A Self-Adaptive Approach for Autonomous UAV Navigation via Computer Vision

Gabriel Fornari[1][0000−0002−7691−113X], Valdivino Alexandre de Santiago Júnior[1][0000−0002−4277−021X], and Elcio Hideiti Shiguemori[2][0000−0001−5226−0435]

[1] Instituto Nacional de Pesquisas Espaciais, Avenida dos Astronautas, 1758, Jardim da Granja - 12227-010, São José dos Campos, SP - Brasil
{gabriel.fornari,valdivino.santiago}@inpe.br
[2] Instituto de Estudos Avançados, Trevo Coronel Aviador José Alberto Albano do Amarante, 1, Putim - 12228-001, São José dos Campos, SP - Brasil
elcio@ieav.cta.br

Abstract. In autonomous Unmanned Aerial Vehicles (UAVs), the vehicle should be able to manage itself without the control of a human. In these cases, it is crucial to have a safe and accurate method for estimating the position of the vehicle. Although GPS is commonly employed in this task, it is susceptible to failures by different means, such as when a GPS signal is blocked by the environment or by malicious attacks. Aiming to fill this gap, new alternative methodologies are arising such as the ones based on computer vision. This work aims to contribute to the process of autonomous navigation of UAVs using computer vision. Thus, it is presented a self-adaptive approach for position estimation able to change its own configuration for increasing its performance. Results show that an Artificial Neural Network (ANN) presented the best performance as an edge detector for pictures with buildings or roads and the Canny extractor was better at smooth surfaces. Moreover, our self-adaptive approach as a whole shows gain up to 15% if compared with non-adaptive methodologies.

Keywords: Unmanned Aerial Vehicles · Computer Vision · Autonomous Navigation · Self-Adaptive.

1 Introduction

It is undeniable that Unmanned Aerial Vehicles (UAVs) are now part of human life. Also called drones, the UAVs emerged in 1916 as flying bombs [8]. Today, different sizes and shapes of drones are built and used for services that encompasses inspection, surveillance, environmental monitoring, cargo transportation and attacks on ground targets [11].

UAVs can be manually and remotely operated by radio control, but can also be autonomous. When a UAV is autonomous, its control system must be able to guide and navigate without human intervention. The most common approach of navigating is using GPS (Global Positioning System), but, given intrinsic problems with this system, such as signal failure in GPS denied environments and malicious attacks, new approaches have been emerging as alternatives, such as the one based on computer vision [11, 1].

In this approach, the information provided by cameras is used to compute the position of the UAV. Although it has certain advantages, such as low cost, low weight and low power consumption considering some UAV models, the approach faces other problems, such as luminosity effects, land coverage and sensor resolution [1]. Within this area, several works propose the use of different methodologies and image processing algorithms to safely and correctly compute the position of the UAV without using GPS. Although some of them propose data fusion for improving performance, none of them presents the idea of adeptness.

Organisms immersed in dynamic environments may have adaptation as an essential advantage for their survival. Dynamic operating environments have characteristics that can not be controlled and that can constantly and quickly change. In such environments, the need of an organism change its own behavior may improve its performance. In the scientific literature, this process is called from different names, such as self-adaptive systems [13], autonomic computing [10], and Dynamically Reconfigurable Systems (SRDs) [16, 7].

In this sense, this work presents a new self-adaptive methodology for autonomous navigation of UAVs based on computer vision. The approach, called Machine Learning for Autonomous Unmanned aerial Vehicle via computer Vision (MLAUVV), is able to configure itself according to different environments, choosing the most suitable image processing algorithms.

This paper is structured as follows. Section 2 discusses studies related to the area of computer vision applied to UAVs. Section 3 presents in detail the proposed methodology. Section 4 presents the main results and analysis based on such results. Concluding remarks are in Section 5.

