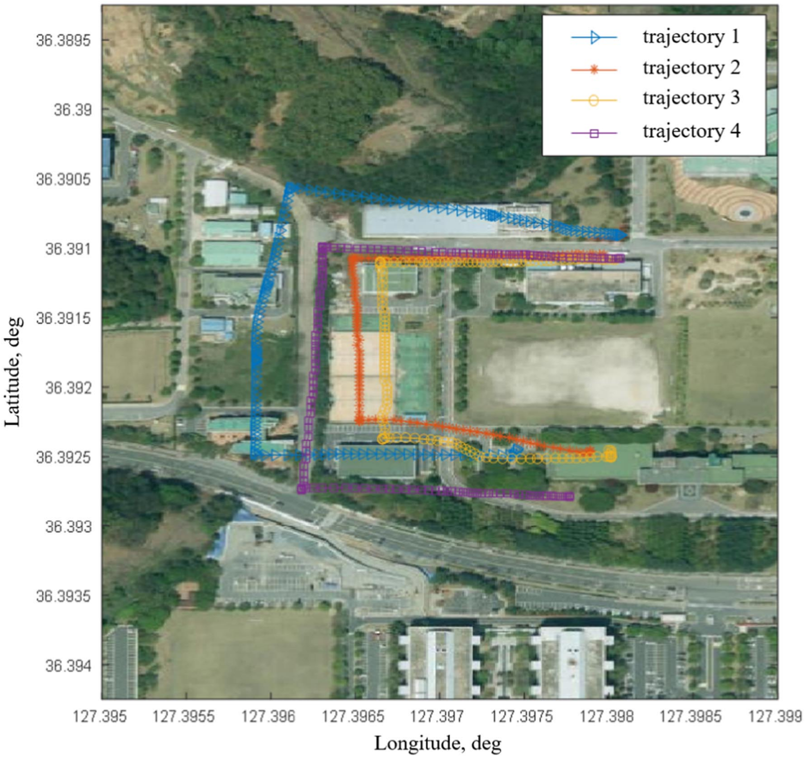

Fig. 7 Flight trajectory with the GIS image.

An NVIDIA Jetson Xavier board was implemented as a navigation computer with an 8-core ARM v8.2 CPU and 512-core Volta GPU. The particle filter works fully in real time on the navigation computer. In addition, two types of sensors were used. The first is an INS sensor for the proposed algorithm. The second is an integrated RTK-GNSS/INS sensor to produce the reference position information. It does not affect the algorithm, but is used in processing the position error. The sensor for INS is VectorNav VN200 with its horizontal position accuracy about 2.5 m. However, it is a GNSS/INS sensor, so INS is assumed as a GNSS/INS sensor with additive noise of 5 m sigma position error. The reference sensor is the Advanced Navigation Spatial Dual, and its horizontal position accuracy with RTK was 0.008 m. The database and network were uploaded to an onboard memory device before flight.

The flight test experiment was conducted four times. To capture a broad range of scenes, the multirotor vehicle is commanded to fly at an altitude of approximately 150 m above ground level (AGL). Then, the navigation filter is manually activated when it reaches a sufficient altitude. Figure 6 shows the time history of the altitude during flight from the reference sensor. The circle and cross marks in the figure denote the beginning and end of the navigation filter, respectively. Figure 7 shows the GIS image with overlaid flight trajectories from the reference sensor, for which the flight distance was approximately 600 m.

The initial state of the particles was set to the GNSS/INS output when the filter was activated. The initial particles were set to a covariance of 5 m. The standard deviation of the longitudinal and latitudinal process noise was 5 m. The standard deviation of the likelihood function, σ_v in Eq. (6), is 3 m. The number of particles was chosen to be 500. The update frequency of the particle filter was 1 Hz and the parameters used in the particle filter are listed in Table 1.

