Deep Learning-based Area Estimation for Unmanned Aircraft Systems using 3D Map

Masatoshi Hamnanaka
Center for Advanced Intelligence Project
RIKEN
Tokyo, Japan
masatoshi.hamanaka@riken.jp

***Abstract*—A method for estimating the flight area of unmanned aircraft systems (UAS) based on deep learning is described. The position of a UAS can be detected by using the global positioning system (GPS). However, GPS sometimes has difficulty capturing signals from satellites that are shielded by mountains and/or buildings. Moreover, GPS signals are very weak and subject to a variety of disturbances. Such problems increase the chance of a crash when flying a GPS-controlled UAS. As a solution, we propose a flight area estimation method using a 3D map created on the basis of deep learning. Our method could estimate the flight area with 92.2% accuracy in a simulation and 98.4% accuracy in a field experiment.**

***Keywords*— deep learning; 3D topographic map; lidar**

I. Introduction

We propose a method for estimating the flying area of unmanned aircraft systems (UAS) by using a 3D map created by deep learning. Position detection is extremely important for controlling the UAS and for preventing collisions. At present, most drones use the global positioning system (GPS) for position detection. However, GPS sometimes cannot indicate a proper position when signals from the satellites are shielded by buildings and/or mountains. Furthermore, GPS signals are very weak and hence are easily affected by a variety of interference [1, 2].

To enable better control of a UAS with GPS, we propose a deep learning [3, 4]-based area estimation method using a 3D map to which the UAS can refer when GPS signals cannot be received. Although deep learning has been used for estimating the positions of persons and objects from camera images, until now no such method has been proposed for estimating the position of a UAS [5, 6]. In order to develop a position estimating method using a 3D topographic map based on deep learning, we have to consider the following six points.

- 2D LiDAR:

  The flight area of a UAS can be estimated by using pattern matching of the 3D topographical map and the ground surface shape acquired by the UAS. In this case, it is easier to perform pattern matching between 3D shapes acquired using three-dimensional LiDAR. However, 3D LiDAR is heavy and difficult to mount on small UASs. Therefore, in this research, we equipped the UAS with a compact, lightweight two-dimensional LiDAR that was originally developed for automated driving of automobiles and used it to acquire a two-dimensional surface cross section (Fig. 1). We describe the acquisition of the 2D shape vector of the ground surface in Section II.A.

- Machine learning:

  Simple pattern matching between a 3D map and the 2D surface vector obtained by the UAS would take an enormous amount of calculations, making it difficult to do in real time.

  Therefore, we divided up the flight area and labeled each divided area in order to apply the machine learning for position estimation and reduce the number of calculations. We describe the area division in Section II.B.

- Deep learning:

  A machine learning technique such as those using support vector machines (SVMs) [7] does not work well for learning the relationship between the 2D surface vector and the label of the flight area because the amount of data obtained from the 3D map is enormous. SVMs would require exponentially increasing resources for constructing a model on such large datasets, which is infeasible.

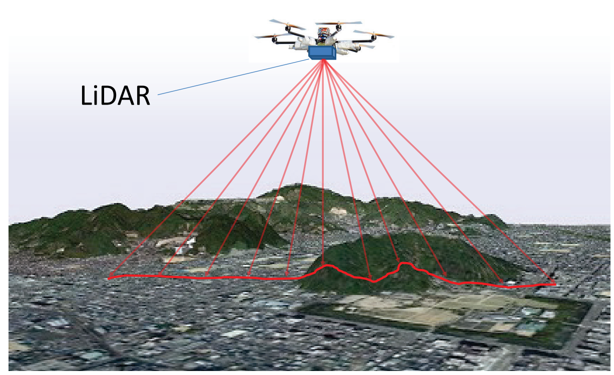

Fig. 1. Acquisition of 2D surface vector by 2D LiDAR.

Therefore, we used deep learning instead. In deep learning, learning data can be divided into groups called mini batches. By using mini batches, deep learning can handle a lot more data than SVMs can. We describe our multi-layer perceptron (MLP) [4], which is a type of deep learning, in Section III.

- Acquisition rate of learning data:

  Deep learning makes it possible to learn the relationship between the 2D surface vector and its label, but there is still a problem in that a long time is required when a large amount of data is involved.

  Since the surface shape will change along with the growth of plants and the construction of buildings, learning needs to be continued in batches in order to sustain the operation of the area estimation system. Therefore, the shorter the learning time, the fewer computing resources are needed.