2 Related Work

The research in navigation based on computer vision applied to UAVs has started in the nineties of the last century [3]. Since then, lots of works have proposed several kinds of solutions based on computer vision to solve different kinds of problems, e.g. obstacle detection, landing and target tracking. Two surveys present several discussions about computer vision algorithms applied to UAVs [11, 1]. Here, the focus will be studies related to pose estimation of UAVs using vision-based systems.

One way of estimating the position of the UAV is through Simultaneous Localization And Mapping (SLAM), which aims to build a map of the environment and, at the same time, estimate the position of the robot. To do that, the robot should return several times to the starting point in order to reduce drift errors, which can be a problem when low endurance drones are used in large environments [15].

Other pose estimation techniques are visual odometry [6, 4] and template matching [20, 15, 3]. The major difference between them is that the former computes the relative position and the latter the absolute position. The relative position is calculated based on overlapped images taken by the UAV, while the absolute position is estimated comparing the UAV image with a reference image (e.g., a satellite image). Below, some papers that use these techniques are discussed.

In [6], visual odometry and template matching techniques were fused together in order to reduce error. As the reliability of odometry decreases on large distances due to small errors caused by relative displacements of successive images, template matching is used to compensate the error. Besides that, Digital Elevation Model (DEM) and high resolution images were used as input information. The algorithm to perform the matching was Normalized Cross Correlation (NCC) and Robust-Oriented Hausdorff Measure (ROHM).

Based on the idea of [6], Conte and Doherty use visual odometry and template matching as an alternative to GPS [4]. The template matching algorithm for odometry was the Kanade-Lucas-Tomasi (KLT), a feature extractor-based method. The template matching is composed by Sobel edge extractor and cross correlation. Since cross correlation is sensitive to rotation and scale changes, Inertial Sensors (INS) onboard the aircraft were fused using Kalman Filter and used to correct the orientation and scale of the images. In another paper [5], the authors improve their methodology by changing some algorithms, such as cross correlation for NCC, and removing Sobel edge extractor before correlation.

In [20], the Phase Correlation (PC) algorithm is tested and compared with NCC and Mutual Information (MI). The authors claim that the PC algorithm presents better results due to its illumination-invariant feature, which is able to ignore the luminance effects between images. The goal of this work is more related to the advantages of the PC algorithm rather than UAV navigation.

In [3], the authors present a new way to perform the correlation between the UAV image and the reference image by using Scale Invariant Feature Transform (SIFT) and RANdom SAmple Consensus (RANSAC). The main advantage of the SIFT algorithm over cross correlation is its scale and rotation invariance. Again, the focus of the paper is more related to the problem of matching images instead of proposing a complete methodology for UAV navigation.

In [14], the authors present a matching algorithm based on class histograms and classified reference maps. The UAV images are segmented and classified by a neural network into grass, asphalt, and house. Based on the classified image, histograms are computed and compared to histograms obtained from the reference images. The authors claim that the methodology is invariant to rotation but not to scale and also that the impact of luminosity is reduced.

The Simulated Annealing meta-heuristic is used to enhance the results estimated by template matching and INS in [15]. The first is computed using Zero-mean Normalized Cross Correlation (ZNCC) between a UAV image and a reference image, and the last is obtained from sensors onboard the aircraft.

All such previous studies do not present a self-adaptive approach to select the most suitable computer vision algorithms according to environmental conditions. For example, information given by DEMs will be useless in flat regions, as well as optical sensors will not work well during the night. In this sense, this work proposes a self-adaptive approach, which is able to change the current running algorithm aiming to increase performance. Although the approach is general and can be used combining different techniques, in this work the template matching is explored observing its performance under the change of land covers.

3 Self-adaptive Approach for Autonomous UAV Navigation

The self-adaptive approach proposed in this work, called MLAUVV, is based on Rice's abstract model for algorithm selection [17]. His model is based on one question: given a whole set of algorithms, which one will best solve a given problem under specific circumstances? Thus, five topics, called by Rice "spaces", should be used to outline the problem of algorithm selection and hence answer the proposed question. The relationship between them are shown in Figure 1. Furthermore, these spaces are here defined in the context of autonomous navigation via computer vision. More specifically, within the context of template matching based on optical images.

