Comparing ORB and AKAZE for visual odometry of unmanned aerial vehicles

Daniel R. Roos[ab1], Elcio H. Shiguemori [b] and Ana Carolina Lorena[c]

[a]Instituto de Estudos Avançados(IEAv), São José dos Campos, SP, Brazil
[b]Instituto de Ciência e Tecnologia (ICT), Universidade Federal de São Paulo (UNIFESP), São José dos Campos, SP, Brazil

Received on January 01, 2015 / accepted on *****, 2015

Abstract

To navigate autonomously, an Unmanned Aerial Vehicle, also known as drone, must be able to estimate its position during flight. This is usually achieved with the use of the global positioning system combined with an inertial measurement unit. Although widely used for this purpose, the GPS has some disadvantages, like the presence of GPS-denied cluttered environments. A way to circumvent this issue is to employ visual odometry techniques with feature-based methods to compute the aircraft motion and thereby allow the position estimation. We employ two recent features detectors and descriptors - ORB and AKAZE - and compare their accuracy and processing time. The objective is to evaluate their suitability for visual odometry task in real-time embedded systems of UAVs. We have concluded that, although ORB is faster to compute, AKAZE shows a better compromise between speed and performance than ORB for images with low resolution.

Keywords: ORB, AKAZE, UAV, visual odometry, feature matching.

1. Introduction

The development and use of Unmanned Aerial Vehicles (UAV), also known as drones, has grown considerably in recent years. The quantity and quality of the sensors embedded in the UAV have increased as well, especially cameras. Drones have various useful applications, including: agricultural, livestock and environmental monitoring; security and surveillance; search and rescue; entertainment; among others.

In several situations an autonomous navigation ability is desired. For this, the UAV embedded system must know the location of the aircraft. Such information is commonly obtained by some Global System Navigation Satellite (GNSS) - among which the most used is the Global Positioning

[1]E-mail Corresponding Author: danielroos@ieav.cta.br

System (GPS) - combined with Inertial Navigation Systems (INS). Although widely used for this purpose, the GPS has some disadvantages, like the presence of GPS-denied cluttered environments. A way of circumventing this issue is to use computer vision techniques and to extract information from embedded cameras. The collected images are employed to calculate the UAV displacement and to estimate its position during flight, which is called visual odometry.

Feature selection and matching is an important step to the visual odometry. There are several point-feature detectors and descriptors available in the literature, but many of them rely on costly descriptors for detection and matching, making their use impractical for real-time purposes. Some of the most popular feature descriptors are SIFT [2] and SURF [3]. But, in the last years, new descriptors emerged, which are much faster to compute or can be more accurate than SIFT and SURF. In this paper, we will deal with two of them: ORB [4] and AKAZE [5].

ORB (Oriented FAST and Rotated BRIEF) is a fast binary descriptor based on the combination of the FAST (Features from Accelerated Segment Test) keypoint detector [6], [7] and the BRIEF (Binary robust independent elementary features) descriptor [8]. It is rotation invariant and robust to noise. On the other hand, AKAZE (Accelerated-KAZE), a speed-up version of KAZE [9], presents a fast multiscale feature detection and description approach for nonlinear scale spaces. It is both scale and rotation invariant.

In this paper we compare the ORB and AKAZE results regarding their accuracy in identifying feature points and also their processing time. The importance of such a comparison is motivated by the goal of applying algorithms that present a compromise between a good performance required for real-time applications and the accuracy required by the visual odometry task.

2. Visual Odometry

Visual Odometry (VO) is the process of estimating the egomotion of an agent (e.g., vehicle, human, and robot) using the input of a single or of multiple embedded cameras [1]. For aerial navigation, the UAV should capture consecutive image frames with its onboard camera. A sufficient scene overlap should be ensured, so that the VO can operate incrementally to estimate the position of the vehicle by evaluating the changes that the vehicle's movement induces on the images. The VO system works along with the inertial measurement unit (IMU), allowing the UAV to navigate regardless of other navigation systems such as the GPS. Figure 1 shows an example of consecutive images captured by an UAV camera.

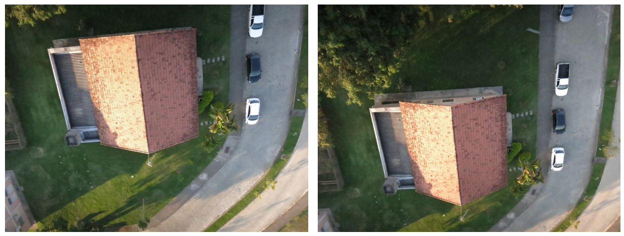


Figure 1 - Examples of consecutive images captured by an UAV camera.

