Scene matching in GPS denied environments: a comparison of methods for orthophoto registration

Erick Menezes Moreira, Otavio Augusto Maciel Camargo, Julio Cesar Duarte, Paulo Fernando F. Rosa
Graduate Program for Computing and Systems Engineering
Instituto Militar de Engenharia
Rio de Janeiro, Brazil
E-mails: {emenezes, otavioamcamargo, duarte, rpaulo}@ime.eb.br

***Abstract*—Unmaned Aerial Systems (UAS) is a subject with great appeal, nowadays. Suitable applications for this kind of vehicles are countless. This work presents a study on scene matching (navigation with global positioning using image registration) in GPS denied environments. Usually, the problem is solved by fusing inertial data with other sources of displacement measurements like simultaneous localization and mapping or visual odometry. Although helpful, those methods establish a navigation system based on relative positioning, since they are limited to foreseeing displacements from previous states. Image registration can refresh the UAS position with georeferenced data, i.e., an absolute global position. This work reaches that goal through a detailed analysis of methods capable of extracting image information (e.g, edges or other invariant key points), enabling the registration process and consequently the estimation of position. Our results were quantified in terms of error estimation and time cost for computation. They have shown that good performance is achieved with a Convolutional Neural Network complemented with SURF approach.**

I. Introduction

Unmanned Aerial Systems (UAS) navigation is a very sensitive technology since it is the foremost activity developed by any vehicle. In most cases, a UAV is guided by a global navigation satellite system (GNSS) receiver and a Inertial Measurement Unit (IMU). The former feeds the vehicle with a georeferenced position and the latter introduces a more immediate notion of movement in the system.

In recent years, many global navigation satellite systems became available, such as GPS, GLONASS, BeiDou and Galileo. They are maintained by their own government who can deliberate about its availability at any time. This means that the minimum satellite constellation needed by a UAV for positioning may not be always available. Besides that, we must consider the numerous areas of shadow, technical problems with receivers or any other hitch which may leave the system unworkable. Lukashevich et al. in [1], Braga et al. in [2] and Conte et al. in [3] talked about other possible problems like: GPS signal blocking and jamming, multi-path signal reflection due to the proximity of obstacles and even phenomena related to the Earth's ionosphere. On the other hand, the IMU embedded at the vehicle and available with redundancies is always present. However, the estimated data is suitable only for a short period of time due to quick accumulation of error.

Another very common sensor embedded in a UAS is the camera. A monocular camera sensor is a lightweight hardware which can be equipped for taking pictures and recording videos. Some techniques were developed to leverage that data, e.g., simultaneous localization and mapping (SLAM) and visual odometry (VO). In SLAM, the guiding system senses the environment, creates a map of it and navigates the area in respect to that map. In turn, VO estimates displacements using an optical flow field, created from analysis of sequential images. The application of those two methods over the sensed data enables the vehicle some form of vision based mobility. Since the estimate is based on an earlier state, it may suffer from drift errors and, as a result, some kind of georeferenced position update is in order, usually, when the GNSS comes in. These geocorrections keep the navigation more reliable as they eliminate the accumulated IMU and drift errors.

As stated, during a flight there may be some absence of GNSS signal leaving the system to the aforementioned error prune methods. For such occasions, another form of georeferenced position input must be in hand; image registration can be employed for this purpose. It consists of taking images of the area where the UAV is going to navigate as reference and, as the vehicle flies, register in it the moving images (i.e., find which piece of the reference image is more similar to the image acquired by the camera sensor). Since our reference image is georeferenced, we can deduce a global position for any point in it.

The hypothesis stated in this article is that it is possible to obtain an absolute position for UAS guidance using image registration. More specifically, the proposed approach is robust enough to accomplish its purpose even when using images from different sensors taken from different heights. We tested the hypothesis against edge detection based methods and keypoint feature detection methods, e.g., Sobel, FAST, BRISK and SURF. In addition, we also tried to detect edges using a pretrained Convolutional Neural Network (CNN) and a multilayer perceptron neural network (NN-MLP).

The remainder of this work is organized as follows: in section II, some basic concepts are shown and in III we present some related works. After that, we formally introduce the problem and the methodology used in IV and the simulation results in V. Finally, in section VI, we show our conclusions and the perspective for future work.

