Implementing some algorithms for image registration

Submitted on 22/12/2020
authored by
Saeed Kazemi (Saeed.Kazemi@UNB.ca)
course EE 6553
Prof. Julian Meng (JMeng@UNB.ca)

Index Terms—Image registration, SIFT, BRISK, BRIEF, FREAK, ORB.

Abstract—These days, object registration and tracking in a sequence of images play a significant role in various areas. One of the methods in image registration is feature-based algorithms that in two steps were accomplished. The first step includes finding features of scene and object images. In this step, for reducing the sensitivity of detected features to the scale changes, scale-space is used. Afterward, we attribute feature points obtained in the first step, a description using brightness value around the feature points. In this project, five algorithms such as Binary Robust Invariant Scalable Keypoints (BRISK) and Scale Invariant Feature Transform (SIFT) are implemented in python. These algorithms could use in object detection and tracking.

I. INTRODUCTION

Image registration can be defined as overlaying two or more images taken at different times, by various sensors or sources, or from different points of view. Image registration, also known as image fusion, matching, or warping, can be defined as the process of aligning two or more images [1]. Due to the variety of different types of images, designing a unified and general-purpose image registration approach is very difficult.

The method designed for an image registration application depends on the geometric transformation between the images, the amount of noise damage, accuracy, and application. However, image registration algorithms are usually implemented in 5 steps; preprocessing, feature detection, feature description, feature matching, image re-sampling, and transformation. Figure 1 demonstrates this process.

In the first step, some algorithms like smoothing, de-blurring, edge detection, and segmentation are used to prepare images for registration. Feature detection is the next step. Feature in an image is a region of the image which is conceptually interesting, or in other words, the abstract area of an image is a feature [2]. These features may be a set of corners, interest points, contours, edges, regions, or larger features like blobs. Furthermore, feature detection algorithms must have some properties such as robustness, repeatability, accuracy, generality, efficiency, and quantity [3].

Then, a descriptor vector for each detected feature is would be produced. The descriptor vector should be unique and be robust to some common changes in images. In the next step of the registration algorithms, features are matched using a descriptor vector. If the number of matched features in the object and scene images is more than a threshold, then a proper transformation is applied to warp the scene image to the object image.

In this project, we have performed some famous algorithms in OpenCV-Python. In the next section, we reviewed the background of this image registration. Then in the third section, we reviewed the implementation method in detail, and the precision of algorithms in rotation destruction was examined in the last part.

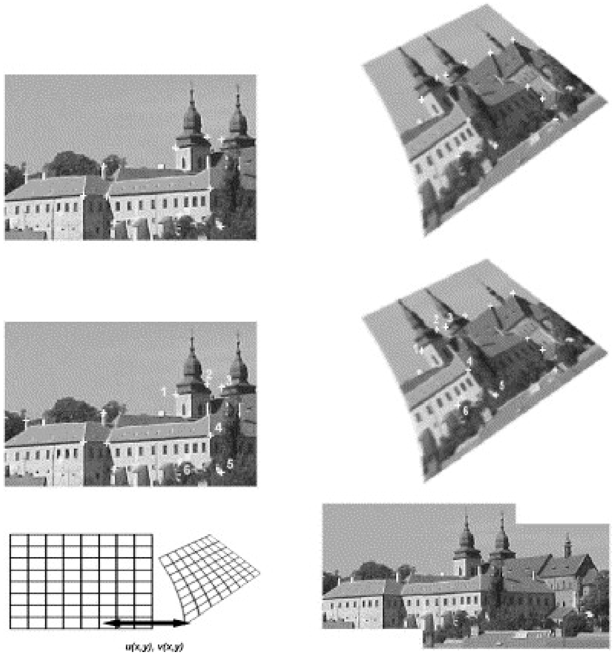

Fig. 1. the five steps of image registration [2]; preprocessing, feature detection, feature description, feature matching, image re-sampling, and transformation.

II. RELATED WORKS

One of the well-known local feature detectors used in image registration algorithms is SIFT (Scale Invariant Feature Transform), which was presented by David Lowe [4]. This algorithm uses Difference of Gaussians (DOG) function for feature detection. DOG function is a good approximation for gradient with low computational time [5]. The descriptor of this algorithm is based on the histogram of the gradient of pixels around the feature.

