International Journal of Engineering Research and Advanced Technology (IJERAT)

DOI: 10.31695/IJERAT.2023.9.11.1

E-ISSN : 2454-6135

Volume. 9, No.11
November - 2023

Computer Vision Matcher for Extracting and Detection Objects

Ikhlas Watan Ghindawi[1] and Lamyaa Mohammed Kadhim[2]

[1]Department of Computer Science

[2]College of Dentistry

University of Al-Mustansiriyah

Bhagdad, Iraq

ABSTRACT

The research examines the issue of the feature matching and object recognition in two images with the use of brute-force matchers. The suggested framework utilized a number of concurrent algorithms, including BRISK (Binary Robust Invariant Scalable Keypoints), SURF (Speeded-Up Robust Features), ORB (Oriented FAST and Rotated BRIEF), and SIFT (Scale Invariant Feature Transform), for descriptor extraction and feature detection. K-Nearest Neighbors (KNN) algorithm in conjunction with the brute-force approach allows for feature matching. Robust Random Sample Consensus (RANSAC) approach calculates transformation between two successive images using the found matches. As a result, the RANSAC approach is used to enhance the removal of outliers. With the use of OpenCV library, the suggested technique was developed and put into practice. Analyses of the speed at which it executes commands and the precision of the feature matching serve as proof of the system's quality and efficacy.

KEYWORDS: Extraction, Feature detection, Feature matching, Robust Random Sample Consensus.

1 INTRODUCTION

Numerous fields, including medicine, robotics, astronomy, engineering, and others, employ computer vision extensively. Each assignment given to vision-based apps begins with a description regarding the image's attributes. They represent the visually distinct, emphasized, and contrast-added areas of an image, which are typically the object's edges and corners. The term "descriptor" refers to the vector that describes image features. Additional digital image processing and analysis procedures require its computation [1]. Due to the growing interest in image analysis, many feature extraction and detection methods have been developed recently. The comprehensive algorithms SURF [2-3], SIFT [4-5], ORB [6], and BRISK [7] are capable of handling the two tasks. Based upon the similarity between their descriptors, features that are detected then described in the images could be matched. Finding image feature correspondences between at least two images in real-time applications can be difficult, particularly in the case of dealing with large-size and high-resolution images. It's crucial to strike a balance between speed and match quality. With regard to feature matching, the brute-force descriptor matcher [8] employs this method. It uses some distance computations to compare the description of one feature from the first image with the descriptions of every feature from the second image. The resulting pair after that returns the nearest one. The brute-force technique often takes longer but is highly exact due to the vast number of comparisons. By adjusting certain parameters, it could perform better and be better at removing outliers.

Matching features in two images of same planar surface in the space is necessary for object detection. They are connected by a homography, and it is accurate to get rid of outliers in order to estimate the homography matrix accurately. LMedS [9] and RANSAC [10] are two robust outlier detection and removal methods. The brute-force technique is integrated with four extraction and detection methods in this study to offer accurate and quick feature matching. The optimum combination moves on to the following stage of analysis that involves estimating the matrix of the homography with the use of RANSAC algorithm and removing outliers. Visualization of the process of object detection in scene images represents the final task. The main aim of this study is to build a model which enables effective and precise feature matching and object detection in set of related images. It will be rejected in the case where there isn't any similarity between the images. On https://github.com/AJakubovic/Application-for-feature-matching-in-images, one might find the designed application and the necessary files. The structure of this study is as follows. The primary steps in image analysis are outlined in Section II. The suggested algorithm for getting the relevant steps is described in Section III. Experimental results have been displayed in Section IV. Section V, which concludes the study and provides some additional guidance.

2. IMAGE ANALYSIS

In Fig. 1, the suggested algorithm's flow is shown. It is broken down into five sections: homography matrix estimation, object detection, feature extraction, brute-force feature matching, and feature extraction.