II. Image registration process

According to Conte et al. [4], Image Registration consists in geometrically aligning two images, that can be in different scales and rotations [5], [6]. The objective of image registration is to find the value of a matrix T through the analysis of the reference and the moving images. T is the transformation matrix (geometric transformation) of points in the moving image into the reference image. For the sake of performance, an approximated position can be given, thus relieving the process of analyzing the whole reference image. This forecast may come from other sensors, methods or fusion of data, e.g., IMU readings, visual odometry, SLAM displacement analysis or a fusion of all of then.

It can be calculated in many different ways, but our tests are focused in methods that detect and analyze edges, corners or keypoint features of the images. The process can be summarized in four steps: (a) *image preparation*, (b) *feature detection*, (c) *feature matching* and (d) *image registration*.

A. Image preparation

In order to analyze images efficiently, some preparation is necessary. This phase aims to make adjustments in the input image so that the features may be detected more accurately, this phase procedure depends on the detection algorithm because each one has its own peculiarities.

Most sensors tend to capture colored images. In our case, RGB images, which increases the demand for computational resources significantly. Therefore, we are going to convert RGB images to grayscale, leaving the images ready for keypoint detection methods and some edge based methods. Some of them may require even more transformations. For instance, the neural network NN-MLP method which needs the input image to be binarized. This demand was met with the application of Otsu's method after the conversion to grayscale.

There are methods that do not require any preparation, like the CNN that computes a RGB image directly through its layers for feature detection.

After the preparation, all edge base methods require scaling to meet the proportion of the reference image, because their algorithms are not scale invariant. Keypoint based methods do not require that, however, in the tests, they presented better results when scaled. In a UAV, the altitude required for scaling can be extracted from distance sensors like the barometer, the ultrasound or the laser range finder.

B. Feature detection

In this phase, we look for distinct features in the moving image for later registration over the reference image that should already be feature detected. Here we describe the methods for feature detection techniques.

The first algorithms are those which use edges as the only feature to be detected. We tested Sobel, Canny and Prewitt methods, which are classic algorithms for edge detection. Besides them, we also tested two more modern approaches: an Artificial Neural Network NN-MLP trained for that purpose, described by Braga et al. [2] and a pre-trained Convolutional Neural Network, which has its initial layers devoted to feature detection.

In NN-MLP, a 3 by 3 filter goes through the image following a left to right, top to bottom approach to filter the image. At each step, the nine selected elements are used as input for computation. In a 800 by 800 pixels image, over seventy thousand computations are made and each one classifies its input as edge or not edge. The result of the detection is printed in a second image, originally blank and with same size as the first one, where a black mark is draw in the pixel corresponding to the center of the 3 by 3 inputs considered edges.

This Neural Network was trained with the suggested architecture in [2]: nine input neurons, eighteen neurons in the hidden layer and two outputs for classification as edge or not edge. Also, it uses the tangent hyperbolic as the activation function, the error is calculated and adjusted using the descent gradient method with momentum rate of 0.85, learning rate of 0.73 and minimum performance gradient of 1e-06. To train this network we used 26 patterns, where 24 are edge patterns and 2 are not edge patterns. The whole training was made in 276 epochs.

Convolutional Neural Network is a state of the art computer vision technique. The CNN used is the VGG-19, it has 19 layers and is trained for classification of a thousand different classes [7]. The image is passed as input with no preparation and after five layers of computation we have as output a batch of 64 images with 224 by 224 pixels each. We inspected visually the 64 filtered images and selected the 52th on behalf of its edgy based pattern. The image result is clear, almost only with borders, one example can be seen in Fig. 3a.

The second set of algorithms for feature detection are those which use keypoints to match the images. After preparation, it tries to detect distinctive and invariant features on each image. Harris, Minimum eigenvalue algorithm, FAST, and BRISK detect corners, SURF detects blobs and MSER detects regions of uniform intensity. It can be seen in Fig. 4b, 4a and 4c respectively.

C. Feature matching

Feature matching consists in comparing the features detected from the reference image with those detected in the moving image.

All edge based methods, after processing and detecting features from the images, have to compare the moving image to every part of the reference image. It does so by sliding the moving image through the reference from left to right and from top to bottom, scanning the whole picture. At each sliding step we crop the reference image R and calculate the correlation between the moving image M and cropped reference C, as showed in eq. 1. The maximum correlation position is the best position found of the moving image in the reference and also the estimated position (x_r, y_r).