Table 1 Parameters used in the flight test experiment

Parameter	Value
P_0, m	$\begin{bmatrix}5 & 0\\0 & 5\end{bmatrix}$
σ_w, m	5
σ_v, m	3
N	500
N_{thres}	167

B. Experimental Results

The time histories of the longitudinal and latitudinal errors with their 3-sigma bounds are shown in Fig. 8. The sky and orange colors denote the results without correction and of the proposed method, respectively. Errors are indicated by circles and stars. The 3-sigma line is represented by a dashed line and dotted line. The results without correction indicate that the filter did not receive image measurements. In other words, the particles are only propagated using the relative movement from the INS without correction using measurements such as GNSS and images. Owing to the accumulation of time, the results without correction indicate that the estimator does not converge. Note that an INS is implemented by adding an error with an average of 5 m to GNSS/INS, a rather harsh condition. The reason for this assumption is to confirm the outcome of position correction using image measurements. Despite the harsh conditions, it was observed that the estimation error of all trajectories using the proposed method converged.

Figure 9 shows the results of the proposed method for the flight trajectory. The solid line represents the ground truth trajectory obtained from the RTK GNSS, and the line with circles represents the estimated trajectory. In a 2-D plane, the lines with circles of all trajectories converge to the true trajectories, maintaining a certain level of error. In other words, the errors did not diverge over a given time span. In contrast, the trajectory without correction is not included because it is outside the bounds of the graph scale.

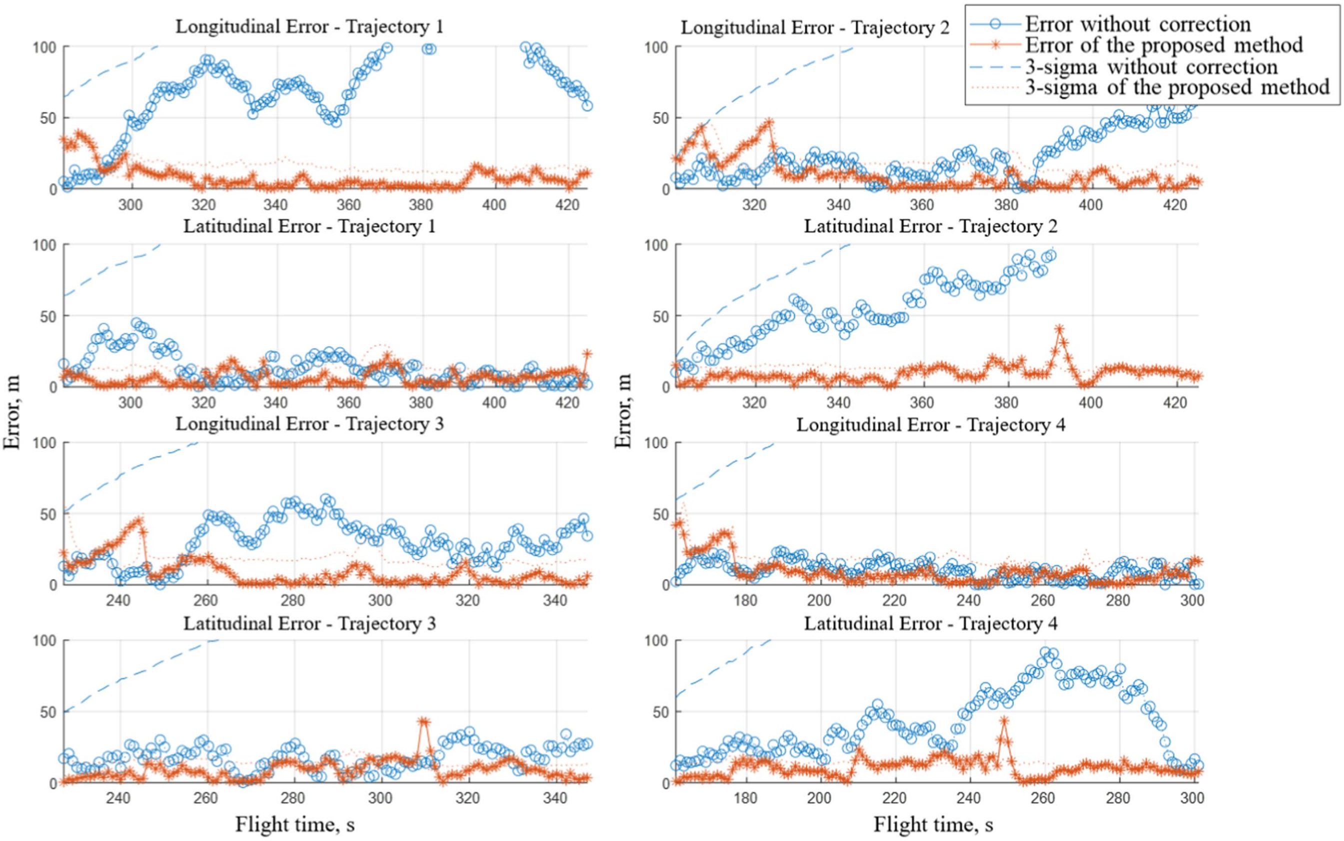

Fig. 8 Time history of the longitudinal and latitudinal error with 3-sigma of each trajectory.

