Design of a Navigation System Based on Scene Matching and Software in the Loop Simulation

Ayham Shahoud
Faculty of Innovative Technology
Tomsk State University
Tomsk, Russia
ayhams86@gmail.com

Dmitriy Shashev
Faculty of Innovative Technology
Tomsk State University
Tomsk, Russia
dshashev@mail.ru

Stanislav Shidlovskiy
Faculty of Innovative Technology
Tomsk State University
Tomsk, Russia
shidlovskiysv@mail.ru

2021 International Conference on Information Technology (ICIT) | 978-1-6654-2870-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIT52682.2021.9491778

***Abstract*—This paper presents a design and a comparative study between two types of navigation systems based on scene matching. The first one depends on the cross-correlation and the other depends on Scale-Invariant Feature Transform (SIFT). A simulation environment that depends on Robot Operating System (ROS) was adopted with the 3D dynamic simulator Gazebo. The IRIS drone model was used and equipped with a camera, inertial sensors, Global Positioning System (GPS), and a compass. To reduce the matching time between the captured image and the georeferenced image, an adaptive window was designed. Numerous experiments were done online using all sub-models connected in a loop as in real work situations with zero cost. The RMS error of the position was less than 2.5m for both systems. Due to the adaptive window, the execution time for the correlation-based method was 40ms, which was three times less than the SIFT-based method execution time.**

Keywords—Correlation; Adaptive Window; Scene Matching; SIFT; ROS; Gazebo

I. INTRODUCTION

The navigation system is the main part of any aerial autonomous vehicle. The accuracy of the navigation system highly affects the autopilot performance and the vehicle task. Navigation systems based on computer vision are commonly used nowadays, especially for drones [1]. These systems are light, cheap, and might benefit from cameras that are already mounted on drones for general purposes. Computer vision navigation systems are independent compared to GPS which suffers from outages and depends on satellite signals.

Although visual navigation systems are independent of other external systems, they are highly dependent on the exterior environment such as features, illumination, shadows, weather, seasons, and scale. Such problems have been studied for years, and a lot of solutions were found. Nowadays, there are robust methods to detect image points (features) that are invariant to scale and rotation like SIFT. In visual navigation systems that use cross-correlation function, rotation and scale problems are treated using other sensors to align images before the matching process. Compasses and altimeters are used in the alignment process [2]. Each method has advantages and disadvantages when taking into account execution time, accuracy, and robustness.

The main problem that arises when talking about computer vision navigation systems is the testing. A lot of difficulties appear, such as the cost, repeating tests, tuning parameters, and wasting time because of bad weather for example. In our work, we have designed a navigation system and tested it using a drone that flies in a 3D environment shown in Fig. 1. Numerous tests had been done to fix and test the algorithm parameters and analyze the results with zero costs. All that due to the high flexibility of the adopted simulation environment.

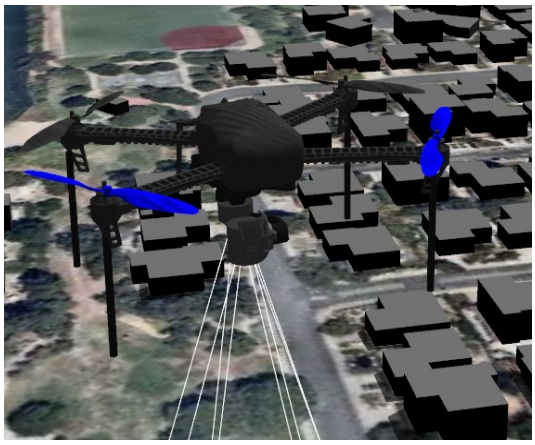

Fig. 1. IRIS drone model flying in the 3D flight environment with onboard sensors and a downward-facing camera fixed on it.

The rest of this paper is organized as follows: Section II for related studies, Section III for scene matching, Section IV for navigation algorithms, Section V for implementation, and Section VI for results analysis and the conclusion.

II. RELATED STUDIES

The cross-correlation function and the convolution function are famous methods for matching two images. They were studied carefully and used for many applications like navigation, path tracking, surveillance, and auto-landing since the last century.

A detailed study of the application of discrete cross-correlation function for an observational-comparative aerial navigation system is explained in [2].