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \max_{(x,y)\in R} \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(M_{ij}-\overline{M})(C_{ij}-\overline{C})}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}(M_{ij}-\overline{M})^2(C_{ij}-\overline{C})^2}} \tag{1}$$

The keypoint based methods, after detecting features, select only the valid ones, based on criteria like intensity threshold level, region area range and, number of octaves. Finally, the features from every image are matched to find common features between them.

D. Calculation of matrix T

The calculation of matrix T consists in finding the transformation that maps points in the moving image to the reference image. Fig. 1 illustrates all steps needed for registration.

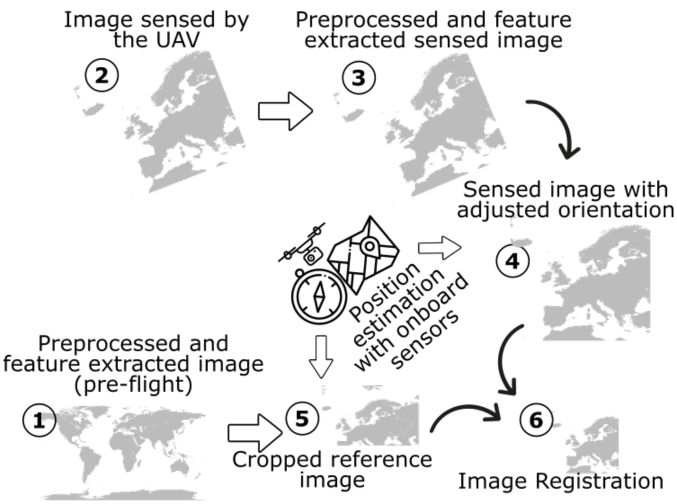


Fig. 1. Overview of image registration process (adapted from [2])

On edge based methods, after the search for maximum correlation position, we are left with the (x_r,y_r). Eq. 2 shows T for edge based registration methods, where C and F are the estimated x_r and y_r, respectively. The whole edge based method process is summarized in Fig. 2. Fig. 2d shows the exact position of the moving image overlapped in the reference. CNN image registration uses correlation too. An illustration of the process is shown in Fig. 3.

$$T = \begin{bmatrix} 1 & 0 & C \\ 0 & 1 & F \end{bmatrix} \tag{2}$$

Keypoint based methods, are a little bit different than edge based methods. After matching the detected features, it estimates a geometric transformation. It maps the matched points in the moving image into the reference image. The matching and registering can be seen in Fig. 4d and 4e.

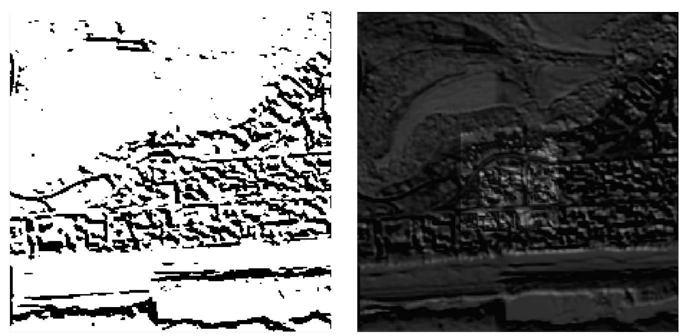
III. Related works

The camera and IMU sensors have been paired in others studies before. Probably, it is due to the fact that this sensors are statistically complementary, as stated in [8]. This means, we have a sensor that has a low frequency with good precision (camera) and a high frequency one which accumulates error very rapidly (IMU), [1] [9].