  In deep learning, when the learning data becomes larger than the memory on the GPGPU, the learning speed suddenly becomes slow. When the learning data is smaller than the GPGPU memory, the number of data items and the learning time are almost proportional.

  Therefore, if we reduce the learning data by random sampling, the learning time can be shortened; however, if we reduce the amount of learning data, the accuracy will decrease. Section IV.A describes the acquisition rate of the data used for learning and the performance evaluation of discriminating the areas.

- Area margin:

  It is difficult to identify a label near the boundary of an area because ground surface shapes are similar yet the labels are different. We therefore create a margin near the boundary and exclude learning data in the margin. Sections IV.B and IV.C show how the performance of discriminating the areas changes depending on the margin width.

- Simulation and field experiments:

  Although a UAS can fly above 50 meters, commercially available LiDAR for automobiles has weak output signals and can only measure up to about 50 meters. We performed experiments under this limitation.

  First, in simulations, we verified the basic performance of discriminating the areas by using a 3D map acquired by a satellite. We also evaluated the acquisition rate and area margin in these simulations, as the data for field experiments was too small.

  Second, in a field experiment, we flew a UAS at an altitude lower than 50 meters and evaluated its performance of discriminating the areas. In these experiments, we evaluated in a limited area where it is possible to estimate correctly even in situations where there are missing data, such as when there is a puddle or the distance from the surface is too far.

The simulation results showed that after learning using an MLP consisting of 11 layers each with 3,000 units, the area on the 3D map could be estimated from the 2D surface vector with an accuracy of 92.2%. The field experiment showed that an MLP consisting of three layers with 1,500 units in the first hidden layer and a gradually decreasing number of units in the subsequent hidden layers (to 23 units in the deepest hidden layer) could estimate the area on a 3D map from the 2D surface vector acquired by the UAS with an accuracy of 98.4%.

Section II of this paper explains the data used for learning, and Section III describes the method for estimating the flying area by deep learning. Sections IV and V explain how we evaluated the method. We conclude in Section VI with a brief summary and mention of future work.

II. LEARNING AND EVALUATION DATA

Estimating the flight area of a UAS by using deep learning consists of two steps. First, the 2D surface vector is acquired by the 2D LiDAR mounted on the UAS. Next, the area is estimated from the 2D surface vector by using a network learned in advance. In this section, we describe the construction of the learning and evaluation data.

A. Acquisition of 2D ground surface shape

A 2D LiDAR (*LD-MRS*, *SICK* AG) is mounted on the UAS to acquire the ground cross-sectional shape. The altitude of the ground surface can be measured to an accuracy of about 20 centimeters if a LiDAR with an angular resolution of 0.125 degrees is mounted facing vertically downward and the UAS flies 100 meters above the ground [8]. In a simulation, we used 100 measurement points, one every 20 centimeters on the left and right of the flight path, and recorded the differences in height between the 100 points as a 100-dimensional 2D surface vector.

The number of measurement points varied between the simulation and field experiment, since the simulation covered a vast area while the field experiment used our own airspace, the area of which was limited. The distance between the measurement points also varied between the simulation and field experiment, as the resolutions of the 3D maps were different. Specifically, in the field experiments, we used 50 measurement points, one for every one meter, and the UAS acquired a 50-dimensional 2D surface vector.

B. Dividing up the flight area

We divided the flight area into a lattice and assigned different labels to its elements. The initial position of the UAS was randomly determined, and a set of the ground shape vectors calculated from the 3D map and the labels of the area were used as learning data.

Near the boundary of the flight area, it is difficult to distinguish the labels because labels may be different despite similar terrain information. Therefore, the performance was improved by providing a certain margin around the area.

We compared two types of area division in the simulations: 2D and 3D (Fig. 2). The height of the UAS was fixed at 100 meters in the simulation using 2D divisions. In the field experiment, the area was not divided up in the height direction.

C. 3D map from satellite

We used Advanced Land Observation Satellite (ALOS) Phased Array type L-band Synthetic Aperture Radar (PALSAR) data for making the 3D map [9] in the simulations. Although high-resolution 3D maps (resolution: ~10 cm) are available for a fee, we were not certain one would be necessary, so we used the free ALOS PALSAR data.

The ALOS PALSAR has a resolution of 30 meters. We reduced the scale to 1/150 in order use it as a 3D map with 20-cm resolution, which is the resolution that was used for acquiring the 2D surface vector in the simulation. The area was a cube with a side of 30 meters, and five areas were overlaid in each direction to generate 125 areas (Fig. 2(b)). The simulation used a satellite 3D map of 35°00'00" to 35°12'29" of north latitude and east longitude of 135°40'00" to 135°52'29". The height of the UAS ranged from 100 meters to 250 meters above the ground.