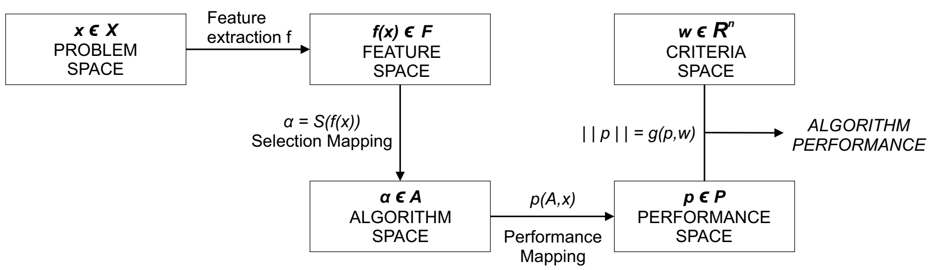

Fig. 1. Abstract model proposed by John Rice explaining the problem of algorithm selection. Adapted from [17].

Problem Space: the set of instances of a problem class. The problem chosen in this work is to estimate the position (latitude and longitude) during the flight of the UAV using only template matching. More specifically, the problem can be stated as finding the correct matching between the reference image and the captured image. One instance of this problem would be all the information gathered by the UAV sensors at an instant of time, which include the UAV image.

Algorithm Space: the set of algorithms that can be used to tackle the problem. In this paper, the template matching methodology was chosen (see [4, 2]). Thus, two different filters (average and Gauss) and three edge extractors (Canny [18], Sobel [18] and a Neural Network [2]) were selected to compare their performances based on the works [4, 2]. To compute the matching, the normalized cross correlation algorithm was elected.

Performance Space: the set of possible performance metrics. In this work, the euclidean distance between the estimated position and the GPS position was chosen. Figure 2 shows how the error is computed.

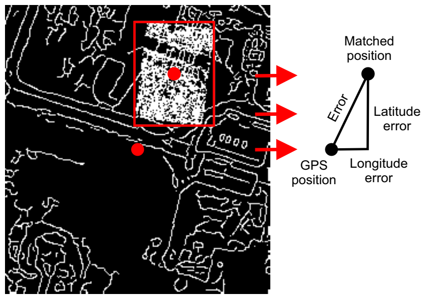

Fig. 2. Error calculation between the position given by the GPS and the position given by the template matching.

Feature Space: the measurable features that characterize the selected problem. In the case of autonomous navigation based on computer vision, it can be divided into some classes, such as environmental conditions, characteristics of the sensor, attitude of the UAV and land cover, being the latter chosen in this work. To analyze the type of soil that the UAV was flying over, 8 properties were extracted from the images: average of the pixels, standard deviation, entropy, maximum and minimum pixel value, difference between the maximum pixel and the minimum one, percentage of borders using Canny algorithm and using Sobel algorithm.

Criteria Space: a set of weights to adjust the relative importance of the performance metrics. In this work, there is only one performance metric, so this space is not needed.

Based on Rice's abstract model, the next step is to test all algorithms to realize about their performances. Once each algorithm has its own performance calculated for an instance of the problem (see Performance Mapping in Figure 1), the Selection Mapping must be able to correctly link a given instance to its best algorithm. To do that, the use of machine learning techniques is proposed; two in particular: neural networks and Bayes networks. Finally, the machine learning techniques can be used to select new instances of the problem maximizing the performance of the overall system.

Now, the self-adaptive approach can be summarized in Figure 3. In this work, the features of the problem are properties of the image captured by the UAV. Based on these properties, an already trained machine learning technique, which can be a Bayes network or a neural network, is able to decide which algorithm is better given the current position of the UAV. After the decision, this approach chooses the best algorithm (filter and edge detector) to estimate the UAV position.