(a) Reference image (b) Moving image

(c) Moving image with edges detected (d) Registration made

Fig. 2. Edge based method step-by-step

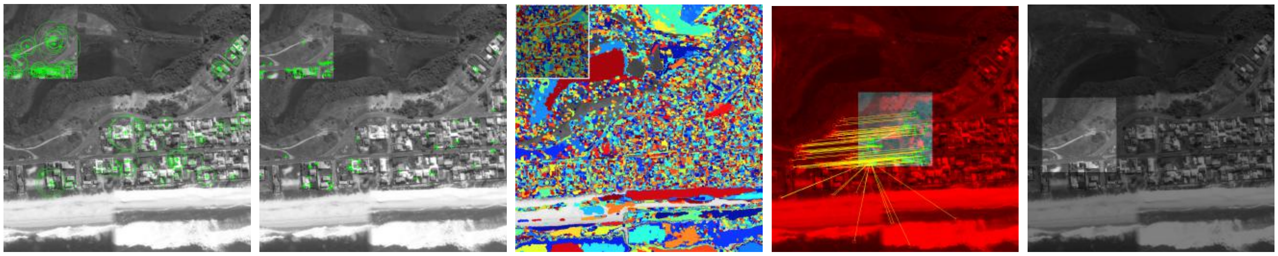


(a) 52th image from the batch output of the 5th layer layer (b) Image registered

Fig. 3. CNN registration method

SLAM approaches, have good results when applied in crowded places, rich in details so that the robot can extract unique references from the landmarks for navigation ([10][11]). Also, the mission of these robots must allow the revisitation of those landmarks for loop closure and consequent orientation of the vehicle. The UAS application scenario can make those two conditions difficult to meet. In high altitudes the images sensed from the scene may became too homogeneous, making it difficult for the detection of references. Besides that, the mission of a UAV usually covers large distances, making the second condition for SLAM a problem.

Recent publications brought application for scene matching



(a) Feature detection Blobs detected | (b) Feature detection Corners detected | (c) Feature detection Regions detected | (d) Features matching in both images | (e) Estimate Geometric transformation

Fig. 4. Keypoint based method step-by-step

in the UAS context. In [2], [3], [4], the authors enhanced the navigation accuracy using image registration based on edge detection, their methods were used as baseline comparison to ours. Their methodology, to the best of our knowledge, use images sensed at the same altitude, while we aim at using images from different altitudes and sensors (satellite and on-board camera).

IV. Problem Formulation

The experiments executed in this work are a study focused in the definition of a method for image registration of orthophotos of different scales and rotations in a satellite sensed image for UAV navigation in GPS denied environment.

We are searching for a robust method that must be capable of dealing with the fact that, a UAV flying an arbitrary route, probably will not acquire images in the same scale and orientation of the reference image. Also, chances are that the sensed area contains different elements compared to the same spoken reference.

In order to estimate the latitude and longitude of a UAV from an aerial photography (sensed or moving image), we use a georeferenced image (reference image) sensed by a satellite or an aerial photography taken previously. In our experiments, we consider a vehicle operating at 150 meters, therefore our reference images are taken from 500 and 1000 meters. Besides acquiring them, we manually georeference and create the *world file* of the image using the *Quantum Geographic Information System - QGIS*. The word file for an affine transformation, i.e., a first order polynomial, consists of 6 parameters A to F, which we use to transform pixel position on the reference image to world coordinates (x_w, y_w), respectively longitude and latitude. In order to compute the transformation, we use eq. (3), where x_r is the horizontal displacement and y_r is the vertical displacement in pixels from the top left corner of the reference image.

$$\begin{bmatrix} x_w \\ y_w \end{bmatrix} = \begin{bmatrix} A & B & C \\ D & E & F \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \tag{3}$$

In order to estimate a position (x_r, y_r) for the UAV in the reference image, we first must compute a transformation which takes a pixel in the moving image (x_m, y_m), sensed by the camera, into that reference image. Eq. 4 shows the transformation process, where T is the resulting transformation matrix (geometric transformation) of points in the moving image into the reference image.

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = T \begin{bmatrix} x_m \\ y_m \end{bmatrix} \tag{4}$$

The only thing that is missing now is to solve the problem of finding the matrix T. For that, we use techniques of image registration based on edge detection and feature detection.

A. Methodology

The methodology expects that the reference image is already loaded in the UAV, while the moving image is sensed by the camera during flight. For improved fairness during tests, no position estimation is given, i.e., all methods analyze the whole reference image. They also use the same variables and input images and they are executed in the same hardware and software conditions.

Before tests, we calculated the **correct position** of all moving images, (x_r, y_r) coordinates in each one of the reference images. The correct position is essential to measure the error of the estimated position after the registration process.