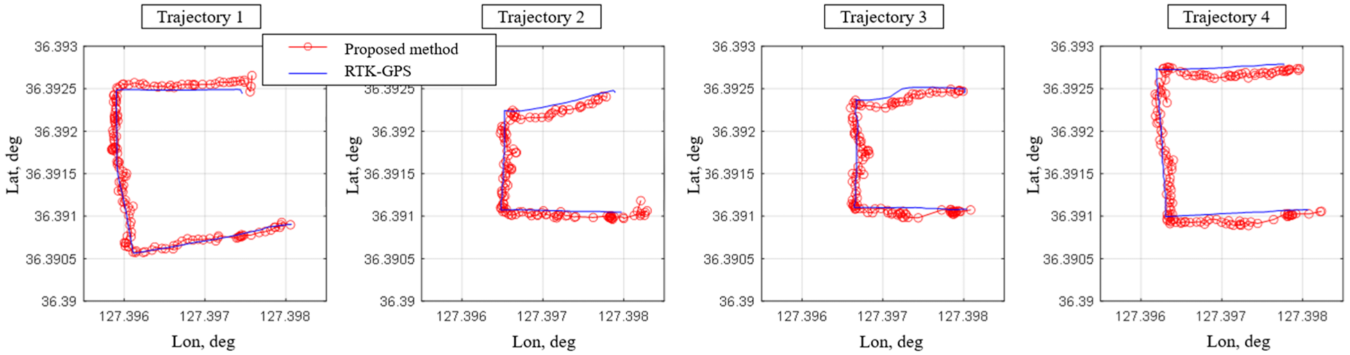

Fig. 9 Flight trajectory and the results of the proposed method.

Table 2 RMSE of longitudinal and latitudinal errors of each trajectory

Trajectory	No correction				Correction			
Error	1	2	3	4	1	2	3	4
Longitudinal, m	75.1	22.6	29.9	10.1	7.9	10.2	9.0	9.1
Latitudinal, m	12.8	78.3	17.4	42.4	6.1	9.2	9.0	10.4

The root mean square errors (RMSEs) of each trajectory are listed in Table 2. The results without correction of the trajectory 1, denoted as “No correction,” result in significantly large errors of 75.1 and 12.8 m, respectively. Although the same INS measurements were taken, the latitudinal and longitudinal errors from image measurements, L_2 distance, are 6.1 and 7.9 m, respectively. In addition, all errors of each trajectory using the proposed method were within 10 m.

These errors are believed to include FCN and gimbal errors. Gimbal errors imply that the camera does not look downward precisely enough due to disturbances during flight. The proposed method assumes that the roll and pitch angles are zero. That is, the center of the image and the horizontal position of the vehicle are the same. However, if the roll angle is not zero, a horizontal position error occurs, proportional to the elevation and tangent of the roll angle. Improving the system error sources with inflight calibration can enhance estimation performance.

V. Conclusions

In this study, an algorithm for dealing with semantically segmented images in vision-based navigation was proposed and verified by a flight test experiment. A novel method to match real segmented images to those of a database was validated to solve the critical issues dealing with images in different domains. Segmented images are modeled as GMMs, for which new quantitative criteria to compare two images are designed as L_2 distance measures. This allows a computationally efficient comparison between the two images. A particle filter using the L_2 distance as a measurement is also proposed to determine the position in conjunction with the INS output containing inherent errors. A flight test experiment was performed to verify the performance of the proposed method. Compared with the RTK GNSS output used as reference data, the proposed idea turned out to provide converged navigation solutions within 10 m position errors under a fairly harsh experimental condition. The authors believe that the proposed method could be an attractive option for the navigation problem of aerial vehicles in GNSS-denied environment.

