An UAV Navigation System Aided with Computer Vision

Bohan Tong
School of Information and Communication Engineering
Dalian University of Technology
Liaoning, China
Email: 1308751722@qq.com

***Abstract*—The navigation system is always the key point in unmanned aerial vehicle(UAV) area. Besides the traditional solution of Inertial Navigation System(INS), as well as the Global Navigation Satellite System(GNSS), the navigation system based on computer vision is an alternative method when GNSS has a poor performance under weak signal situation. In this paper, we construct a navigation system based on scene matching method. The matching area with rich and stable features is selected, then the real-time dynamic scene is matched with pre-stored images to provide location information. The navigation system constructed in this paper combines the method of feature point matching and deep learning methods by Dlib library. At the end of this paper, the navigation system is implemented with C++, and is simulated in Microsoft AirSim visual environment.**

***Keywords*—navigation system; scene matching; AirSim visual environment**

I. INTRODUCTION

INS/GNSS integrated navigation system is the most common method of UAV positioning and navigation nowadays, the INS system will give real-time flight data via the sensors inside, like the UAV's height, velocity and attitude; while the GNSS system will provide precise positioning data to make up for the increasing error of INS. However, in practical applications, the navigation system above has a few disadvantages, for example, the GNSS system is sensitive of external interference and can not meet requirements of real-time control in high dynamic and velocity scenes, and the INS system will accumulate error quickly over time, with its huge size and expensive cost[1].

In recent years, with the rapid development of vision sensors and image processing technologies, the speed and resolution of vision sensors have continued to increase, while costs have continued to decrease. Therefore, scene matching positioning navigation technology based on computer vision has attracted much attention as a new navigation method. In this system, the current environment image is collected by the airborne vision sensor, and matched with the pre-stored electronic map, providing error correction information for the INS, and then constructing an autonomous navigation system with high precision, anti-interference, and independent of satellite positioning.

II. SELECTION OF SCENE MATCHING AREA

The procedure of scene matching area selection includes processing on the map of the scheduled flight route before flight, and selecting the image with appropriate size, large amount of information and good matching as the navigation reference map, so as to match the real-time image obtained by the airborne visual sensor with the pre-stored reference map in flight, so as to assist the INS to provide location information. Among them, the key to the selection of matching area is to select the feature and determine the matching area[2]. For a given scene image, its features should be analysed from general to detail as follows: the information contained, stability, uniqueness, etc[3][4].

A. Information in matching area

The information in the matching area is the basic requirement in scene matching method. Suppose the image with single texture is selected as the matching area, such as flat ground and desert, the visual navigation system may not locate accurately above these areas. In order to filter out the images with less information, we use gray variance to describe every image in this system. The gray variance means the degree of deviation in each pixel from the image's mean gray value, it shows the variation degree of the whole image gray level. The image with bigger variance always has a larger amount information contained, while the smaller ones always represents for less significant characteristic and poorer matching performance. The gray variance can be described as follows[3]:

$$V_{ar} = \sqrt{\frac{1}{mn-1}\sum_{i=1}^{m}\sum_{j=1}^{n}(f(i,j)-\overline{f})^2} \tag{1}$$

In (1) above, f represents for a image, $\overline{f}$ is the image's mean gray value, and m,n is the number of rows and columns.

B. Stability of matching area

However, the images filtered by the information amount may not all suitable for the requirements of scene matching. Because of the external factors like CMOS model, time, seasons, the pre-stored images and real-time images may have far differences. Therefore, the selected matching area should have stable ground objects for reference. In this paper, we judge the image's stability by Frieden entropy[3]:

$$H(f) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} p_{ij}e^{1-p_{ij}} \tag{2}$$

The f represents for a $M*N$ image, and definition of p_{ij} is as follows:

$$p_{ij} = f(i,j) / \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j), f(i,j) \geq 0 \tag{3}$$

C. *Uniqueness of matching area*

What's more, the matching method should be able to identify the similar objects in one reference image. The analyze on image's uniqueness is based on statistical analysis of correlation surface. Each position in the real-time image is matched with the reference map and then the system outputs the series of similarity values. The values are arranged into a two-dimensional correlation plane, and the peak value is always the matching position.

III. Scene matching method

The scene matching method is similar to the traditional image alignment and matching. That is, to determine the spatial relationship between images shot in different environments. However, the scene method should have a real-time and robustness performance. In this system, we combine the methods of feature point matching and deep learning.

A. *Matching methods based on feature points*

Due to the limitation of performance and size on embedded platform, the matching methods should be simplified for faster speed. So the Oriented Fast and Rotated Brief(ORB) algorithm is applied in this system[5].

