An Experimental Evaluation of Algorithms for Aerial Image Matching

Ricardo C. Bonfim Rodrigues and Sergio Roberto M. Pellegrino
Eletronic Engineering and Computer Department
Technological Institute of Aeronautics (ITA)
São José dos Campos, Brazil
{rcezar, pell}@ita.br

Abstract — many computer vision systems have been proposing image matching approaches for robots autonomous navigation. These systems have shown good results for ground robots, but aerial vehicles can present much more instability in its camera coordinates during the flight. So in this sense the goal of this paper is to evaluate and compare effectiveness of the image matching algorithms, Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) over a set of aerial images from an Unmanned Aerial Vehicle (UAV). Experimental results show robustness of image matching over different camera perspectives, angles and position, encouraging the use of the computer vision methods for UAV navigation.

Keywords- UAV navigation; image matching; SURF; SIFT.

I. INTRODUCTION

In a general way the human vision is very useful for orientation. Remember a very common practice used by the ancient sailors, the astronavigation, a navigation based on observation of sun, moon, planets and stars. Or still simpler, how many times have you arrived to certain places without knowing the path, just by the using instructions such as, turn right on the supermarket, keep going until big red building, etc. This is the essential idea of this work, an UAV autonomous navigation system based on previous information extracted from digital images with computer vision techniques.

The model of the UAV navigation system consists of a real time approach responsible to capture images during a fly and match them with a map of target points obtained in a previously phase. The goal is to estimate a relative UAV position based on the matched targets, used as waypoints, them once this relative position is given, the aircraft's could have an optimal path to go while meeting certain objectives and mission constraints in regions of target points. This approach is focused on the targets recognition; it means the error given by these methods is not accumulative, once information about the targets as global coordinates is known.

Although the idea seems to be simple, and it is, one of the big problems discussed here is the image matching, in our case used to target points (landmarks representing waypoints) recognition. Many vision systems use image matching for navigation, especially with the Simultaneous Localization and Mapping (SLAM) approach where the goal is to localize a robot in the environment while mapping it at the same time [1-4]. For static environments or small variations they are quite good. But how accurate would they be to our application using aerial images captured from an UAV, where the outside environment is non static and the images to be matched were taken in different times resulting in variation of illumination, angle, scale and deformations between them.

The two competing methods for scale invariant image descriptors, SIFT and SURF were chosen, adjusted and evaluated for the purpose of this work, the results show efficacy and prove theses methods as an efficient solution for image for autonomous UAV navigation problem.

The next section describes the basic ideas of the algorithms used in this work. Section III gives us details about the experiments and results. The sections IV and V describe the analysis and conclusion.

II. SIFT AND SURF BASIC DEFINITIONS

Scale-invariant feature transform (or SIFT) is a robust method proposed by Lowe in 1999 [5] to find keypoints and describe local features. According to Lowe, his descriptors are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. The algorithm is patented in the US and the owner is the University of British Columbia.

The David Lowe's method is based on extremas of Laplacian from image scale space representation. The idea is to smooth the image using Gaussian functions in many scales simulating all zooms images, see Figure 1. When extremas from Gaussian differences are applied to scale space and at least two of simulated zoom images have objects with the same apparent distance along the scales, so the image location is assumed to be invariant to scale and rotation. The SIFT method is performed using a cascade filtering approach, it maximizes the cost of features extraction once the more expensive operations are applied only over locations that has passed by the initial filter.

The SIFT keypoint descriptors contain gradient and orientation information, these features are supposed to be less sensitive to changes such as 3D viewpoints change.

So Lowe proposes an approach based on [6] works which uses a bioinspired method where neurons respond to a gradient at particular orientations and spatial frequency.

As shown in right side of the Figure 2, gradient samples are accumulated into orientations histograms in a 4 x 4 matrix of 16 bins which corresponds to subregions around the keypoint of interest. Each histogram contains information of 8 gradients leading to a 128 element feature for each keypoint. Lowe's uses the Euclidian distance from all these features in a database in order to find the nearest neighbor keypoint for best candidate match, this search is performed using a method called Best-bin-first search based on k-d tree algorithm, where the high probability is calculated with a limited amount of computation. The Euclidian distance of the keypoints feature vector with minimum distance is assumed to be the best candidate.