For the experiment, we use ten different flight locations, and for each one we acquired two reference images, one at five hundred meters and another at one thousand meters of altitude. Also, for each location, we have five more images at one hundred and fifty meters of altitude. We are testing five moving images for each reference image of each location, summing up one hundred tests with seventy satellite images of 800 by 800 pixels of resolution. Moreover, during flight, vehicles may sense images with a different orientation compared to our reference or even some noise introduced by scenario change or weather. In order to emulate this conditions, this experiment is executed with images aligned, rotated by a maximum of ten degrees and with noise addition (brightness). In addition, all moving images are from April 2015 while the references are from August 2017. Figure 5 exemplifies these transformations.

Also, we have compared the most famous edge based algorithms: Sobel, Canny and Prewitt. The results can be seen in table I, where the error and match rate are described in section V. The results are very similar, and based on them, we opt to use only the Sobel method in our tests.

(a) Image rotated and blended with a non rotated

(b) Image rotation before cropping

(c) Image without noise and brightness addition

(d) Image with noise and brightness addition

Fig. 5. Images modes used in the experiments

TABLE I
SOBEL, CANNY AND PREWITT COMPARISON

Average	Sobel	Canny	Prewitt
Error	2.92	3.03	3.04
Time cost	2.01	2.00	1.99
Match rate	0.61	0.64	0.64

V. SIMULATION RESULTS

The methods we used in our experiments are listed in table II.

TABLE II
METHODS AND ALGORITHMS TESTED

Algorithm	Use
Sobel with Correlation	Edge detection
NN-MLP with Correlation	Edge detection
CNN with Correlation	Edge detection
SURF	Keypoint Detection
MSER	Keypoint Detection
Harris	Keypoint Detection
Brisk	Keypoint Detection
Min Eigen	Keypoint Detection
FAST	Keypoint Detection

The first criteria used to measure the results was the time cost, which considered only operations involving the moving image. In a real flight, a UAV would have reference images processed and loaded in memory.

The second criteria evaluated to measure results was the average error (eq. 5), which consists in the euclidean distance in pixels between the estimated position and the correct position (assigned manually).

$$\varepsilon = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{5}$$

We also calculated the match rate r, which is the number of correct assignments c_i in image registration, divided by the total number of comparisons T of the method in current mode and altitude (eq. 6).

$$r = \frac{\sum_{i=1}^{n} c_i}{T} \tag{6}$$

Figure 6a shows the average time cost results. It is clear that keypoint based methods have better performance. In this criteria, the CNN method is the only edge based method with results similar to keypoint based methods. In figure 6b, we have the average error results. We can see that the error is small and regular in most algorithms, except when rotated images are used. Keypoint methods are more precise when registering rotated images. Finally, figure 6c shows the matching rate results where the CNN method demonstrates the best results overall. With an average of 77% of matches, 20% more than the method SURF which figured in second place.

This results show that NN-MLP edge detection got better results than Sobel in images rich in details. When using the reference image at 1000 meters the results from Sobel were much worst than NN-MLP. On the other hand, Sobel results in images with noise and brightness addition were similar to ones for the CNN. Also NN-MLP did not perform weel with brightness noise. In addition, all the edge based methods proved to be able to deal with small rotations, as good as keypoint based methods.

Among the edge based methods, all of them have similar image quality results. Sobel is much faster than NN-MLP to detect edges, although for the image dimensions in this experiment, this was not relevant. Another difference lies in the thickness of the border, NN-MLP makes borders thicker than Sobel, generating better results when the reference image is in higher altitude. CNN detections were faster and more accurate than the other ones.

Keypoint based methods are very fast computationally but do not do well with images that are too rich in features, like urban and residence areas. In most registrations with that kind of images, keypoint methods are not able to find enough matches.

VI. FINAL CONSIDERATIONS

A. Conclusion

This work has shown that CNN methods have excellent performance for image registration. The detected features are more robust, even with small rotations. It is recommended that