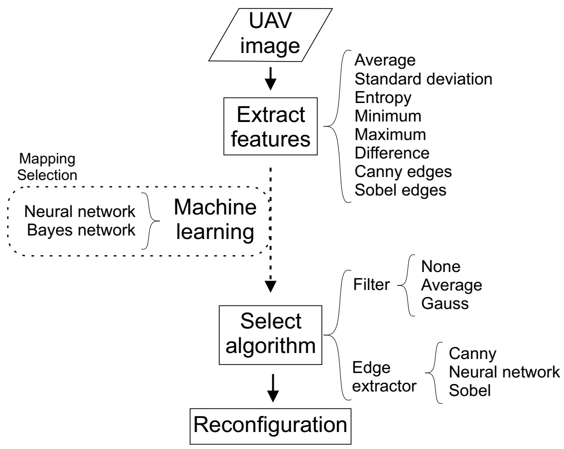

Fig. 3. The MLAUVV approach based on Rice's abstract model and machine learning techniques.

4 Evaluation

In order to verify the proposed self-adaptive methodology, tests using real UAV flight data were performed. The data used in this work was obtained by an airplane and a UAV. The airplane provided the reference image that covers the area of civil contingency agency in Revinge, Sweden. It depicts an area of approximately 1 km^2 and has spatial resolution of 0.5 $meters/pixel$. The UAV flew in a portion of the same area, covering a small part of the reference image. The flight was done by a Yamaha R-MAX at a velocity of 3 m/s at approximately 60 m above the ground. The camera was fixed in a gimbal and adjusted to point nadir. The captured images have resolution of 288x360 $pixels$ and spatial resolution of approximately 0.12 $meters/pixel$ width and 0.11 $meters/pixel$ height.

Based on Rice's model, each image (instance of the problem) has its best algorithm. Initially, a mapping was done in which each image pointed to a specific algorithm. The problem was then to train a classifier able to correctly link each instance to a given algorithm. Then, instead of adopting one algorithm per image, one algorithm was adopted for a set of images.

Now, Rice's abstract model (see Section 3) can be applied in order to train the machine learning techniques again. The only thing that will change is the performance metric because now it should consider a certain amount of images instead of only one. It is important to note that the best algorithm will be chosen for a group of images and not anymore for single ones. It means that the best algorithm can be one for a single image but another for the entire group. The new performance metric chosen was the accumulated error, calculated by summing the error of each separate picture of a given class.

Since the UAV takes pictures while moving, it is expected that the ground changes over time, e.g., at a first moment, the UAV can be flying over a house but later it can be flying over a lake. Based on these changes, the pictures were grouped.

Four groups were chosen in this work tanking into account the presence of buildings, roads or none of them. If a picture presents a house or other building, it is classified in the group “building”. If not, but presents a crossroad, it is classified as “crossroad”. If the picture neither shows a building nor a crossroad but if it presents a road, it is classified in the group “road”. If the picture does not show anything discussed before, it is considered as “smooth”. The images were chosen based on a visual inspection.

Basically, what was made so far is depicted in Figure 4. First, the error of each picture for each one of the algorithms was computed (left side of Figure 4). Besides that, the features were also extracted for each one of the images (right side of the Figure 4). Both information were placed together in a table. Based on the table, the accumulated error was calculated for each of the classes discussed above (building, crossroad, road and smooth).