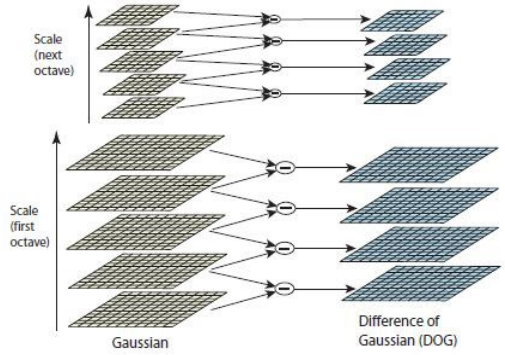

Figure 1 – "For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated" [5].

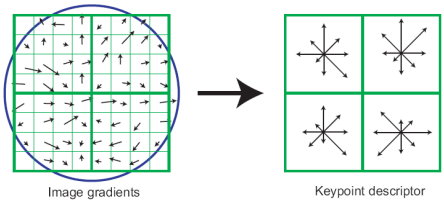

Figure 2 – "A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradientmagnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array" [5].

The Speeded Up Robust Features (SURF) is a high-performance scale and rotation-invariant image keypoints detector and descriptor proposed by Herbert Bay et al in 2006 [7] used to many computer vision tasks such as 3D reconstruction and object recognition. The method was based on some properties of SIFT that uses relative strengths and orientations of gradients to reduce the effect of photometric changes. The idea is to analyze an input image at different scales using Hessian matrices that guarantees scale changes invariance and provide interest points with rotation and scale invariant descriptors.

The SURF keypoint descriptor is based on the intensity of the keypoint neighborhood and provides gradient information similarly to SIFT [7], but with some different properties in order to speed the image matching. To reduce the time for feature and matching this approach exploit integral images for speed, fast computation of box type convolution filters (see Figure 3), distribution of first order Haar wavelet responses in x and y direction rather than the gradient and use only 64 dimensions. According to Bay his methods reduces the time for feature computation and matching, and has proven to simultaneously increase the robustness.

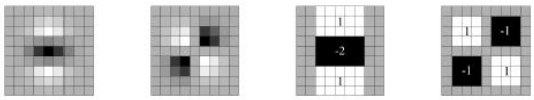

Figure 3 – Half left: the (discretised and cropped) Gaussian second order partial derivative; Half right: approximate second order Gaussian derivatives (box filters). The grey regions are equal to zero [7].

III. EXPERIMENTAL RESULTS

The experiments in this work were conducted in order to evaluate the results from target recognition method and so validate the applicability of the model proposed for UAV navigation using computer vision. This section describes the details of the experiments and results from the SIFT and SURF algorithms.

A. Data base building

The database was build from five videos captured by a small UAV during a monitored flight over an urban area. The aircraft flew in a circular trajectory passing by a similar path; it means many objects can be found in all of the recorded videos.

In order to build our knowledge base eight images were selected to be target points and represent known points in the map, they contains landmarks and objects mostly present in all of the videos, see Figure 4.

A total of 508 sample images were extracted in an interval of 60 frames per second from all the videos during the five laps over the trajectory. Those samples were divided in five sets of images, representing the five laps, and they include samples of all the eight targets and also samples of unknown areas in the interest region. As they were extracted in different times of the flight with different altitudes and path deviation, the objects in the images have variation of perspectives, scale and illumination; see examples in the Figure 5. Those differences are important to validate the matching algorithm in real scenarios, where the environment is non static.

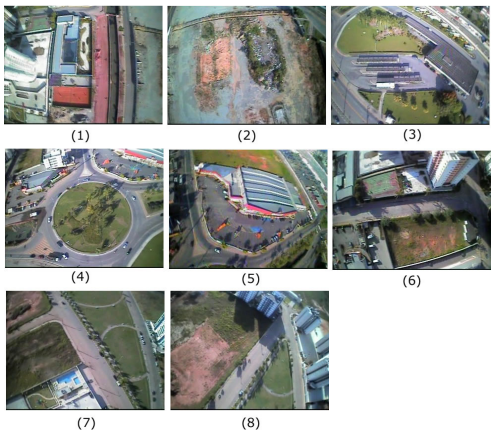

Figure 4 - Set of Target points extracted from the UAV flight.