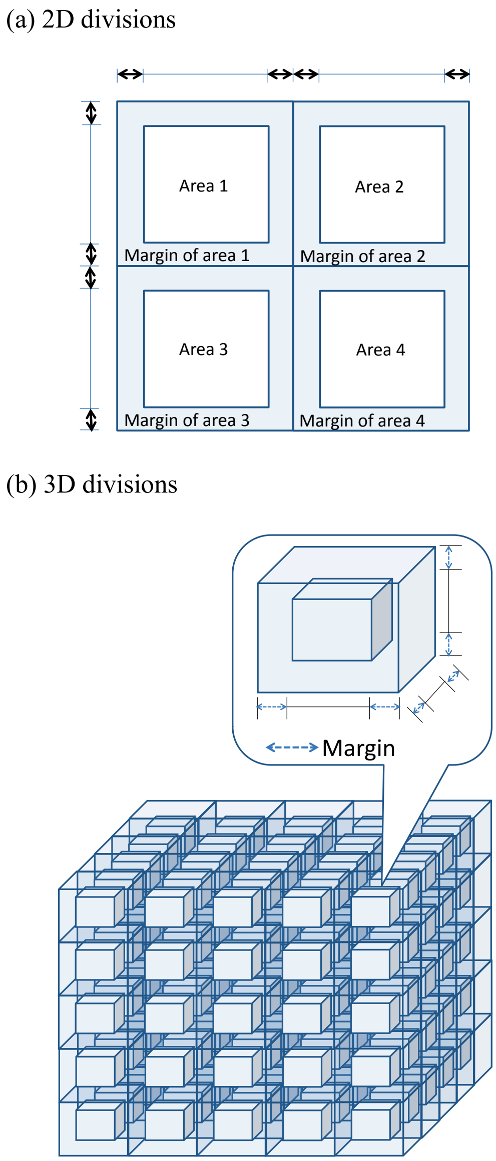

Fig. 2. 2D and 3D division and area margin.

D. 3D map from UAS

The 3D map used in the field experiment was created by flying the UAS equipped with LiDAR. The UAS flew over an area 200 meters square (36.6217 to 36.619900 north latitude and east longitude 139.1090 to 139.111240), which had a difference in ground altitude of 100 meters. Because the GPS equipped on the UAS (*Matrice 600* from *DJI*) [10] has an error of 1 meter, we conducted ten flights at a constant altitude and calculated the average altitude obtained with a resolution of 1 meter (Fig. 3).

E. Acquistion rate of learning data

The flight area in the simulation had a resolution of 20 centimeters for each axis. If we acquired data from all areas where five cubes with 30-m sides are stacked vertically and horizontally, even if the flight direction of the UAS were fixed, the number of data would exceed 400 million. It is difficult to calculate on such large data in the limited memory we had available. Moreover, if we divided up the data, the learning time would be unrealistically long. Therefore, we randomly sampled from all data of each area at an acquisition rate with which the calculation would be completed in a realistic amount of time.

F. Evaluation data

The evaluation data for the simulation experiment was randomly extracted from the 3D map. We avoided using the same data in the learning and evaluation in the simulation. The evaluation data for the field experiment were acquired by flying a UAS equipped with LiDAR.

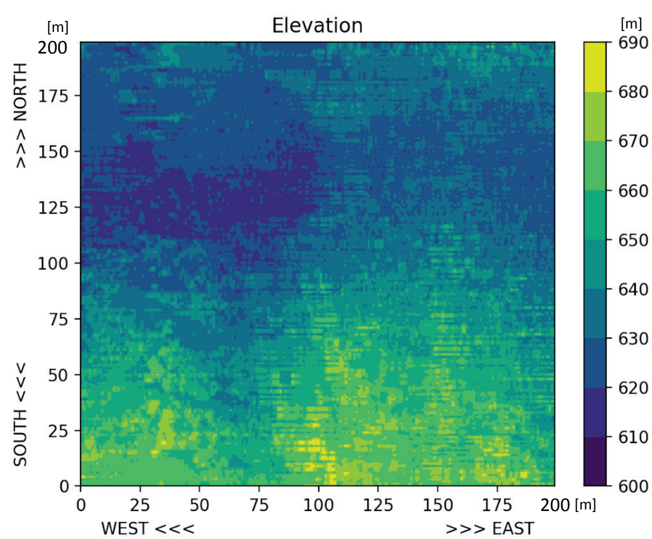

Fig. 3. 3D map from UAS.