Reference [3], shows an image matching system for an autonomous Unmanned Aerial Vehicle (UAV) based on neural networks. The final position was calculated using cross-correlation between the reference image and the obtained image of edges.

A vision-based absolute localization system for the UAV depending on optimizing a similarity measure between the image and the reference map is implemented in [4]. The reference map was a set of georeferenced images, and the similarity measure was the mutual information between the two images.

David Lowe published his famous paper about scale-invariant feature transform SIFT [5]. It was an algorithm used to detect and describe the local features in images. SIFT applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, and the individual identification of wildlife.

A fusion algorithm for position estimation of the UAV based on vision aided with multi-sensors is implemented in [6]. The Extended Kalman Filter (EKF) was used as a fusion algorithm and local features were used for scene matching. The proposed computer vision navigation system successfully replaced the GPS signal outages.

Reference [7], explains the development of a robust and efficient airborne scene matching algorithm for UAV navigation. A novel aerial image matching technique based on Simplified Haar-like Local Binary Pattern (SHLBP) was proposed to obtain the position of matching points. Random Sample Consensus (RANSAC) was also applied to remove mismatches by iteratively minimizing the average residual.

A fast and robust scene matching method for navigation is introduced in [8]. The "coarse to fine" matching method was used. It combines area-based and feature-based matching methods. It was used to meet the requirements of navigation, including real-time performance, sub-pixel accuracy, and robustness. The results showed that the proposed method achieved sub-pixel matching accuracy and improved angle accuracy to 0.1˚.

III. SCENE MATCHING

A. Scene Matching Based on Cross-Correlation

Computer vision navigation systems that depend on correlation techniques are very trustworthy because of their high success rates in a lot of applications. Accurate and robust results were obtained, especially in military applications like cruise missiles. The main problem at that time was the large execution time of the cross-correlation matching method. Nowadays, this problem becomes less significant with the existence of high-performance processing units and the Graphics Processing Unit (GPU).

Before calculating the cross-correlation, it is necessary to align both the captured image and the reference image as shown in Fig. 2. Alignment means that the captured image must be rotated and scaled to match the reference image scale and orientation [9]. Let '*T'* be the captured image and '*I'* the georeferenced image (the map). The Normalized Cross-Correlation (NCC) is given by the following equation:

$$R(x,y)=\frac{\sum_{x',y'}(T(x',y').I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2.\sum_{x',y'}I(x+x',y+y')^2}} \quad (1)$$

The position of the drone will correspond to the patch that produces the highest correlation value with the georeferenced image as shown in Fig. 3.

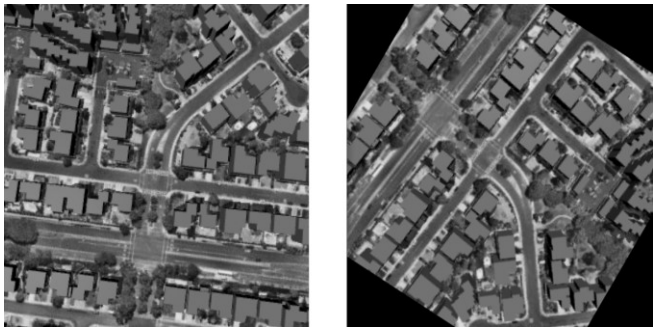

Fig. 2. The Scaled image to left and after being rotated to the right.

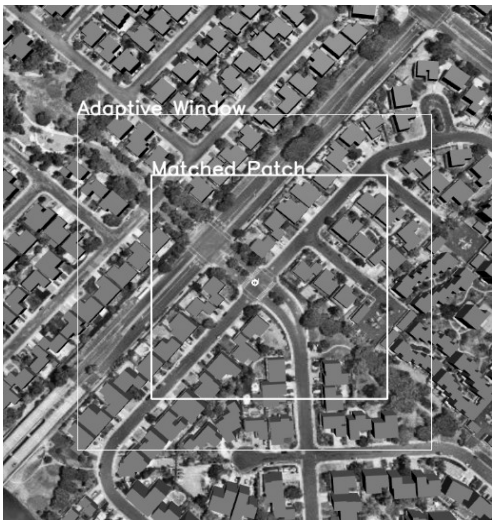

Fig. 3. The adaptive window on the map and the matched image patch position.