References

[1] Xu, Y., Pan, L., Du, C., Li, J., Jing, N., and Wu, J., “Vision-Based UAVs Aerial Image Localization: A Survey,” *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, Assoc. for Computing Machinery, New York, Nov. 2018, pp. 9–18.
https://doi.org/10.1145/3281548.3281556

[2] Conte, G., and Doherty, P., “Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information,” *EURASIP Journal on Advances in Signal Processing*, Vol. 2009, Jan. 2009, pp. 1–18.
https://doi.org/10.1155/2009/387308

[3] Koch, T., Zhuo, X., Reinartz, P., and Fraundorfer, F., “A New Paradigm for Matching UAV and Aerial Images,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 3, July 2016, pp. 83–90.
https://doi.org/10.5194/isprs-annals-III-3-83-2016

[4] Long, J., Shelhamer, E., and Darrell, T., “Fully Convolutional Networks for Semantic Segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, New York, 2015, pp. 3431–3440.

[5] He, K., Gkioxari, G., Dollár, P., and Girshick, R., “Mask R-CNN,” *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, New York, 2017, pp. 2961–2969.

[6] Tian, Y., Chen, C., and Shah, M., “Cross-View Image Matching for Geo-Localization in Urban Environments,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, New York, 2017, pp. 3608–3616.

[7] Hong, K., Kim, S., and Bang, H., “Vision-Based Navigation Using Gaussian Mixture Model of Terrain Features,” *AIAA Scitech 2020 Forum*, AIAA Paper 2020-1344, Jan. 2020.
https://doi.org/10.2514/6.2020-1344

[8] Nassar, A., Amer, K., ElHakim, R., and ElHelw, M., “A Deep CNN-Based Framework for Enhanced Aerial Imagery Registration with Applications to UAV Geolocalization,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, New York, 2018, pp. 1513–1523.

[9] Goforth, H., and Lucey, S., “GPS-Denied UAV Localization Using Pre-Existing Satellite Imagery,” *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, New York, May 2019, pp. 2974–2980.

[10] Choi, J., and Myung, H., “BRM Localization: UAV Localization in GNSS-Denied Environments Based on Matching of Numerical Map and UAV Images,” arXiv preprint arXiv:2008.01347.

[11] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, Aug. 2002, pp. 174–188.
https://doi.org/10.1109/78.978374

[12] Bhat, D. N., and Nayar, S. K., “Ordinal Measures for Image Correspondence,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 4, April 1998, pp. 415–423.
https://doi.org/10.1109/34.677275

[13] Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, Vol. 60, No. 2, Nov. 2004, pp. 91–110.
https://doi.org/10.1023/B:VISI.0000029664.99615.94

[14] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding*, Vol. 110, No. 3, June 2008, pp. 346–359.
https://doi.org/10.1016/j.cviu.2007.09.014

[15] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., “ORB: An Efficient Alternative to SIFT or SURF,” *2011 International Conference on Computer Vision*, IEEE, New York, Nov. 2011, pp. 2564–2571.
https://doi.org/10.1109/ICCV.2011.6126544

[16] Noble, F. K., “Comparison of OpenCV’s Feature Detectors and Feature Matchers,” *2016 23rd International Conference on Mechatronics and*

Machine Vision in Practice (M2VIP), IEEE, New York, Nov. 2016, pp. 1–6.
https://doi.org/10.1109/M2VIP.2016.7827292

[17] Ahrendt, P., "The Multivariate Gaussian Probability Distribution," Tech. Rept., Technical Univ. of Denmark, 2005.

[18] Rogers, R. M., *Applied Mathematics in Integrated Navigation Systems*, AIAA, Reston, VA, 2007, Chap. 14.
https://doi.org/10.2514/4.861598

[19] Ester, M., Kriegel, H. P., Sander, J., and Xu, X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *KDD 1996 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD-96*, Vol. 96, AAAI Press, Menlo Park, CA, 1996, pp. 226–231.

[20] Douc, R., and Cappé, O., "Comparison of Resampling Schemes for Particle Filtering," *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, IEEE, New York, Sept. 2005, pp. 64–69.
https://doi.org/10.1109/ISPA.2005.195385

Z. Sunberg
Associate Editor