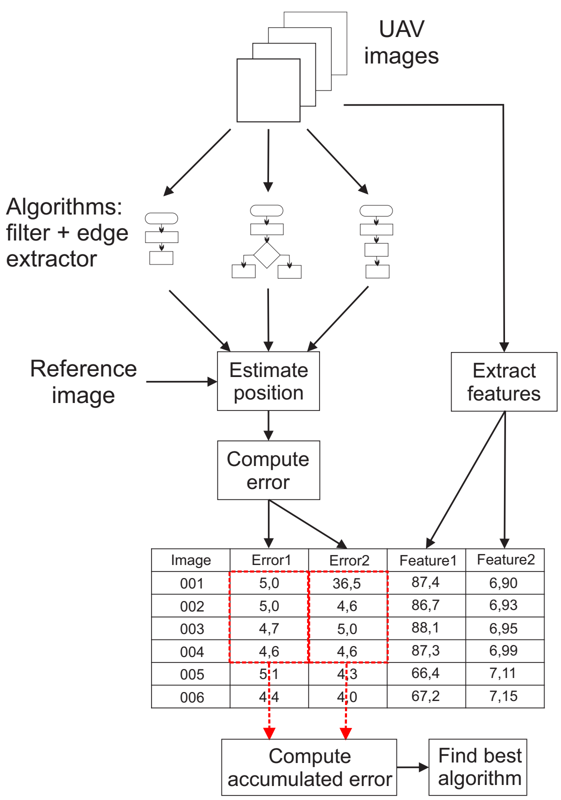

Fig. 4. Flowchart depicting the steps developed to elect the best algorithm.

Table 1 shows the accumulated error for each algorithm. As can be seen, the best algorithm for the classes building, crossroad and road is the same and is composed by the average filter and the neural network edge extractor. On the other hand, the algorithm composed of no filter and the Canny edge extractor claims to be the best in the class smooth.

Table 1. Accumulated error of the 9 algorithms for each class of images.

Algorithm			Accumulated error (m)			
Id	Filter	Edge	Building	Crossroad	Smooth	Road
1	Gauss	Canny	3290	4393	6455	5047
2	Gauss	Neural network	4955	4551	7587	4407
3	Gauss	Sobel	1495	1066	7708	4085
4	Average	Canny	3159	4393	6419	5043
5	Average	Neural network	569	462	6708	3331
6	Average	Sobel	1435	1092	7680	3543
7	None	Canny	3684	4592	5930	5129
8	None	Neural network	929	985	6882	3728
9	None	Sobel	1867	1767	7809	4210

Based on the results of Table 1, it is possible to choose the best algorithm for a certain performance metric and to train machine learning techniques. These steps are depicted in Figure 5. First, the images were splitted according to the classes defined previously. For each class, the algorithm with best performance, i.e., with the smallest error was selected. Finally, the classifiers were trained using the features extracted from the pictures.

In this work, 520 images were selected and grouped into the four classes described above. The images were splitted into two groups, being the first used for validation of the classifiers and composed by 30 % of the original amount of images and the second used for training, being composed by the remaining amount (70 %). Using only images from the second group, the training was made through cross correlation for the whole group (70 %), and for 50 % and 30 % of the original amount of pictures.

The architecture of the Bayes network [12] was built using a genetic algorithm while its conditional probabilities were built with the simple estimator. Among other configurations, the Multilayer Perceptron (MLP) [9] trained with 21 neurons in the hidden layer presented better accuracy in the classification. Using the validation set, the hit ratio was calculated and is shown in Table 2. As it can be seen, in the case of Bayes network the number of hits decreases when the number of images of the training set grows. As cross validation had been used here, the explanation is probably the selection of images for training and validation sets, which was random.