The NCC method needs alignment between the captured image and the reference map. The alignment process creates the need for other sensors like a compass and altimeter. The NCC resistance to illumination variation (because of the normalization) makes it a good choice, especially in outside environments.

The captured image new scale '*s*' is calculated using the following equation:

$$s=h\times f\times I_r \quad (2)$$

'*h*' is the height of the drone which could be obtained from an altimeter or GPS. 'I_r' is the map resolution that is calculated one time using a known object in the 3D environment. '*f*' is the camera focal length [9].

Let 'ψ_m' be the orientation of the map and 'ψ' the current image orientation. The image must be rotated to establish the alignment. The rotation angle 'α' is given by the following equation:

$$\alpha=\psi-\psi_m \quad (3)$$

B. Scene Matching Based on SIFT

Traditional corner detectors such as Hessian and Harris work well in case of small variations in the scale. The performance of these detectors degrades with large-scale variations. Since in navigation tasks the scale and orientation change with camera motion, then more robust detectors against the scale and rotation variation are needed [5, 10]. A good detector must fulfill the following characteristics:

- The accuracy: the position of the detected point must be accurate because it will directly affect the navigation solution.
- Repeatability: the ability to detect the same point every time it appears in the scene.
- Time of detection: it must be acceptable according to the application.
- The point must be well described, so it should have unique descriptor.

As shown in the related study, one famous feature detector and descriptor called SIFT was presented by David Lowe.

In short, SIFT constructs the scale-space of the image for several layers, then searches for the features. Features are defined in SIFT detector as the minimum or the maximum endpoints that maintain themselves within all constructed layers, in accordance with a certain cost function.

In the second step of SIFT, the gradian is calculated in an area of 16x16 pixels around the feature point. The calculated gradian is used to define the orientation of the feature relative to its surrounding pixels. A vector of 128 bytes called descriptor is obtained and connected to each feature. The matching between the captured image and the referenced image is realized on the level of descriptors and can be done using Euclidian distance. A matching result between an image and the map is shown in Fig. 4. Previous studies proved that SIFT provides good resistance to scale and orientation variations, but not good under illumination variations [10].

The center of the camera and the center of the drone are assumed to be identical. The drone position can be calculated using one of the following techniques: 2D-2D matching, 3D-3D matching, or 2D-3D matching [11].

In our work, we used 2D-3D matching. In short, knowing the 2D coordinates of a set of points in the image coordinate frame, and the coordinates of their correspondences in the 3D world coordinates frame is enough to calculate the camera position. The camera orientation also could be calculated [11, 12]. We used the "Perspective-n-Point" (PnP) algorithm to solve the 2D-3D matching problem. RANSAC was applied to calculate the optimal solution.

C. Design of the Adaptive Window

Instead of matching the captured image with the whole map, we cropped a part of the map and matched it with the captured image as shown in Fig. 3. The size of the cropped image changes according to the drone velocity and to the scaled image size as shown in the following equations:

$$W_w = 1.3 \times s \times T_w + v_x \times \tau \quad (4)$$

$$W_h = 1.3 \times s \times T_h + v_y \times \tau \quad (5)$$

$W_{w,h}$ is the cropped window size, width, and height respectively. $T_{w,h}$ is the online image size, width, and height. $v_{x,y}$ is the drone velocity in x and y directions (pixels/s), $v_{x,y} \times \tau$ could be approximated to the previous movement amplitude. 'τ' is the ROS publishing image period, in our work, 'τ' depends on the execution time of the algorithm. New images will not be captured until the current process ends. '*s'* is the image scale after alignment in the NCC-based situation only.

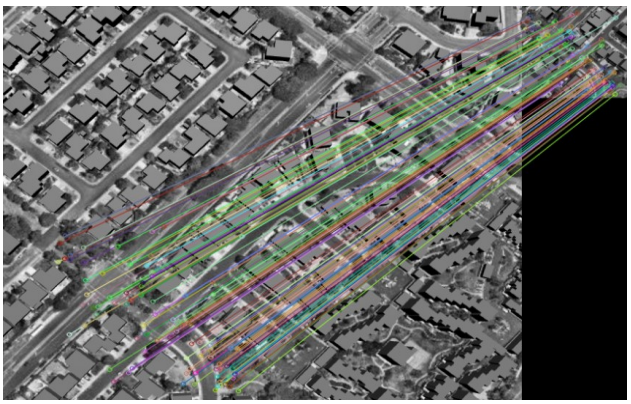

Fig. 4. Features matching between the captured image to the right and reference image (map) to the left.