Figure 5 - Image samples extract during the flights over the interest area. These samples show one of the targets from different perspectives and altitude.

B. *Experimental Settings*

This experiment was conducted using implementations in C++ integrated with the open source library OpenCV (Open Source Computer Vision) version 1.1 which includes an implementation of SURF. A Hob Hess´s implementation of algorithm SIFT is used due to its integration with OpenCV.

The images used in this experiment were captured from a camera model ccd 540 sony atached to the UAV during monitored flights and were resized to 320x240 pixels in order to decrease the processing time. This size was chosen empirically after showing good results in previous experiments. Both target points and sample images described previously have the same size.

C. *Evaluation with image matching algorithms*

The main goal of this experiment is the evaluation of effectiveness and efficiency of the image matching algorithms for target recognition in the proposed model. Having this in mind the algorithm should be capable to recognize the target points in a set of sample images representing the images captured in real time during a flight. The works in this subsection use the eight target points and the sample images described in the data base building subsection. The image matching is performed according to algorithms methods in two main steps, the keypoints localization and keypoints matching. Once we have the matches we try to calculate the homograph matrix between the images. If homograph is found we consider it an image match, in other words, the target point was recognized in the sample image. The homography are computed using RANSAC method to avoid outliers [8].

IV. ANALISES AND RESULTS

All the samples were tested with all the eight target points using SIFT and SURF implementations. The set of test images include samples of targets in the same order that they appear in the videos. The Table 1 and 2 provide information about all targets found (image matches), number of test images, total processing time, number of targets not recognized and the accuracy of the algorithms.

According to Table 1 and Table 2, the SIFT algorithm was capable to match almost two times more images than SURF. Both algorithms reached a high accuracy of instances classified correctly, but this is performed just over the matches (targets found in the set test images). In this way, note that the number of targets not found presented in the Tables 1 and 2 does not indicates the number of matches missed, but the Targets in which the algorithms could not match with any sample. One of the reasons for it was the noise in the test images containing target points, see Figure 6. It means a serious problem to the aircraft that could be lost once it can not found an expected waypoint.

TABLE 1- EXPERIMENTAL RESULTS FROM SIFT ALGORITHM

	# Targ. Found	**# Test images**	**(ms) P. Time**	**# of Targ. not found**	**% Average of Targ. classified correctly**
Lap 1	46	103	1810	0	97,82
Lap 2	38	94	1624	0	94,73
Lap 3	42	91	1426	1	95,23
Lap 4	50	118	2005	0	92
Lap 5	40	102	1764	1	95
Total:	**216**	508	8629	**2**	**Average:** 94,96

TABLE 2- EXPERIMENTAL RESULTS FROM SURF ALGORITHM

	# Targ. Found	**# Test images**	**(ms) P. Time**	**# of Targ. not found**	**% Average of Targ. classified correctly**
Lap 1	33	103	131	0	100
Lap 2	20	94	119	1	100
Lap 3	25	91	115	1	100
Lap 4	23	118	154	1	100
Lap 5	14	102	145	0	100
Total:	115	508	**664**	3	**Average: 100**

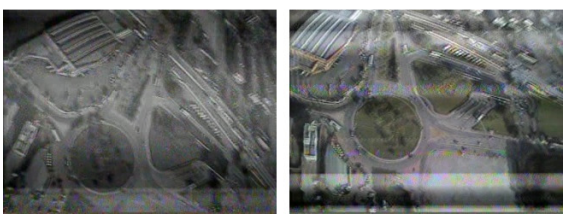

Figure 6 - Example of not recognized noised data base samples including the Target 3.

Although the SURF does not present any false positive over all its matches, its number of matches is smaller than SIFT, therefore SIFT seems to be more robust in this aspect.