III. FLIGHT AREA ESTIMATION BASED ON DEEP LEARNING

Pattern matching between the 3D map and the ground surface vector acquired by 2D LiDAR normally requires a very large number of matchings, which makes it difficult to perform in real time. We estimated the flight area from the shape of the ground surface in real time by using MLP [4].

Deep learning methods include convolution neural networks [11], MLPs, deep belief networks [12], etc. The advantage of MLP is that it can be used for any purpose. We chose to use MLP here because it is difficult to find meaning in the ground surface vector data itself.

The MLP for estimating the flight area is shown in Fig. 4. The input for the simulation experiment was 100 units corresponding to the 100-dimensional ground surface vector. We used 50 input units corresponding to the 50-dimensional ground surface vector in the field experiment, as the LiDAR signal was too weak to detect the ground shape if the height of the UAS was more than 50 meters. Each dimension of the surface vector was normalized to zero mean and variance one. The output units of the MLP were area labels. The MLP had *n* hidden layers; i.e., we varied *n* and the number of units in the experiments. The input layer, intermediate layers, and output layer were all fully connected.

Supervised learning with back propagation was used to determine all of the parameters of the network, including the bias of the units and the weights of the connections between the units.

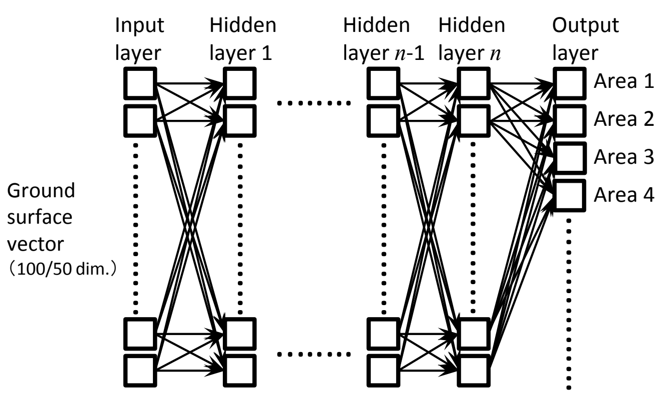

Fig. 4. MLP for estimating the area labels.

IV. SIMULATION EXPERIMENT

The simulation examined the effect of area division in two dimensions and in three dimensions.

A. *Performance evaluation varying acquisition rate in the case of 2D division*

We compared the performance of discriminating the four areas shown in Fig. 2(a) for learning data acquisition rates of 40%, 20%, and 10%. The total number of data was 2.25 million (30 meters (length of one side) / 0.2 (resolution of 20 cm)) ^ 2 (two directions in the vertical and horizontal directions) × 4 (the number of areas). Thus, 40% of the data amounted to 0.9 million (2.25 million × 0.4). The evaluation data comprised 10% of the total data, and it did not include any of the learning data. Although we could have extracted more than 40% of the data, it could not be stored in the memory on the GPGPU (*NVIDIA Quadro M600*) that we used.

Table I lists the accuracies for the data acquisition rates, which were obtained by subtracting the error rates from 1. The highest accuracy was for the 40% rate, then 20%, and then 10%. These results show that, as the number of data used for learning increases, the accuracy increases, and as the number decreases, the accuracy decreases. Because of the high accuracy, we could use an MLP consisting of seven hidden layers with 60 units each and 3,000 learning epochs.

TABLE I. PERFORMANCE EVALUATION: EXTRACTION RATE IN 2D DIVISIONS

Extraction rate	***40%***	***20%***	***10%***
Accuracy [%]	97.8	86.4	85.0

B. *Comparison with and without margin in 2D division*

We compared the discrimination performances with and without the margin for the data acquisition rate of 40%. Three margin widths were compared: 0% (no margin), 1% (30 cm), and 5% (150 cm), where one side of the area was 30 meters. The evaluation data and numbers of hidden layers, units, and epochs were the same as the previous simulation.

Table II shows that the 0% margin had the highest accuracy and the 5% margin had the lowest accuracy. This result means that discrimination performance is high regardless of the margin being small or large as long as there is sufficient learning data.

TABLE II. PERFORMANCES EVALUATION: WIDTH OF MARGIN

Extraction rate	***0%***	***1%***	***5%***
Accuracy [%]	97.8	96.7	92.0

C. *Comparison with and without margin in 3D division*

We compared the accuracies with and without the margin when discriminating a cubic area consisting of 125 cubic areas 30 meters on each side (Fig. 2(b)).