In order to understand the impact of the self-adaptive approach on a path traveled by the UAV, the steps depicted in Figure 3 were followed. To do that, a hypothetical route was created using the 30 % of the images not engaged in the

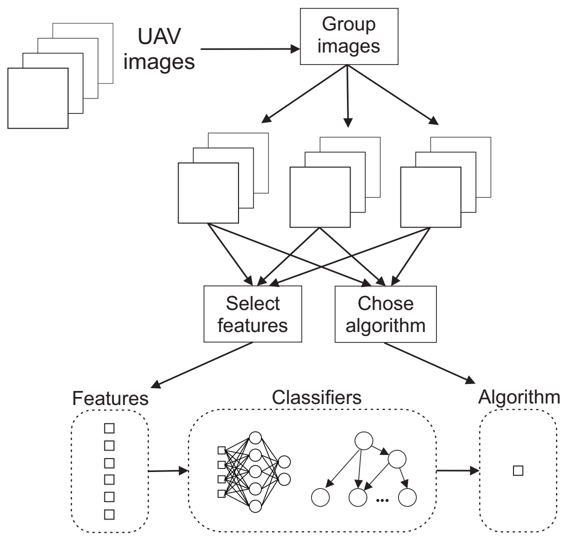

Fig. 5. Flowchart depicting the training of the classifiers.

Table 2. Percentage of Bayes network and multilayer perceptron hits.

	% of hits		
Classifier	30% of training	50% of training	70% of training
Bayes network	97.44%	97.44%	95.51%
Multilayer perceptron	98.08%	98.08%	98.08%

training. The classifiers decided which algorithm should be used for each image based on its features. The time taken for choosing the algorithm is composed by the time of classification plus time of feature extraction. Running on a Intel i7 notebook, the time taken to classify one image was approximately 0.03 s. Figure 6 shows the error related to each picture using both classifiers, Bayes network and MLP, over 4 different terrains. It is interesting to note that, sometimes, due to incorrect classifications, one classifier has better result than the other (e.g., image number 31 of Figure 6).

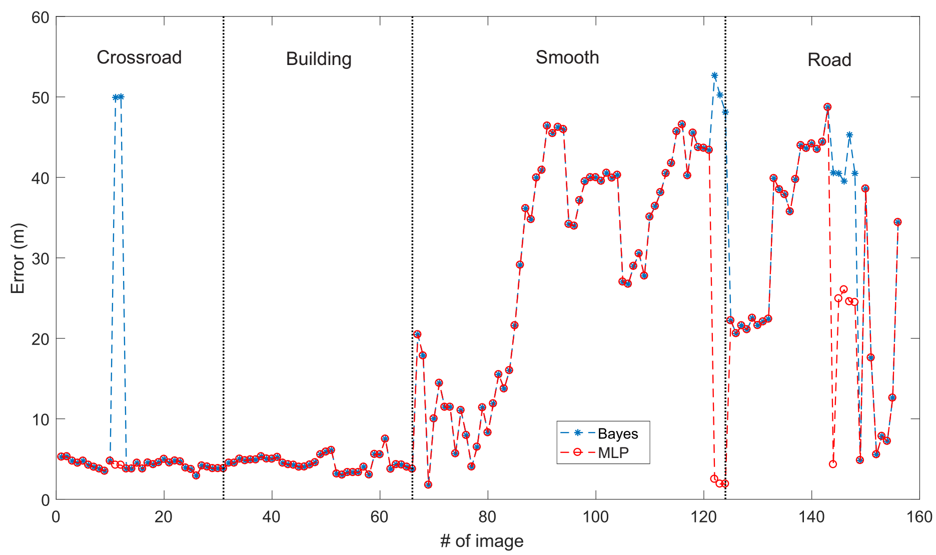

Fig. 6. The error related to each image for both classifiers, Bayes network and MLP. The vertical black line is used split the different image classes.

Observing Figure 6, it is important to notice that the algorithm composed by average filter and Neural Network edge extractor presented good results when the land cover has a building or a crossroad. When the surface present only a road, even choosing the best algorithm (average filter + Neural Network), the error is too high. The same thing occurs when the surface is smooth.

At this point, it would be interesting to compare the results of the self-adaptive approach and the non adaptive approach. To do this, the accumulated error, i.e., the sum of the error of each image, was calculated for all approaches, as shown in Table 3. When comparing the adaptive approach with the 9 non adaptive algorithms it is possible to see the improvement of up 15 % of the result. Besides that, the Bayesian classifier presented a worse performance than the neural network because it wrongly classified some images (e.g., images number 11 and 12 of Figure 6).