Observe that the total time of processing of SURF implementation is far better than SIFT. Taking into account all the 508 images compared to target images, the average time for each pair image comparison is 1.3 ms for SURF against 16.98 ms for SIFT.

A. *Algorithm Validation*

The confusion matrix is a |Y| x |Y| bi-dimensional array where the position *(i,j)* denotes the number of examples of class *i* predicted as examples of the actual class *j*. In other words, each column represents the predicted examples and each row represents the actual examples. Such matrix can be used to compare the classification by combining their elements into more sophisticated formulas like precision and recall. Precision is the ratio between the correctly predicted examples from a given class over the total number of actual examples of such class. On the other hand, recall is defined as the ratio between the number of correctly predicted examples from a given class and the total number of predicted examples for such class.

A perfect Precision score is 1.0 and means that every target found was classified correctly (but says nothing about whether all targets were found) whereas a perfect Recall score of 1.0 means that all target were found (but says nothing about how many targets were classified incorrectly). So we also use a measure that combines Precision and Recall as the harmonic mean of precision and recall, the traditional F-measure or balanced F-score give by the Equation 1:

$$\text{F - measure} = 2.\frac{\text{Precision.Recall}}{\text{Precision} + \text{Recall}} \qquad (1)$$

In Table 3 it is possible to observe the behavior of the SIFT and SURF algorithms with respect to precision, recall, and the F-measure. Note that SURF reached the perfect score in these formulas. Remember that this score was obtained only among the matches, in our case, the targets found. Therefore if we take in account the number of matches and targets found we can say that SIFT implementation obtained relevant results as well.

TABLE 3 – PRECISION, RECALL AND F-MEASURE AVERAGES FOR ALL TARGETS USING SIFT AND SURF

	Precision	**Recall**	**F- measure**
SIFT	0,960	0,94	0,947
SURF	**1**	**1**	**1**

V. CONCLUSION AND FUTURE WORKS

The experiments using image matching methods for targets recognition presented in this work encourage the vision system as a potential solution for UAV autonomous navigation. The outstanding Precision and Recall values as well as the F-measure demonstrate the suitability of image matching algorithm for aerial images.

Although SIFT show more robustness for targets recognition, the SURF seems to be very effective for the UAV navigation system, because of its capacity and accuracy of scene recognition and especially by its low time of processing, which is extremely important in real time application. Another point to be considerate is the importance of defining target points well located in the map, in order to have enough samples for recognition.

New experiments will be formulated and performed in futures works using different databases and other methods of image matching for comparison. A map based on graphs has been studied and will be proposed for data storing regarding to the target points, it will provide flight instructions, orientation and strategies for missing of targets recognized.

ACKNOWLEDGMENT

The author is a scholarship holder from the Coordination for the Improvement of Higher Level and Education Personnel CAPES. Thanks to BRVANT enterprise for giving us the videos used to generate the date base in the experiments.

REFERENCES

[1] T. Bailey and W. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II”. IEEE Robot. Autom. Mag. 13(3), 2006 , pp. 108-117.

[2] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping (SLAM): Part I”, IEEE Robotics & Automation Magazine,vol. 13, no. 2, pp. 99–110, Jun. 2006.

[3] A. Davison, “Real-Time Simultaneous Localization and Mapping with a Single Camera”, in IEEE International Conference on Computer Vision, October 2003, pp. 1403–1410.

[4] J. Kim and S. Sukkarieh, “Real-time implementation of airborne inertial-slam”, Robot. Auton. Syst., vol. 55, no. 1, pp. 62–71, 2007.

[5] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision, v.60 n.2, p.91-110, November 2004.

[6] S. Edelman, N. Intrator, and T. Poggio, "Complex cells and object recognition", 1997.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Vangool, "Speeded-up robust features (surf)", *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, June 2008.

[8] P. Marquez-Neila; J. Garcia Miro, J. M. Buenaposada and L. Baumela, “Improving RANSAC for fast landmark recognition” CVPR Workshops. IEEE Computer Society Conference , June 2008 . pp. 1 – 8.