Bay et al. [6] introduced the Speeded-Up Robust Features (SURF) detector-descriptor. In this algorithm, the Haar wavelet is used to describe the keypoint. Binary Robust Invariant Scalable Keypoints (BRISK) is another binary descriptor, which was introduced by Leutenegger et al. [7]. The algorithm uses the scale-space and the Adaptive and Generic Accelerated Segment Test (AGAST) corner detector [8] to find the features robust to the scale changes. For this purpose, a pyramid of images is organised, and then the AGAST corner detector with the same thresholds is applied on all layers of the pyramid. The scale of each of these features is estimated by interpolation of the score of each feature.

After detecting feature points, a description vector is composed for each one. The description vector is a 512-bit binary code, which is based on comparing the brightness of the feature point with its neighbours. Then, in the end, the description vector in the object image is compared with all of the description vectors of the sensed image through logical XOR operator. If the minimum error was lower than a predefined threshold T, then both features are considered as matched pairs [7].

Alahi et al. introduced the Fast Retina Keypoint (FREAK) descriptor in [9]. This descriptor is similar to the BRISK but with a different sampling pattern which is inspired by the human visual system, the retina. In this sampling pattern, going away from the centre of the pattern, the standard deviation of the Gaussian function becomes larger and the density of pattern points becomes smaller. They showed that the FREAK algorithm requires less memory at runtime and is faster than the BRISK algorithm.

Calonder in [10] introduced Binary Robust Independent Elementary Features (BRIEF). Like FREAK, BRIEF is a descriptor that compare brightness pairs around the keypoint. These pairs were selected randomly. Rublee et al. based on BRIEF, proposed a very fast binary descriptor that was called ORB [11]. They claim that their algorithm was rotation invariant and resistant to noise.

III. THE IMPLEMENTED METHOD

For implementing, the OpenCV library in python that called OpenCV-Python and OpenCV-Contrib-Python were used. Also the python version was 3.9 and the codes are available in the GitHub repository [12]. Each algorithm generally has implemented in three steps, detecting keypoints, computing descriptions, and matching the keypoints. In the first step, the keypoints need to detect in both images. For doing this, the detect method was utilized. Some algorithms like FREAK, and BRIEF did not have the own feature extraction method. For this two algorithms, FAST feature detection [13] was applied. Figure 2 illustrates features along with its direction and scale in an image. As this figure shows, the size of the circles denote the scale of the detected features while the radials denote their direction.

Fig. 2. The location of features which were extracted by SIFT. The size of the circles denote the scale of the detected features while the radials denote their direction.

In the second phase, for each detected keypoints a description was computed. According to each algorithm, the size of feature vectors were different. Table I indicates the size of description vector for each algorithm. Finally, based on these description vectors in both images, we have found the matches points in both images.

TABLE I
The size of the feature descriptor vector in each algorithm

Algorithm Name		Output Size
1	SIFT	128
2	ORB	32
3	BRISK	64
4	BRIEF	32
5	FREAK	64

IV. Experimental Results

Due to the variety of destructive, designing a unified and general evaluation for these algorithms is very difficult. Therefore, it would be good idea to evaluate each transformation. For example, changing scale, changing view point, rotation, etc must be evaluated alone.

In addition for each output in this project, we need to count the correct and incorrect matches for evaluation. These limitations caused that in this project focus on the rotation transformation. Therefore, the implemented algorithm was evaluated on the two images, scene and object images. Although these two images have a particular transformation, scale, and viewpoint changing, covering, we added some rotation on the object images. The scene and object images were shown in Figure 3. The rotation degrees for this project were 30, 60, and 90 degrees. Also for convenience, we investigated only the top 15 match points for each algorithm.