Here, we tested only a 10% acquisition rate with and without a 5% margin on the side of the 30 meter cubes, as the learning time would have been too long and the memory capacity of the GPGPU would have been exceeded if the higher rate were used. The evaluation data was 1/4 of the learning data for the cases with and without the margin.

We evaluated the effect of changing the number of units and layers, i.e., 1,000, 2,000, and 3,000 units and 1 to 15 layers, and varying the epoch number. The network with the highest performance had 11 hidden layers with 3,000 units each. Table III shows the results using this network where case (a) included the margin in both the learning and the evaluation data, case (b) included the margin in the learning data but not the evaluation data, case (c) included the margin in the evaluation data but not the learning data, and case (d) did not include the margins in the learning or the evaluation data, all for up to 3,000 epochs of learning. Figure 5 shows the learning curves. The learning time for 3,000 epochs was about half a day.

There were many errors in detection due to the margin in cases (a) and (b), while case (d) showed the best accuracy. These results demonstrate that it is better not to include a margin in the learning data when the UAS already knows its position by GPS.

The performance was very low in case (c), which did not include the margin in the learning data but included one in the evaluation data. This result means that when the UAS position is obtained only by this method, it is better to use a network learned with data including a margin.

TABLE III. PERFORMANCE WITH AND WITHOUT MARGIN

	(a)	(b)	(c)	(d)
Learning data	With margin	With margin	Without margin	Without margin
Evaluation data	With margin	Without margin	With margin	Without margin
Accuracy [%]	84.7	86.3	72.2	92.2

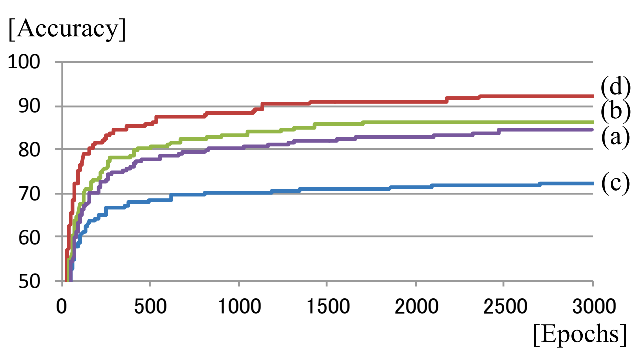

Fig. 5. Learning curves with and without margin.

V. FIELD EXPERIMENT

We evaluated the discrimination performance for the 3D map shown in Fig. 3. The UAS was a *Matrice 600* manufactured by *DJI* and equipped with *LD-MRS* LiDAR made by *SICK AG* (Fig. 6).

It is harder to estimate the position in the field experiment compared to the simulation, for three reasons.

- In the field experiment, the experiment area is limited and so the learning data are fewer than in the simulation. Therefore, we did not divide up the map in the vertical direction, since the LiDAR could only measure up to 50 meters.
- Even if the surface shape just below the UAV is acquired with a LiDER, the surface shape of the right end and the left end might not be acquired because the LiDAR signal is too weak. Therefore, we carefully examined the structure of MLP in order to increase the interpolation capability.
- The surface vector obtained by UAS includes errors because of the effect of UAV and wind. Therefore, we used zinbal (*Dji Ronin-MX*) for mounting the LiDAR on the UAV and flew the UAV at constant speed (Fig. 6).

Fig. 6. *Matrice 600* equipped with *LD-MRS*.

A. *Learning and evaluation data*

We fixed the direction of the UAS to north and the altitude to 35 meters above the average altitude in the area because of the limited memory of the GPGPU; that is, we would need fewer learning data compared to if we changed the direction and altitude. The area was 200 meters square (Fig. 3). The LiDAR on the UAS scanned a width of 50 meters. The area in which the surface vectors could be acquired was 200 meters north to south and 150 meters east to west. We made 30,000 (= 200 × 150) surface vectors in total for learning, as the resolution of the 3D map was 1 meter (Fig. 7).

We constructed the evaluation data by flying the UAS five times for 20 minutes at 35 meters (the average altitude) over the area. The UAS detected the surface vector every two seconds and collected a total of 3,000 (5 times × 20 minutes × 60 seconds / 2 seconds) surface vectors with area labels. The flight route was programmed in advance and it ran through multiple GPS points randomly selected in the area.