Table 3. Accumulated error of the 9 algorithms and the adaptive approach.

Algorithm	Filter	Edge extractor	Accumulated error (m)
1	Gauss	Canny	5547
2	Gauss	Neural network	6502
3	Gauss	Sobel	4337
4	Average	Canny	5478
5	Average	Neural network	3248
6	Average	Sobel	4150
7	None	Canny	5614
8	None	Neural network	3785
9	None	Sobel	4706
Adaptive	Bayes 70%		3114
Adaptive	Perceptron 70%		2776

5 Conclusions

In this work, the MLAUVV methodology was presented. Based on the original problem of algorithm selection proposed by John Rice and on machine learning techniques, the presented methodology aims to automatically chose the best algorithm within the context of navigation control for UAVs using computer vision.

To validate this approach, tests were performed using data from a UAV flight. The data was splitted into classes that took into account characteristics of the land cover, such as the presence of buildings or roads. For each one of these classes the best algorithm, composed by a filter and an edge extractor, was chosen. A MLP and a Bayesian network were trained in order to classify the images in their respective classes. In the end, the results obtained using the adaptive approach were compared to the results using non adaptive approaches. An improvement of 15 % was reached. Furthermore, for the set of images and the set of algorithms explored in this work, the algorithm composed by average filter and Neural Network edge extractor was set as the best when the land cover has a building, a crossroad or a road. When the surface is smooth, the best algorithm is composed by the Canny edge extractor and no filter.

As future work, four topics will be explored in ore detail. First, new filters and edge extractors will be added, such as the median filter and partial area effect [19]. Further work will be done concerning machine learning techniques, and Support Vectors Machines and Decision Trees are the next goals in this topic. New performance metrics will also be explored, such as processing time and memory use of the algorithms. Finally, new tests will be performed using data from other flights.

Acknowledgments. The authors would like to thank José Renato Garcia Braga for its collaboration and discussions during the development of this work and to Department of Computer and Information Science (IDA) of Linköpings Universitet for providing the images used in this work. Gabriel Fornari would like to acknowledge the scholarship provided by CNPq under the process number $140694/2016 - 1$. This work is partially supported by the Swedish Research Council (VR) Linnaeus Center CADICS, ELLIIT, and the Swedish Foundation for Strategic Research (CUAS Project, SymbiKCloud Project).