3. Feature Extraction and Matching

To find feature points and their correspondences in the images, features must be detected independently in consecutive images and matched afterwards. This matching is based on some similarity metric between the descriptors of the images. This approach is suitable for the UAV position estimation and navigation, in which there are large motions and viewpoint changes. This process can be divided into three main steps, as follows:

3.1 Feature Detection

During this step, salient keypoints are searched in the image as local features. A salient keypoint is a pattern in the image that differs from its immediate neighborhood in terms of intensity, color, and texture. Some properties for a good feature detector are [1]:

- Localization accuracy - in position and scale;
- Repeatability - re-detection of a large number of features in the next images;
- Computational efficiency;
- Robustness - to noise, compression artifacts, blur, etc;
- Distinctiveness - accurate matching across different images; and
- Invariance - to illumination, rotation, scale, perspective distortion.

In ORB, the detection step is based on the FAST keypoint detector, which is an efficient corner detector suitable for real-time applications due to its computation properties. Since FAST does not include an orientation operator, ORB adds an orientation component to it as described in [4], which is called oFAST (oriented FAST).

AKAZE makes use of Fast Explicit Diffusion (FED) scheme [10] embedded in a pyramidal framework in order to build an accelerate feature detection in nonlinear scale spaces. By means of FED schemes, a nonlinear scale space can be built much faster than with any other kind of discretization scheme [9].

3.2 Feature Description

In this step, the region around each detected feature is converted into a compact descriptor that can be matched against other ones using some similarity or distance measure. A simple example of a feature descriptor is the intensity of the pixels in a patch around the feature point, which represents its appearance. However, the local appearance of the feature is not a good descriptor because it can change due to orientation, scale, and viewpoint changes.

The feature description performed by ORB is based on BRIEF descriptors, which uses simple binary tests between pixels in a smoothed image patch. It is robust to lighting and perspective distortion, but it is very sensitive to in-plane rotation. To circumvent this problem, ORB uses the keypoint orientation information obtained with oFAST to compute oriented BRIEF features, and then uses a learning method to find less correlated binary tests and choose a good subset of them. This leads to better performance in nearest-neighbor applications. This resulting descriptor is named rBRIEF (rotated BRIEF).

The feature description performed by AKAZE is based on a LDB (Local Difference Binary) descriptor [11], which follows the same principle as BRIEF. [5] proposed a modification of LDB, so called M-LDB (Modified-Local Difference Binary) that exploits gradient and intensity information from the nonlinear scale space. Their modification consists of subsampling the grid, instead of using the average of all pixels inside each subdivision of the grid. This is performed in steps that are a function of the scales of the feature, so that the scale-dependent sampling makes the descriptor robust to changes in scale.

3.3 Feature Matching

This step performs the matching between features of two consecutive images using a similarity or distance measure to compare their feature descriptors and choose the best correspondence. As both ORB and AKAZE use binary descriptors, both of them use Hamming distance to efficiently compute the matching.

4. Experiments

The experiments were conducted with two distinct databases of aerial images:

- The first one consist of a set of 1098 images (7360x4912 pixels) obtained from a SONY ILCE-7R camera, with focal length of 45mm, on-board in a fixed wing UAV flying at an average height of 360m over a rural area.
- The second set consists of 148 images (4000x3000 pixels) and was obtained from a Canon PowerShot S110 camera, with focal length of 5.2mm, on-board in a quadcopter flying(b) at an average height of 28m over a region with urban landmarks.

Figure 2 shows an example of rural area image from the first data set. Examples of images from the second data set can be seen in Figure 1.

The amount of detected and matched features have a direct impact on the performance of the algorithms, therefore, in our experiments, the images used in the two datasets were resized for 736x491 and 640x480, respectively, in order allow for real-time processing.

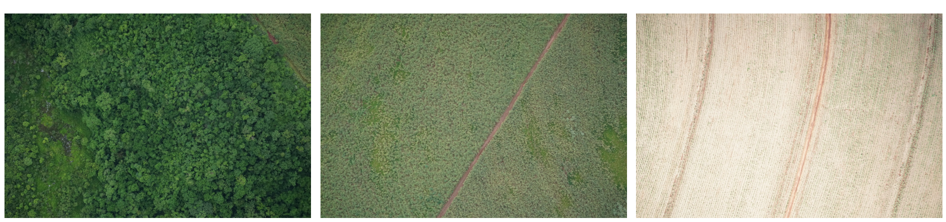


Figure 2 - Example of rural images from the first data set.

Table 1 shows the average number of keypoints matched between two consecutive images for both ORB and AKAZE. As ORB finds a larger number of keypoints, its parameter to define the maximum number of features to retain was set to 1518 and 1780 for the datasets 1 and 2, respectively, in order to keep approximately the same amount average of detected features and facilitate the comparison. Table 2 shows the processing time of the algorithms.

	Dataset 1	Dataset 2
ORB	1403	1632
AKAZE	1518	1780

Table 1: Average number of keypoints matching.

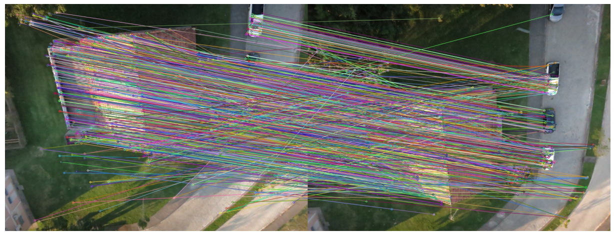
	Dataset 1	Dataset 2
ORB	0.24s	0.25s
AKAZE	0.90s	0.71s

Table 2: Average processing time in consecutive frames.