Since in the SIFT-based method there is no need for alignment, then normally '*s*' must be equal to 1. In our work, we fixed it to 0.5 and scaled the captured image by the same value. This number has been fixed after a lot of tests to benefit from the main characteristic of SIFT which is the robustness against scale variation in reducing execution time. This contribution reduced the size of the matched images, and thus the time consumed in detecting and matching features became smaller.

IV. Navigation Algorithms

A. NCC-based Navigation Algorithm

Taking into account the following assumption, we have implemented the algorithm shown in Fig. 5. The world coordinate frame origin is assumed to be identical to the launch point. The y-axis is pointed to the north and the x-axis is pointed to the east.

The captured image was converted into grayscale, then enhanced to improve the correlation result. After that, the captured image was aligned to the scale and orientation of the map, then matched with the map using the NCC function. Finally, the position of the drone corresponded to the patch with the highest correlation value.

B. SIFT-based Navigation Algorithm

Using the same previous world coordinate frame and assuming a flat ground. We have implemented the algorithm shown in Fig. 6. After detecting and describing the features in both images, the matching process was done on the level of descriptors, then PnP and RANSAC were used to calculate the optimal position of the drone.

V. Implementation

A. Software and Hardware Equipment

For simulated implementation, an HP pavilion laptop was used with i510300 - 2.5Ghz CPU. A camera model with a 60˚ field of view and an image size of 300x300 pixels was used. All the programs were written using Python 3.7 under Linux. OpenCV 3.6 library was used for image processing.

B. Simulation Environment

From Gazebo we used an urban 3D model as a flight environment for the drone. From Ardupilot we used the IRIS drone model. Ardupilot is an open source, unmanned vehicle autopilot software suite capable of controlling autonomous drones. The drone was equipped with a camera, inertial sensors, compass, and GPS. We wrote subscribers to the camera images, the compass, and the navigation solution published by ROS with MAVROS communication protocol. The navigation solution published by ROS is the output of the fusion algorithm (EKF) between the GPS and inertial sensors, and it was used as a reference.

MAVROS protocol is a middle protocol that translates the messages of the models into ROS messages. ROS messages are easy to use and common between different robot systems.

Launching and controlling the drone path was done using Software In The Loop (SITL). SITL is a simulator that allows ArduPilot to run on the computer without any hardware. All the previous software (ROS, Gazebo, Ardupilot, SITL, along with our Python program) together formed a flexible simulation environment that helped us easily to perform a lot of tests.

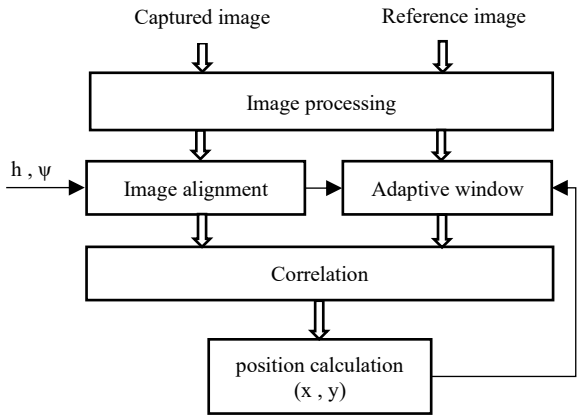

Fig. 5. NCC-based navigation algorithm.