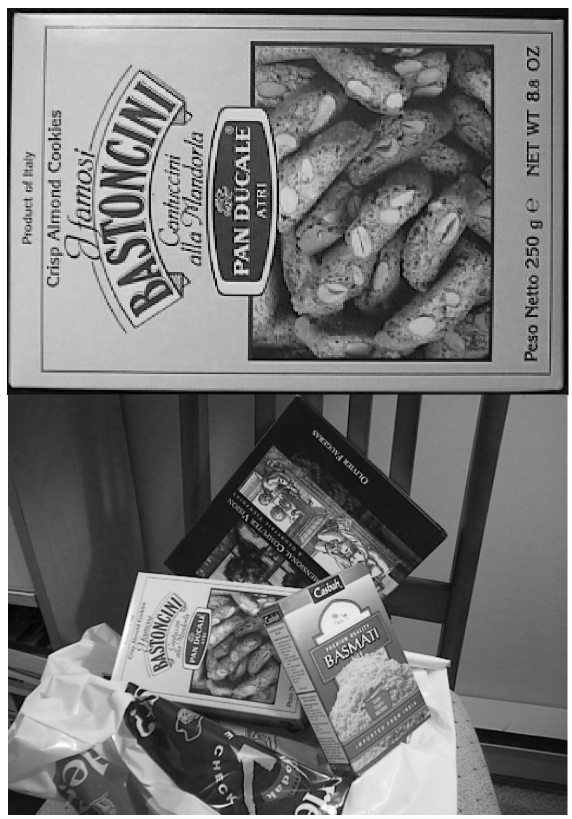

Fig. 3. The scene and object images which were used for evaluation.

The criteria which was used for evaluation is Precision [14]. This criterion is defined as the number of correct matched keypoints relative to the total number of matched keypoints. Below equation indicates the Precision formula.

$$Precision = \frac{\#Correct\ matches}{\#Correct\ matches + \#False\ matches}$$

The precision is a number between zero and one. If the precision of an algorithm is one, it means that all points were properly matched; otherwise, if it is zero it means that all points were falsely matched. In this project, this criteria shows as a percentage number. Table II illustrates the precision for different rotations. This table shows that the BRISK and ORB outperform other algorithms. It is worth mentioning that by increasing the rotation degree, transformations are more destructive, and algorithms have found fewer correct matches. Therefore, the number of correct matched points is decreased when the index of rotation degree is increased.

TABLE II
The result of precision in each algorithm for different rotations.

Destruction	SIFT	BRISK	BRIEF	FREAK	ORB
Original	80%	100%	70%	30%	100%
30 degree	70%	100%	0%	40%	100%
60 degree	70%	100%	0%	30%	100%
90 degree	70%	100%	0%	10%	100%

V. CONCLUSION

In this project, some algorithms were implemented for image registration. Based on the result of our algorithms, the best performance belongs to brisk and ORB. These two algorithms could match about 100 percent corresponding points in both images correctly. Figure 4 shows the result of the BRISK algorithm for different rotations in objects. based on the [7], BRISK used a rotation estimation for each keypoint and all keypoints were normalized by this rotation estimation.

The worse result among these five algorithms belonged to BRIEF. This method could not find even a corresponding match in both images for rotation destruction.

Although SIFT and FREAK used a similar method for estimating the direction of keypoints, these two methods did not have a good perfume in rotation destruction.

VI. REFERENCES

[1] F. P. Oliveira and J. M. R. Tavares, “Medical image registration: a review,” Computer Methods in Biomechanics and Biomedical Engineering, vol. 17, no. 2, pp. 73–93, Jan. 2014, issn: 1025-5842. doi: 10.1080/10255842.2012.670855. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10255842.2012.670855.

[2] B. Zitová and J. Flusser, “Image registration methods: A survey,” Image and Vision Computing, vol. 21, no. 11, pp. 977–1000, Oct. 2003, issn: 02628856. doi: 10.1016/S0262-8856(03)00137-9.

[3] Ali Ismail Awad and Mahmoud Hassaballah, “Image Feature Detectors and Descriptors,” Studies in Computational Intelligence, vol. 630, 2016. doi: 10.1007/978-3-319-28854-3.

[4] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004, issn: 09205691. doi: 10.1023/B:VISI.0000029664.99615.94. [Online]. Available: https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94.

[5] ——, “Object recognition from local scale-invariant features,” in Proceedings of the IEEE International Conference on Computer Vision, vol. 2, IEEE, 1999, pp. 1150–1157. doi: 10.1109/iccv.1999.790410.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346–359, Jun. 2008, issn: 10773142. doi: 10.1016/j.cviu.2007.09.014.

[7] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust invariant scalable keypoints,” in Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2548–2555, isbn: 9781457711015. doi: 10.1109/ICCV.2011.6126542.

[8] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and generic corner detection based on the accelerated segment test,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6312 LNCS, Springer Verlag, 2010, pp. 183–196, isbn: 3642155510. doi: 10.1007/978-3-642-15552-9{_}14. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-15552-9_14.

[9] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast retina keypoint,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 510–517, isbn: 9781467312264. doi: 10.1109/CVPR.2012.6247715.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6314 LNCS, Springer Verlag, 2010, pp. 778–792, isbn: 364215560X. doi: 10.1007/978-3-642-15561-1{_}56. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-15561-1_56.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2564–2571, isbn: 9781457711015. doi: 10.1109/ICCV.2011.6126544.