In order to evaluate under open conditions, we did not use the 3,000 surface vectors in the learning data that were nearest the GPS point of each surface vector in the evaluation data. We thus used 27,000 surface vectors for the learning data and 3,000 surface vectors for the evaluation data.

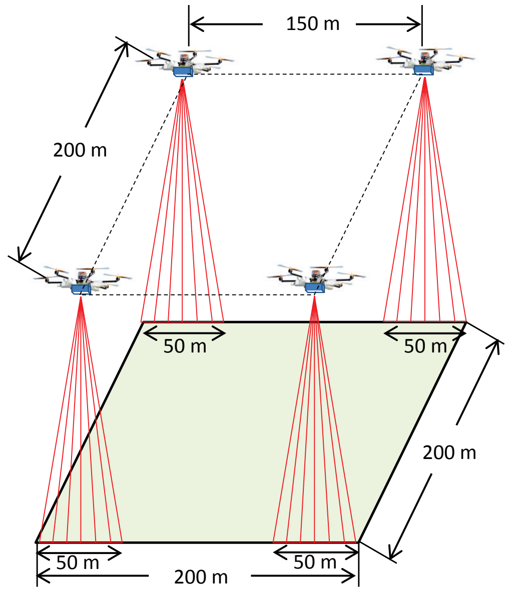

Fig. 7. Flight area in the field experiment.

B. *Effect of varying the number of hidden layers and number of units*

The performance of MLPs depends on the number of hidden layers and the number of units. By randomly changing the number of hidden layers and the number of units, we found that the performance was high with a deeply layered network in which the number of units gradually decreased the deeper the layer.

Therefore, we tried to determine the appropriate number of hidden layers and units in a network in which the number of units in the hidden layers gradually decreases at a constant rate. We changed the number of hidden layers from 1 to 7 in increments of 1 and changed the decrease rate of the units from 0.3 to 1.0 in increments of 0.1. The maximum number of units was 3,000. For example, when the number of hidden layers is 3 and the decrease rate is 0.5, the network consists of a first hidden layer with 1,500 (3000 × 0.5) units, a second hidden layer with 750 (1,500 × 0.5) units, and a third hidden layer with 375 (750 × 0.5) units.

Since it takes a long time for deep networks to learn, we attempted to save time by gradually narrowing the search area. Table IV shows the accuracy as a result of learning up to 2,000 epochs for each network. Gray colored cells indicate that the accuracy is lower than 0.5. The accuracies were low in the following three cases.

- When the number of layers and the number of units are high.
- When there is only one hidden layer.
- When the number of units near the output layer is very low.

Table V shows the results of learning for up to 15,000 epochs for each network whose accuracy is higher than 0.4 in Table 3.

Table VI shows the results of 30,000 epochs for the eight networks whose accuracy is higher than 0.94 in Table V, and Fig. 8 shows the learning curves of those networks. The network with three hidden layers with a decrease rate of 0.6 had an accuracy of 0.9762 after 20,509 epochs, and the one with seven hidden layers with a decrease rate of 0.5 had an accuracy of 0.9759 after 29,677 epochs.

TABLE IV. LEARNING UP TO 2,000 EPOCHS

	Decrease rate of number of units							
Number of layers	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	0.51	0.50	0.50	0.49	0.49	0.46	0.49	0.50
2	0.67	0.71	0.75	0.76	0.74	0.74	0.73	0.66
3	0.61	0.67	0.66	0.69	0.70	0.70	0.64	0.63
4	0.57	0.62	0.61	0.61	0.59	0.66	0.30	0.05
5	0.55	0.60	0.62	0.58	0.61	0.05	0.05	0.06
6	0.36	0.60	0.61	0.66	0.58	0.05	0.05	0.05
7	0.19	0.49	0.63	0.66	0.05	0.05	0.05	0.05
8	0.14	0.23	0.05	0.05	0.05	0.05	0.05	0.06

TABLE V. LEARNING UP TO 15,000 EPOCHS

	Decrease rate of number of units							
Number of layers	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	0.65	0.65	0.64	0.65	0.65	0.64	0.64	0.65
2	0.83	0.87	0.85	0.85	0.87	0.87	0.87	0.84
3	0.83	0.90	0.93	0.95	0.95	0.92	0.93	0.89
4	0.80	0.91	0.94	0.94	0.93	0.91	–	–
5	0.77	0.91	0.95	0.93	0.88	–	–	–
6	0.53	0.91	0.94	0.93	0.85	–	–	–
7	–	0.81	0.95	0.94	–	–	–	–

C. *Augumenting learning data with random noise*

Data augmentation improves the performance of deep learning in pattern recognition [13]. We attempted to augment the learning data by adding uniformly distributed random noise to each element of the surface vectors.