1) Key point detection: The first step in ORB is to detect key points by Features from Accelerated Segments Test(FAST) algorithm. Supposed that there is a circle with a pixel p in Fig. 1 as the center, if a continuous series of points on this circle has a bigger or smaller intensity than the center, the pixel p could be considered as key points. Here Ip stands for the intensity of pixel p, the threshold to specify the pixels is h. To reduce the searching time, we set the radius of circle to three pixels, and could compare the value of p with only four near pixels, for example, the pixels labeled 1,5,9 and 13 in Fig. 1.

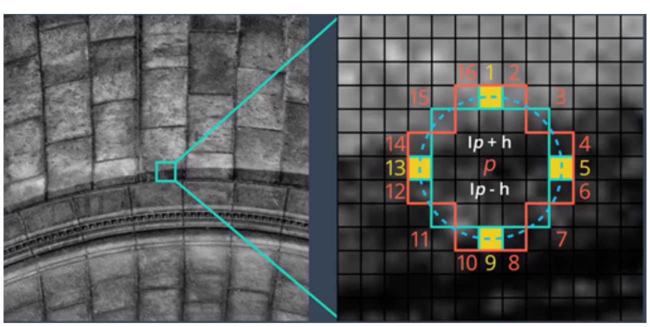


Fig. 1: Key points detected by FAST algorithm

The main task of FAST algorithm is to detect feature points from the image efficiently. For example, Fig. 2 shows a image with smaller size and more details as real-time image and a reference map with wide field of vision. Fig. 3 illustrates 10000 feature points detected by using the FAST algorithm.

(a) real-time image (b) reference map

Fig. 2: Two original images

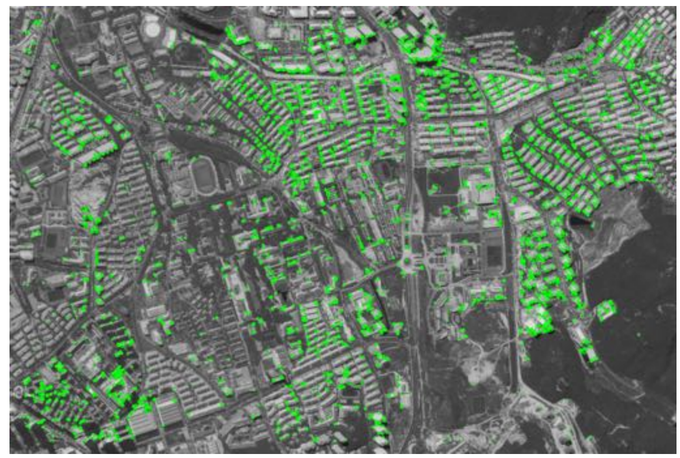


Fig. 3: Feature points detected by FAST

2) Feature descriptor: The second step of ORB algorithm is to transform the key point detected by FAST to feature vectors with Binary Robust Independent Elementary Features(BRIEF). In the neighborhood square area of a key point called patch, the first step is to use Gaussian filter to reduce high frequency noise. Then randomly select a pair of points $< x, y >$ as shown in Fig. 4, compare the brightness of these points, returns 1 if x is brighter than y and write to a array. After loop for 128-512 times, a binary code like $V_1 = \underbrace{[001011 \cdots 011]}_{\text{128-512 bits string}}$ is generated, which is the descriptor of this key points.

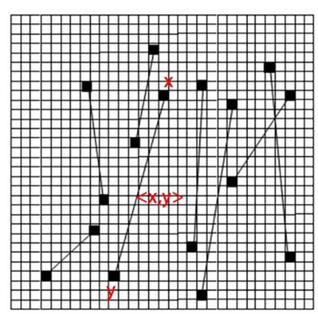


Fig. 4: BRIEF descriptor

To compare the accuracy and speed of BRIEF and other algorithms like SIFT and SURF, we have tested matching performance, the ten best matching points are drawn in Fig. 5, Fig. 6 and Fig. 7 respectively. The Table 1 shows the time cost and accuracy of best matching 10 feature points, using BruteForceMatcher methods to match pairs of features. The test environment is Intel(R) Core(TM) i7-8750H CPU@2.20GHz, AMD(R) Radeon(TM) Pro 560X, 16GB RAM, macOS 10.15.6, Xcode 12.1, OpenCV 4.4.0.