[12] SKazemii/ImageRegistration. [Online]. Available: https://github.com/SKazemii/ImageRegistration.

[13] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, pp. 105–119, 2010, issn: 01628828. doi: 10.1109/TPAMI.2008.275.

[14] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1615–1630, Oct. 2005, issn: 01628828. doi: 10.1109/TPAMI.2005.188.

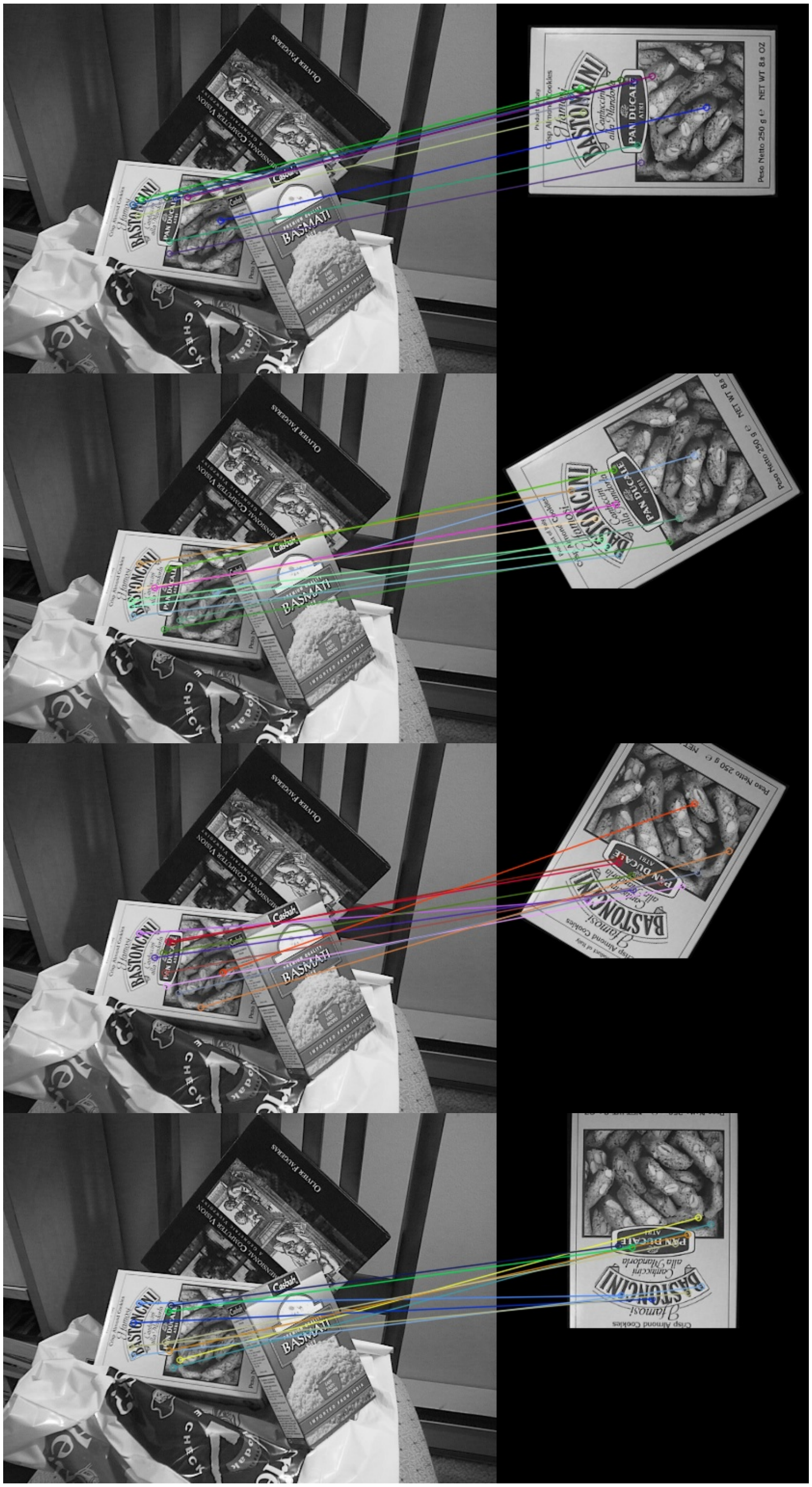

Fig. 4. Match points on both the scene and object images by the BRISK algorithm.