The total number of augmented learning data was 54,000: 27,000 original data and 27,000 augmented data in which random noise was added to the original data. The total number of non-augmented learning data for comparing the augmented learning data was also 54,000: 27,000 original data repeated.

We compared three noise levels: 0.25 meters, 0.5 meters, and 1 meter. We used the network with three hidden layers with a decrease rate of 0.6 and 7 hidden layers with a decrease rate of 0.5 because of high performance.

The results show that accuracy for evaluation data increased to 0.9838 when using the augmented learning data of the 0.5-meter noise level for learning 7 hidden layers with a decrees rate of 0.5. The learning time for 3,000 epochs was about 7 hours using an NVIDIA GTX 1080Ti GPGPU.

Because the experiment areas are different, it is difficult to compare the simulation and the field experiment. The performance of the field experiment was superior because it acquired the ground surface shape in more detail than the simulation and the network configuration was examined more carefully.

TABLE VI. LEARNING UP TO 30,000 EPOCHS

	Decrease rate of number of units		
Number of layers	0.5	0.6	0.7
3	–	0.9762	0.9705
4	0.9615	0.9607	–
5	0.9670	–	–
6	0.9756	–	–
7	0.9759	0.9669	–

TABLE VII. AUGUMENTED LEARNING DATA WITH RANDOM NOISE

	Noise level			
Number of layers and decrease rate	Original	0.25 m	0.50 m	1.00 m
3 layers with a decrease rate of 0.6	0.9762	0.9730	0.9721	0.9712
7 layers with a decrease rate of 0.5	0.9759	0.9782	0.9838	0.9813

VI. CONCLUSION

We developed a method to estimate areas by deep learning of surface vectors acquired by LiDAR. The following four points are the main contributions of this study.

- Development of an area estimation method using 2D LiDAR

  We proposed a method that can estimate the position of a UAS by using a lightweight 2D LiDAR mounted on the UAV. Pattern matching between 2D surface vectors and 3D maps normally requires a very large number of matchings. In order to estimate the area of UAS in real time, we used machine learning for the area estimation.

- Use of deep learning for area estimation

  SVMs would require an exponential increase in resources to deal with the enormous 2D surface vectors from a 3D map. Therefore, we used less costly deep learning for the area estimation.

- Performance evaluation through simulation

  We conducted a simulation using a wide range 3D map from a satellite and determined that our method could estimate the correct area of the UAS flight with an accuracy of 92.2%.

- Evaluation in real world

  We augmented the learning data with 1% random noise from a 3D map of a 200 meter square area made by using LiDAR on the UAS. The MLP with seven hidden layers consisting of 1,500, 750, 375, 187, 93, 46, and 23 units could estimate the flight area of the UAV when the map was divided into 25 areas with an accuracy of 98.4%, after leaning the augmented data.

We plan to learn an MLP for a huge scale area by using the satellite data from the Advanced Land Observation Satellite (ALOS 2) Phased Array type L-band Synthetic Aperture Radar 2 (PALSAR 2).

ACKNOWLEDGMENTS

This work was supported by the Japan Society for the Promotion of Science (JSPS KAKENHI Grant Number 17K19972).

REFERENCES

[1] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned Aircraft Capture and Control Via GPS Spoofing," Joarnal of Field Robotics, Vol. 31, Issue 4, pp. 617–636, 2014.

[2] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the Impact of Stealthy Attacks on Industrial Control Systems," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1092–1105, 2016.

[3] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep Learning," Nature, Vol. 521, Issue 7553, pp. 436–444, 2015.

[4] Amari S, Ozeki T, Karakida R, Yoshida Y, and Okada M, "Dynamics of Learning in MLP: Natural Gradient and Singularity Revisited," Neural Comput, Vol. 30, Issue 1, pp. 1–33, 2018.

[5] J. C. Caicedo and S. Lazebnik, "Active Object Localization with Deep Reinforcement Learning," Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV '15), pp. 2488–2496, 2015.

[6] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[7] V.Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.

[8] SICK, The LD-MRS family – compact laser scaners for outdoor use, from: https://www.comtrol.com/elements/uploads/fckeditor/file/LD-MRS_4001021.pdf, 2018.