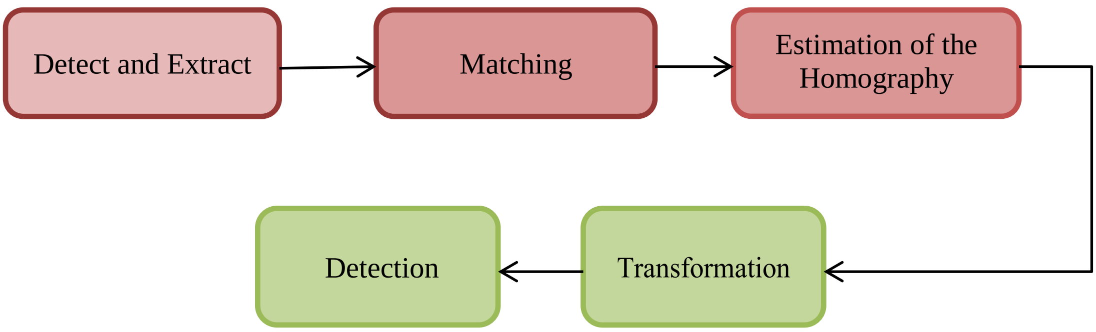

Fig1. The suggested algorithm for the object detection and feature matching

2.1 Feature Detection and Extraction

In the area of computer vision, a set of image features (i.e. keypoints) has been used for interpreting its content. Thus, to find and recognize objects in the multimedia files, image interest point detection is utilized. The algorithms used for feature detection should be repeatable, or able to find the same feature in multiple images, and effective, or able to produce outcomes in close to real-time. In a process of feature extraction, every one of the interest points is described by the matching fixed-length vector (i.e. descriptor). Many algorithms must be tested for producing a feature detection and extraction approach that is effective and accurate enough. In the presented work, four of them are taken into account. ORB and BRISK, two algorithms that use binary bit-string descriptors, execute Hamming distance calculations relatively quickly. Their primary flaw is a reduction in the robustness of image transformations (translation, rotation and scaling). SURF and SIFT, are more robust but take more time to compute since they use Euclidean distance between the descriptors. Those algorithms are compared using the same 1920x1080-pixel-resolution image. The next metrics factors are taken into account while evaluating the algorithm's

DOI : 10.31695/IJERAT.2023.9.11.1

effectiveness: detection time (DT), total time (TT), number of the detected features (NODF), and extraction time (ET). The results that were obtained are shown in Table I. Although SURF and SIFT and other binary descriptor-based algorithms, such as BRISK and ORB, identify many more features, they have been concentrated in same descriptors, and that results in the visual redundancy. SURF and SIFT find fewer features, yet they're specific and recognizable. When using SIFT/SURF descriptors instead of BRISK/ORB descriptors, descriptor extraction takes longer.

2.2 Feature Matching

Finding the correspondence between the descriptors in the images that depict the same object (scene) from various perspectives is what is meant by the matching of described and detected features. In this study, a brute-force algorithm has been utilized for matching the features between two images. Depending on the type of acquired descriptors, it may employ the Hamming distance or Euclidean distance. Brute-force matching is frequently used in conjunction with the kNN and ratio test [11, 12] to remove outliers from the results.

TABLE I. Results For The Process Of Feature Detection And Extraction

Algorithms	NODF	ET (sec)	DT (sec)	TT (sec)
ORB	49,967	0.3720	0.1580	0.530
SIFT	6,899	1.1940	1.6910	1.1940
BRISK	20,289	0.1770	1.270	0.1770
SURF	8,689	0.695 0	0.40	0.6950

2.3 Homography and Object Detection

Assuming that m represents a homogeneous pixel with the coordinates of $[x, y, z]^{T.}$ Consider that a pair of 2-D images from various points of view each depict the same object in plane. Homography H, which represents a non-singular projective matrix mapping points from one image (m1) to corresponding points in second image (m2), allows for the identification of the same features in both images:

$$\mathbf{m_2} = \mathbf{H}\mathbf{m_1} \tag{1}$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{2}$$

The eight matrix elements are typically unknown since 1 of 9 elements is typically assigned some fixed unity value (h33 = 1). Since each of them provides two equations, it results in the conclusion that a minimum of 4 inlier pairs (m1; m2) are needed [13].