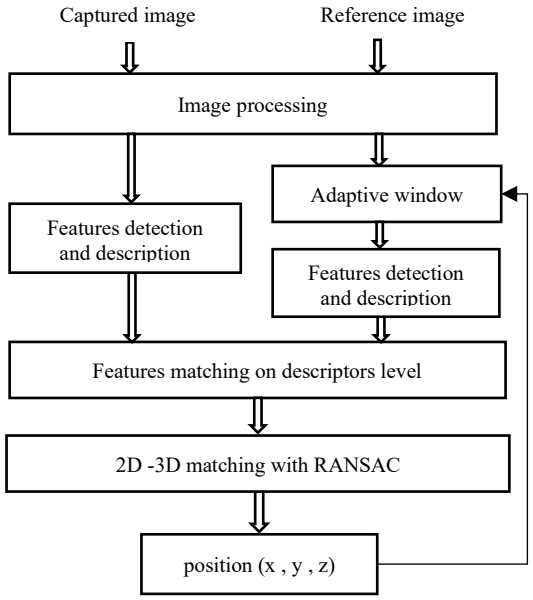

Fig. 6. SIFT-based navigation algorithm.

C. *Map Specification*

A georeferenced image was used as a map. It was captured from a height of 500m with an image size of 900x1000 pixels and a resolution of 0.53 m/pixel.

D. *Experiments*

The position was calculated online during the flight and stored in a "csv" file. After that, the results were plotted using Octave. The drone flew on a path with an average speed of 7ms^{-1} and an average height of 150m. The captured images were processed online using one of the previous algorithms, and the position was calculated and stored. The following figures show the results obtained from the two algorithms on the same path. In the figures, "ref" refers to the reference path and "vision" to the calculated path.

The results of the NCC-based method are shown in Fig. 7 and Fig. 8. The RMS error of the position in the (x,y) plane is equal to 2.4m. The SIFT-based results are shown in Fig. 9, Fig. 10, and Fig. 11. The RMS error of the position in the (x,y) plane is equal to 2.1m. In Fig. 11, we notice a 10m difference between the reference and the measured height. This difference was mainly due to the fact that the detected features used in position calculation lie on trees, buildings, and streets in the used 3D urban model.

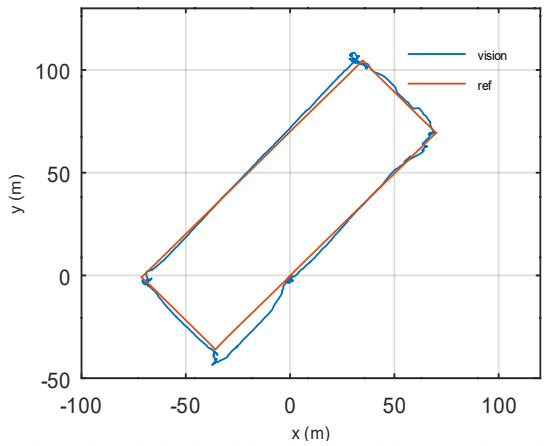

Fig. 7. The calculated and the reference path in (x, y) plane using NCC-based navigation method.

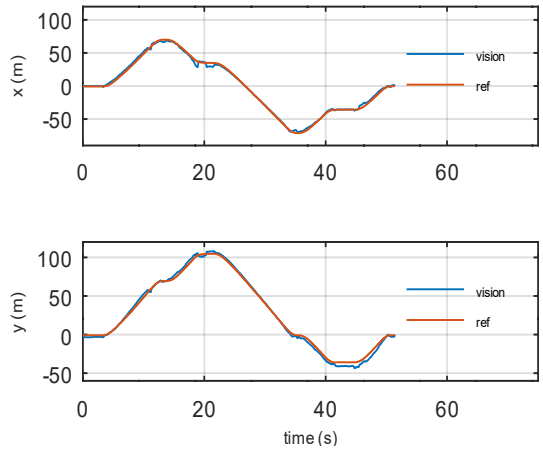

Fig. 8. The position on x and y using NCC-based navigation method.

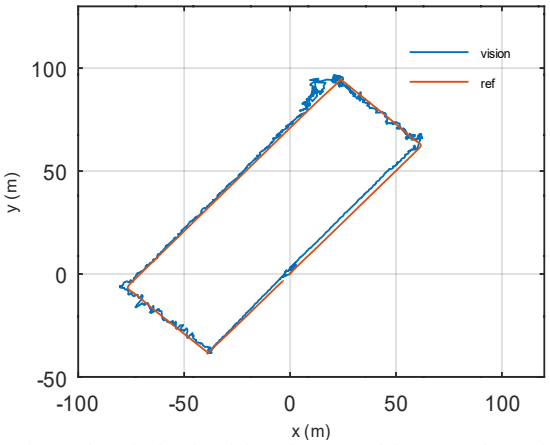

Fig. 9. The calculated and the reference path in (x , y) plane using SIFT-based navigation method.