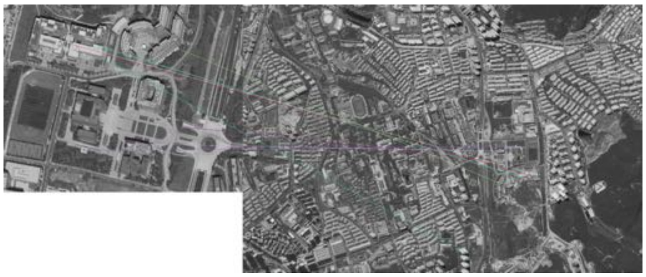


Fig. 5: Matching results by BRIEF

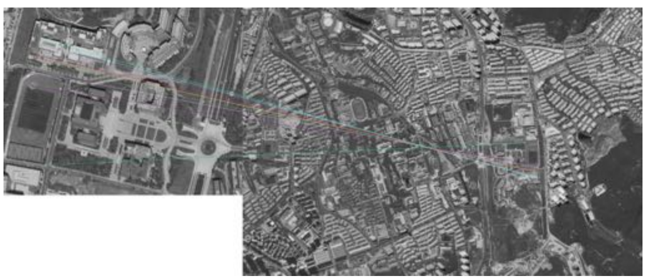


Fig. 6: Matching results by SIFT

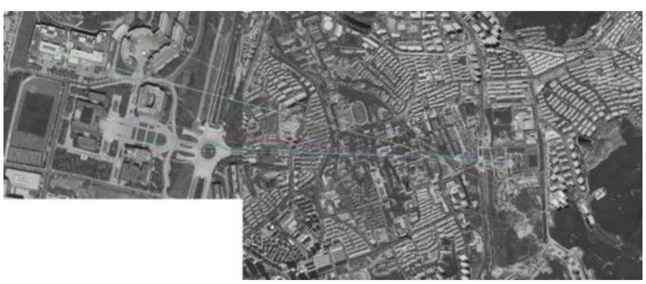


Fig. 7: Matching results by SURF

TABLE I: Results of scene matching

Methods	Time cost/s	Accuracy
BRIEF	0.4925	90%
SIFT	2.4381	100%
SURF	1.5539	100%

From the figure and table above, the matching results by BRIEF has faster speed than SIFT and SURF, but performs about 10% poorer in accuracy.

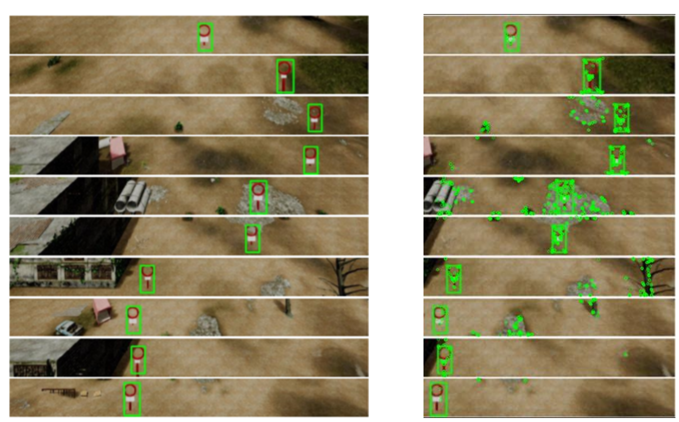
B. Matching methods based on deep learning

Dlib contains a wide range of machine learning algorithms. All designed to be highly modular, quick to execute, and simple to use via a clean and modern C++ API. It is used in a wide range of applications including robotics, embedded devices, mobile phones, and large high performance computing environments[6]. In this system, we labeled the feature and objects in a series of train images, than trained a model for detection based on Dlib library.

IV. Simulation

The navigation system is implemented with C++ in Microsoft Airsim environment[7] as shown in Fig. 8. The environment has a few obstacle circles, and the main task is to cross the circles in order. The reference map is generated by the camera on UAV, and the scene matching areas are selected according to the features like brightness and information mentioned in Section 2. The next step is to use ORB algorithm to detect all the feature points and label all features to train the detection model. The labels and features points are shown in Fig. 9. When the UAV takes off, it will firstly scan all the features by calling to the trained model and ORB algorithm. Then we will get a array describing the obstacles' position, the UAV will finish the remaining tasks with vision algorithms. And Fig. 10 is shows the moment when UAV is crossing one circle.

Fig. 8: Simulation environment



(a) real-time image (b) reference map

Fig. 9: Two original images

Fig. 10: UAV crossing obstacle

V. Conclusion

In this article, we address a UAV navigation system aided with computer vision. We first elaborate the selection methods of scene matching area, according on the information, stability and uniqueness of the image. Then the matching method based on feature points and deep learning are presented, and their performances are compared with other existing algorithm, which proves that they are more suitable for high dynamic and velocity scenes. Finally, we simulate the system in Airsim virtual environment.

References

[1] C. Zhao, Y. Zhou, Z. Lin, J. Hu, and Q. Pan, "Overview of vision navigation technology for uav scene matching," *Chinese Science: Information Science*, no. 5, p. 1, 2019.

[2] L. Song, "Research on key technology of uav visual navigation in flight," Ph.D. dissertation, Xi'an: Northwestern Polytechnical University, 2015.

[3] Z. Liu, H. Wang, and B. Chen, "Research on selection method of scene matching area," *Computer technology and development*, vol. 23, no. 12, pp. 128–133, 2013.

[4] Y. Zhang, J. Su, and B. Li, "Research on scene matching adaptability in scene matching guidance," *National security Geophysics Series (11) – geophysical application frontier*, 2015.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[6] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[7] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065