References

1. Al-Kaff, A., Martín, D., García, F., de la Escalera, A., María Armingol, J.: Survey of computer vision algorithms and applications for unmanned aerial vehicles. Expert Systems with Applications **92**, 447–463 (feb 2018). https://doi.org/10.1016/j.eswa.2017.09.033
2. Braga, J.R.G., Velho, H.F.C., Conte, G., Doherty, P., Shiguemori, E.H.: An image matching system for autonomous UAV navigation based on neural network. In: 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV). pp. 1–6. IEEE (nov 2016). https://doi.org/10.1109/ICARCV.2016.7838775
3. Cesetti, A., Frontoni, E., Mancini, A., Ascani, A., Zingaretti, P., Longhi, S.: A Visual Global Positioning System for Unmanned Aerial Vehicles Used in Photogrammetric Applications. Journal of Intelligent & Robotic Systems **61**(1-4), 157–168 (jan 2011). https://doi.org/10.1007/s10846-010-9489-5
4. Conte, G., Doherty, P.: An Integrated UAV Navigation System Based on Aerial Image Matching. In: 2008 IEEE Aerospace Conference. pp. 1–10. IEEE (mar 2008). https://doi.org/10.1109/AERO.2008.4526556
5. Conte, G., Doherty, P.: Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information. EURASIP Journal on Advances in Signal Processing **2009**(1), 387308 (dec 2009). https://doi.org/10.1155/2009/387308
6. Dong-Gyu Sim, Rae-Hong Park, Rin-Chul Kim, Sang Uk Lee, Ihn-Cheol Kim: Integrated position estimation using aerial image sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(1), 1–18 (2002). https://doi.org/10.1109/34.982881
7. Ferguson, Scott and Siddiqi, Afreen and Lewis, Kemper and de Weck, O.L.: Flexible and reconfigurable systems: Nomenclature and review. In: ASME Design Engineering Technical Conferences, Design Automation Conference, Las Vegas, NV, Paper No. DETC2007/DAC-35745. pp. 1–15 (2007)
8. Filho, P.F.F.S.: AUTOMATIC LANDMARK RECOGNITION IN AERIAL IMAGES FOR THE AUTONOMOUS NAVIGATION SYSTEM OF UNMANNED AERIAL VEHICLES. Ph.D. thesis, Instituto Tecnológico de Aeronáutica (2016)
9. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design, vol. 2. Springer-Verlag, London (2014)
10. Huebscher, M.C., McCann, J.A.: A survey of autonomic computingdegrees, models, and applications. ACM Computing Surveys **40**(3), 1–28 (2008). https://doi.org/10.1145/1380584.1380585
11. Kanellakis, C., Nikolakopoulos, G.: Survey on Computer Vision for UAVs: Current Developments and Trends. Journal of Intelligent and Robotic Systems: Theory and Applications **87**(1), 141–168 (2017). https://doi.org/10.1007/s10846-017-0483-z
12. Kotsiantis, S.B.: Supervised Machine Learning: A Review of Classification Techniques. Informatica **31**, 249–268 (2007). https://doi.org/10.1115/1.1559160
13. Krupitzer, C., Roth, F.M., Vansyckel, S., Schiele, G., Becker, C.: A survey on engineering approaches for self-adaptive systems. Pervasive and Mobile Computing **17**(PB), 184–206 (2015). https://doi.org/10.1016/j.pmcj.2014.09.009

14. Lindsten, F., Callmer, J., Ohlsson, H., Tornqvist, D., Schon, T.B., Gustafsson, F.: Geo-referencing for UAV navigation using environmental classification. In: 2010 IEEE International Conference on Robotics and Automation. pp. 1420–1425. IEEE (may 2010). https://doi.org/10.1109/ROBOT.2010.5509424
15. Liu, X., Wang, H., Fu, D., Yu, Q., Guo, P., Lei, Z., Shang, Y.: An area-based position and attitude estimation for unmanned aerial vehicle navigation. Science China Technological Sciences **58**(5), 916–926 (may 2015). https://doi.org/10.1007/s11431-015-5818-z
16. Lyke, J.C., Christodoulou, C.G., Vera, G.A., Edwards, A.H.: An introduction to reconfigurable systems. Proceedings of the IEEE **103**(3), 291–317 (2015). https://doi.org/10.1109/JPROC.2015.2397832
17. Rice, J.R.: The Algorithm Selection Problem. Advances in Computers **15**(C), 65–118 (1976). https://doi.org/10.1016/S0065-2458(08)60520-3
18. Szeliski, R.: Computer Vision, Texts in Computer Science, vol. 5. Springer London (2011). https://doi.org/10.1007/978-1-84882-935-0
19. Trujillo-Pino, A., Krissian, K., Alemán-Flores, M., Santana-Cedrés, D.: Accurate subpixel edge location based on partial area effect. Image and Vision Computing **31**(1), 72–90 (2013). https://doi.org/10.1016/j.imavis.2012.10.005
20. Wan, X., Liu, J., Yan, H., Morgan, G.L.: Illumination-invariant image matching for autonomous UAV localisation based on optical sensing. ISPRS Journal of Photogrammetry and Remote Sensing **119**, 198–213 (sep 2016). https://doi.org/10.1016/j.isprsjprs.2016.05.016