From the previous results, we can figure out that our assumption of flat ground was not completely accurate, but it was within acceptable margins. We can roughly say that the 10m difference is approximately the average height of features that were considered as inliers with RANSAC to calculate the position using PnP. It is an approximation and not an exact value because the 10m difference includes other errors from the used sensors and the map resolution. These errors are out of the scope of this paper.

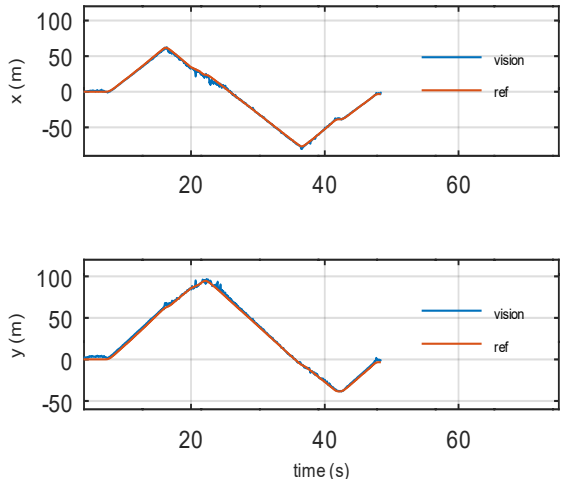

Fig. 10. The position on x and y using SIFT-based navigation method.

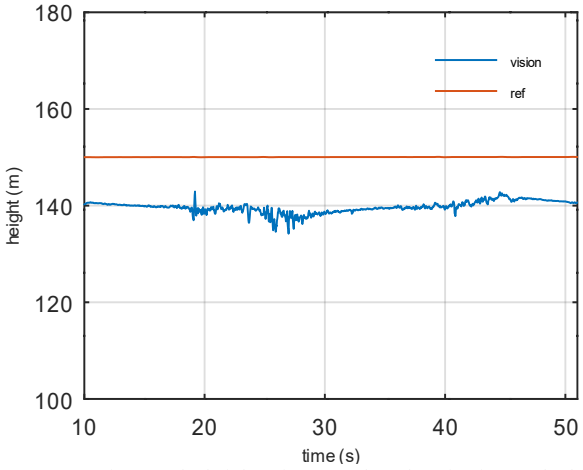

Fig. 11. The height using SIFT-based navigation method.

VI. Results Analysis and Conclusion

A. Result Analysis

We can notice a remarkable disturbance in the position when the drone stops or starts to move. The main reason for that disturbance is the vibrations (in large magnitude up to 7º) when the drone changes the speed or the direction. A summary of the results is shown in Table I.

The execution time of the NCC-based method is equal to 40ms, and it is equal to 121ms in the SIFT-based method. Detecting and describing feature points consume more time than the NCC calculation. The template matching in OpenCV is highly optimized which is why we have gotten such results. Taking into account the position accuracy, both algorithms almost have the same performance with a little advance for SIFT. The RMS of the error is less than 2.5m in the horizontal plane for both algorithms. These results were expected because SIFT depends only on a set of points on the image. Losing a part of the image during large vibrations will not cancel the solution as long as there will be enough matched points to solve the 2D-3D matching problem (at least 4 matchings for PnP).

The SIFT-based method needs less information about the environment. The flat ground assumption was assumed just to facilitate the calculation of the 3D coordinates of any point on the map.

TABLE I. Results Summary

Navigation Algorithm	Comparison parameters		
	RMS error in (x,y) plane (m)	*Execution time (ms)*	*Needed input measurements*
based on SIFT	2.1	121	-
based on NCC	2.4	40	height and heading angle

The SIFT-based method outputs the height and the rotation matrix of the camera or drone relative to the world coordinate. As opposed to SIFT, The NCC-based method needs the height and heading angle to calculate the position.