[9] A. Rosenqvist, M. Shimada, and M. Watanabe, ALOS PALSAR: Technical outline and mission concepts, 4th Internationa Symposium

on Retrival of Bio- and Geophysical Parameters from SAR Data for Land Applications, 2004.

[10] Dji, Matrice600, from https://www.dji.com/matrice600, 2018.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems, pp. 1097–1105, 2012.

[12] G. E. Hinton, S. Osindero, and Y. W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," Journal of Neural Computation, Vol. 18, Issue 7, pp. 1527–1554, 2006.

[13] J.J . Lv, X. H. Shao, J. S. Huang, X. D. Zhou, and X. Zhou, "Data augmentation for face recognition," Journal of Neurocomputing, Vol. 230, Issue C, pp. 184–196, 2017.

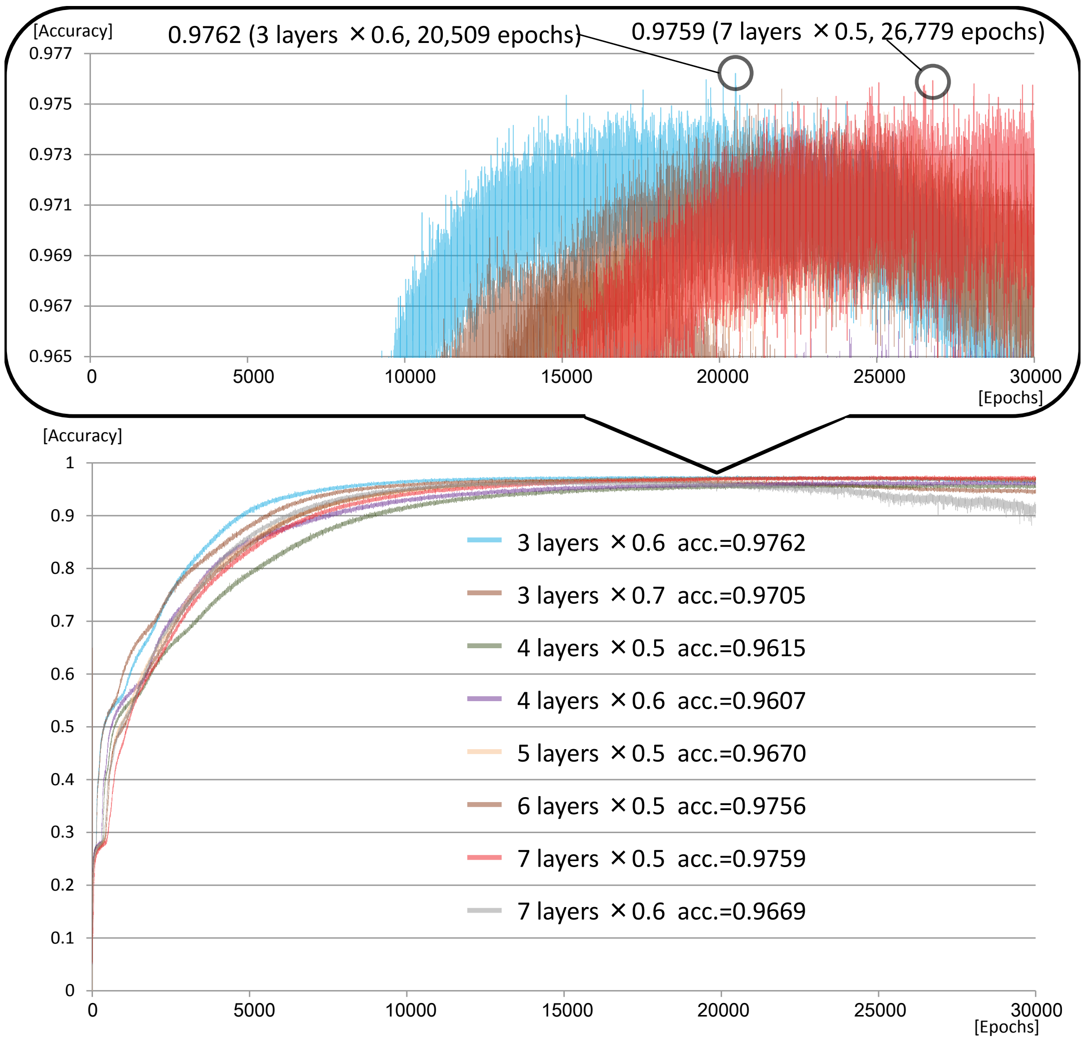

Fig. 8. Learning curves of eight networks whose accuracy is higher than 0.94 in Table V.