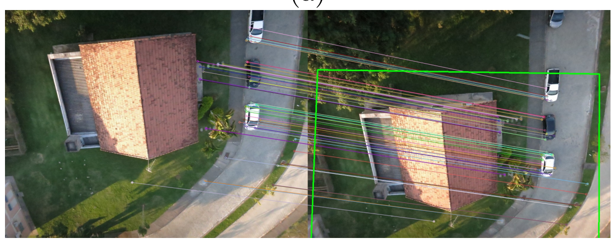
Since the keypoints are detected independently in each image, the feature matching step generates a lot of incorrect matches. Thus, we used RANSAC [12] to eliminate large amount of outliers. Table 3 depicts the average ratio of correct matching (their accuracy) for dataset 1 after applying RANSAC, and Table 4 brings the same information about dataset 2. Figure 3 illustrates the feature matching and outiliers removal steps on the same images from the Figure 1, with circles drawn to represent the keypoints generated in the feature detection and description steps, and lines connecting the matched keypoints.

	Matches	Inliers	Inlier ratio
ORB	27	22	0.81
AKAZE	131	125	0.95

Table 3: Average ratio of correct matching for dataset 1.



(a)



(b)

Figure 3 - Examples of consecutive images with all matches (a) and only inlier matches (b).

	Matches	Inliers	Inlier ratio
ORB	121	111	0.92
AKAZE	293	275	0.94

Table 4: Average ratio of correct matching for dataset 2.

5. Conclusions

As could be seen in Table 2, ORB is faster to compute than AKAZE and the processing time of AKAZE quickly rises with increasing image resolution. However, after filtering the matches and remove outliers, AKAZE presents a greater amount of correct matches than ORB, which could be observed in Tables 3 and 4. For the dataset1, which is composed of rural images, ORB had a fall in the absolute amount of inliers. We could conclude that, although ORB is faster to compute, which is desirable in real-time application, AKAZE shows better compromise between speed and performance than ORB for images with low resolution, typically, with 640x480 pixels. In this paper we compared the performance of ORB and AKAZE on images captured from UAV onboard cameras. In future work, we intend to validate it in a real visual odometry application showing the influence of both of them in the accuracy of position estimation of UAVs.

References

[1] Fraundorfer, F., Scaramuzza, D. Visual odometry: Part II: Matching, robustness, optimization, and applications, IEEE Robotics & Automation Magazine, 19(2): 78-90, 2012.
doi:10.1109/MRA.2012.2182810

[2] Lowe, D.G. Distinctive image features from scale-invariant keypoints, International journal of computer vision, 60(2): 91-110, 2004.
doi:10.1023/B:VISI.0000029664.99615.94

[3] Bay, H., Tuytelaars, T., Van Gool, L. Surf: Speeded up robust features, European conference on computer vision. Springer Berlin Heidelberg, 404-417, 2006.
doi:10.1007/11744023_32

[4] Rublee, E., Rabaud, V., Konolige, K., Bradski, G. ORB: An efficient alternative to SIFT or SURF, IEEE International conference on computer vision (ICCV), 2564-2571, 2011.
doi:10.1109/ICCV.2011.6126544

[5] Alcantarilla, P.F., Solutions, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces, IEEE Trans. Patt. Anal. Mach. Intell, 34(7): 1281-1298, 2011.

[6] Rosten, E., Drummond, T. Machine learning for high-speed corner detection, European conference on computer vision. Springer Berlin Heidelberg, 3951: 430-443, 2006.
doi:10.1007/11744023_34

[7] Rosten, E., Porter, R., Drummond, T. Faster and Better: A Machine Learning Approach to Corner Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(1): 105-119, 2010.
doi:10.1109/TPAMI.2008.275

[8] Calonder, M., Lepetit, V., Strecha, C., Fua, P. Brief: Binary robust independent elementary features, European conference on computer vision. Springer Berlin Heidelberg, 6314: 778-792, 2010.
doi:10.1007/978-3-642-15561-1_56

[9] Alcantarilla, P.F., Bartoli, A., Davison, A.J. KAZE features, European Conference on Computer Vision. Springer Berlin Heidelberg, 7577: 214-227, 2012.
doi:10.1007/978-3-642-33783-3_16

[10] Grewenig, S., Weickert, J., Bruhn, A. From box filtering to fast explicit diffusion, Joint Pattern Recognition Symposium. Springer Berlin Heidelberg, 6376: 533-542, 2010.
doi:10.1007/978-3-642-15986-2_54

[11] Yang, X., Cheng, K.T. LDB: An ultra-fast feature for scalable augmented reality on mobile devices, IEEE International Symposium on Mixed and Augmented Reality, 49-57, 2012.
doi:10.1109/ISMAR.2012.6402537

[12] Fischler, M.A., Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM, 24(6):381-395, 1981.
doi:10.1145/358669.358692