Due to the adaptive window and the scaling of 0.5, we got an execution time for the SIFT-based algorithm of 121ms. We tried to further reduce the execution time by scaling both captured images and the adaptive window with values less than 0.5, but this returned large errors and forced us to further reduce the speed. The SIFT-based execution time is acceptable for integrated navigation applications to correct the inertial navigation system drift. As a standalone system, it will be better to further reduce the execution time by using a suitable GPU.

In the integrated navigation systems, we might think of benefiting from the fast execution time and the robustness against the illumination variations of NCC. In these systems, other sensors offer the needed information (h and ψ) for the alignment process. The NCC will be very efficient if suitable matching areas are available in the environment such as street intersections. False matching occurs in the NCC-based method due to the fact that NCC is a mathematical operation, and it always has a result. The navigation algorithm must be able to decide if that match should be accepted or rejected. Excluding false matches can be done using statistical or artificial solutions, a topic we shall be focusing on in future work.

B. Conclusion

This paper presented an implementation of two navigation systems based on computer vision. The first system depended on the normalized cross-correlation function to match a captured image with a reference map. The second system used the local features detected by SIFT in the matching process. An adaptive window was designed to avoid matching the captured images with the whole map in both systems.

The navigation systems were tested on a path using the IRIS drone model equipped with a camera, inertial sensors, GPS, and compass in an advanced simulation environment. Numerous experiments were conducted using all sub-models connected together in a loop identical to real work situations with zero cost. The obtained results proved the efficiency of the designed adaptive widow in reducing the execution time.

Finally, a comparative study between the two systems was done. The results showed that the NCC-based method has an execution time of 40ms, which was three times less than the SIFT-based execution time. The SIFT-based method RMS error of the position was 2.1m, which was less than the correlation-based method RMS error of the position that was equal to 2.4m.

Our future work will focus on finding solutions for false matches that take place in the NCC-based method. We will also work on compensating for the drone vibration effects on localization.

REFERENCES

[1] Belmonte, L.M.; Morales, R.; Fernández-Caballero, A. Computer Vision in Autonomous Unmanned Aerial Vehicles—A Systematic Mapping Study, Applied Sciences. 2019, 9, 3196.

[2] Matuszewski, Jan & Grzywacz, Wojciech. (2017). Application of Discrete Cross-Correlation Function for Observational-Comparative Navigation System. Annual of Navigation. 24. 10.1515/aon-2017-0004. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] J. R. G. Braga, H. F. C. Velho, G. Conte, P. Doherty and É. H. Shiguemori, "An image matching system for autonomous UAV navigation based on neural network," 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand, 2016, pp. 1-6, doi: 10.1109/ICARCV.2016.7838775.

[4] A. Yol, B. Delabarre, A. Dame, J. Dartois and E. Marchand, "Vision-based absolute localization for unmanned aerial vehicles," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 2014, pp. 3429-3434, doi: 10.1109/IROS.2014.6943040.

[5] Lowe, David. (2001). Object Recognition from Local Scale-Invariant Features. Proceedings of the IEEE International Conference on Computer Vision. 2.

[6] Abdi, G. & Samadzadegan, Farhad & Kurz, Franz. (2016). Position estimation of unmanned aerial vehicles based on vision aided with multi sensors fusion. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XLI-B6. 193-199. 10.5194/isprs-archives-XLI-B6-193-2016.

[7] Duo, Jingyun & Zhao, Long. (2017). A robust and efficient airborne scene matching algorithm for UAV navigation. 1337-1342. 10.1109/ICCSN.2017.8230327.

[8] Ulas, Cihan. (2013). A Fast and Robust Feature-Based Scan-Matching Method in 3D SLAM and the Effect of Sampling Strategies. International Journal of Advanced Robotic Systems. 10. 10.5772/56964.

[9] Conte, Gianpaolo & Doherty, Patrick. (2009). Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information. EURASIP J. Adv. Sig. Proc.. 2009. 10.1155/2009/387308.

[10] Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.

[11] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," in IEEE Robotics & Automation Magazine, vol. 18, no. 4, pp. 80-92, Dec. 2011, doi: 10.1109/MRA.2011.943233.

[12] Youyang, Feng, Wang Qing, Yang Yuan, and Yan Chao. “Robust Improvement Solution to Perspective-n-Point Problem.” International Journal of Advanced Robotic Systems, (November 2019).