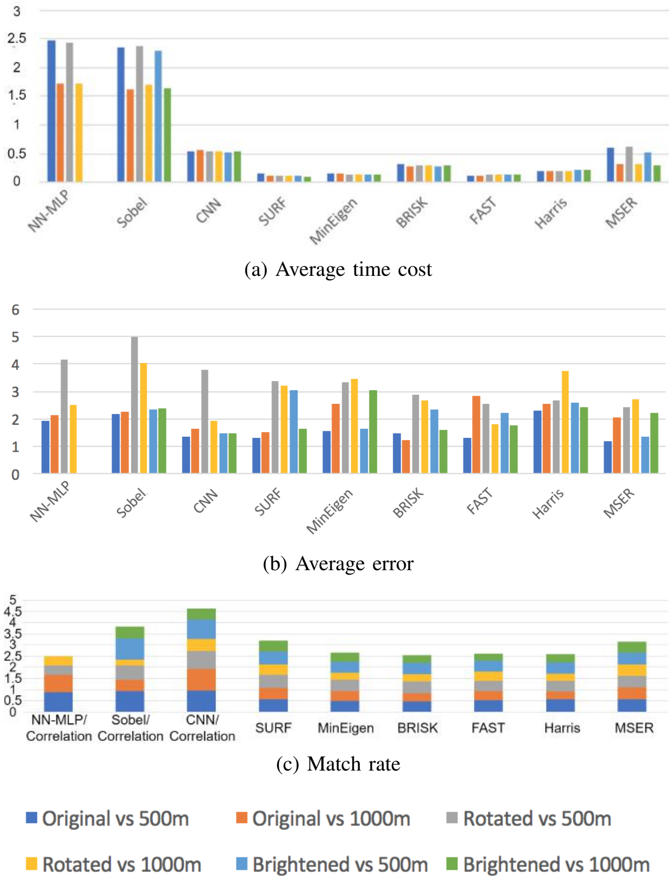


Fig. 6. Match rate, average time cost and average error

CNN should be used combined to a keypoint detection method, in order to mitigate its disadvantages in dealing with too much noise, change in brightness, big rotations or different scales, that are the strongest attributes for keypoint detection methods like SURF, who has the best performance in its category.

As a warning, we must emphasize that employing vision based algorithms have a larger computational cost than just using a GNSS. This implies the use of a more powerful hardware which in turn demands a higher battery capacity. Augmenting the battery will increase its size and hence its weight, cyclically, a heavier vehicle will spend more energy on its maneuvers. Having said that, the idea of owning a navigation system with global positioning capabilities, without the costs of launching and maintaining a constellation of satellites, is quite worth the trouble.

B. Future Steps and Perspectives

Since the results using data of past flight experiments were promising, we aim next embedding these results for a real-time estimation and a full GPS free flight, during sunlight hours and at night time. Besides that, another interesting experiment would be to analyze the behaviour of the method at lower altitudes, to find out what is the limit for sensing moving images.

REFERENCES

[1] P. Lukashevich, A. Belotserkovsky, and A. Nedzved, "The new approach for reliable uav navigation based on onboard camera image processing," in *2015 International Conference on Information and Digital Technologies*, July 2015, pp. 230–234.

[2] J. R. G. Braga, H. F. C. Velho, G. Conte, P. Doherty, and . H. Shiguemori, "An image matching system for autonomous uav navigation based on neural network," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Nov 2016, pp. 1–6.

[3] G. Conte and P. Doherty, "An integrated UAV navigation system based on aerial image matching," *IEEE Aerospace Conference Proceedings*, 2008.

[4] ——, "Vision-based unmanned aerial vehicle navigation using geo-referenced information," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, p. 387308, Jun 2009. [Online]. Available: https://doi.org/10.1155/2009/387308

[5] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, Dec. 1992. [Online]. Available: http://doi.acm.org/10.1145/146370.146374

[6] B. Zitov and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977 – 1000, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885603001379

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[8] G. Loianno, M. Watterson, and V. Kumar, "Visual inertial odometry for quadrotors on se(3)," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1544–1551.

[9] G. Balamurugan, J. Valarmathi, and V. P. S. Naidu, "Survey on uav navigation in gps denied environments," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, Oct 2016, pp. 198–204.

[10] A. d. O. P. Barcelos, F. S. Vidal, and P. F. F. Rosa, "Active stereoscopic camera to build an occupancy grid for autonomous navigation," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, June 2014, pp. 1162–1167.

[11] F. S. Vidal, A. d. O. P. Barcelos, and P. F. F. Rosa, "Slam solution based on particle filter with outliers filtering in dynamic environments," in *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, June 2015, pp. 644–649.

[12] T. Lindeberg, *Scale Selection*. Boston, MA: Springer US, 2014, pp. 701–713. [Online]. Available: https://doi.org/10.1007/978-0-387-31439-6_242

[13] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2548–2555. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126542

[14] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint." in *CVPR*. IEEE Computer Society, 2012, pp. 510–517. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#AlahiOV12