3. PROPOSED ALGORITHM FOR OBJECT DETECTION AND FEATURE MATCHING

Two 2-D images of one object that have been taken from various points are needed for feature matching. Images could be represented using actual camera data or a set of sample data. First, 1 of the 4 algorithms that have been listed

DOI : 10.31695/IJERAT.2023.9.11.1

in Section II is used to perform the process of feature detection and descriptor extraction. In an event when k = 2, brute-force matchers are after that merged with the kNN, yielding 2 nearest neighbors in the training image for every one of the descriptors in query image. By using a straightforward selective principle, it aids in removing outliers just as the ratio test. The possible pair is approved or rejected based on ratio of distance between nearest and second-nearest corresponding feature. The match has been excluded from further investigation if the ratio exceeds the threshold, which is typically 0.8. The rest of the matches are inliers, and they are arranged according to how far apart the descriptors are. The key-points in every one of the inliers are after that separated to object and scene vectors based upon the corresponding image, which is then referred to as train or query. Attempting to estimate the homography matrix could be done after separating inlier keypoints. In order to remove more outliers, the robust RANSAC algorithm is also applied. The perspective transformation could be utilized in order to identify corresponding features in the training image for group of the features in the query image, such as corners, if homography H is successfully established. That is the technique for detecting an object.

4. EXPERIMENTAL RESULTS

In this part, suggested algorithm's quality and performance are evaluated with the use of the metrics of the mean error of matching (MEM, equation (3)), number of inlier matches (NOIM), ET, and algorithm speed (S), which is calculated as the ratio of the inlier matches to the ET. It is preferred that mean error of a number of the inlier matches and ei (i = 1,..., n) be Euclidean matching and have a short execution time. In equation (3), n is the SIFT/SURF or BRISK/ORB Hamming distance, respectively.

A. Example 1

The test images used in this illustration are examples from OpenCV directory. A scene image (box in scene) and object image (box) have resolutions of 512x384 and 324x223, respectively. Locality, or robustness to occlusion and clutter, is a key quality of a good feature [14]. In this instance, the scene (train) image has occlusion, meaning that the object (query) image is partially concealed. In Fig. 2 and Table II, respectively, are the results of the processes of object detection and feature matching using Algorithm 1 that were attained.

Table II. Results of Utilizing Proposed Method

Detection algorithm	**NOIM**	**S (NOIM/sec)**	**ET (sec)**	**MEM**
ORB	168	295.7750	0.5680	42.6010
SIFT	94	138.2350	0.680	166.2060
BRISK	70	62.0570	1.1280	84.1570
SURF	119	288.1360	0.4130	0.2110

DOI : 10.31695/IJERAT.2023.9.11.1

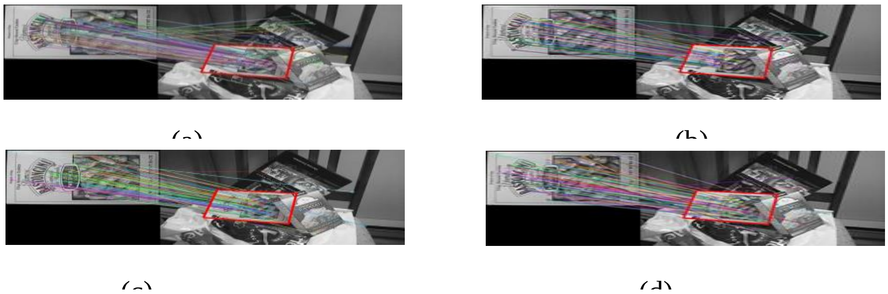

Fig2. Object detection and Feature matching utilizing the ORB (a), BRISK (b), SIFT (c) or SURF (d) algorithms

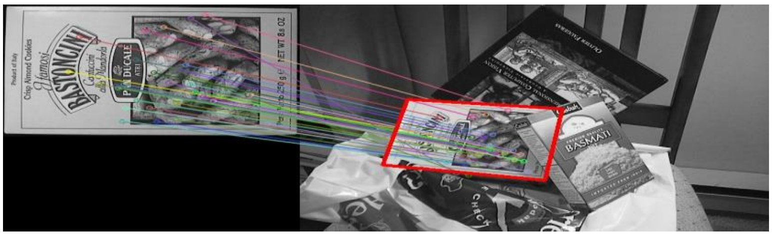

Fig3. Feature matching and object detection with the use of the SURF (optimum number of the matches (47) exhibited)

It has been evident that SURF feature extraction and detection algorithm combined with the brute-force process of matching produces data with the minimum mean error. Object detection in the scene image has been found satisfactory for every employed algorithm in this situation, with SURF and ORB being the fastest (Fig. 2). The number of the inliers matches and ET are low as well, because to the reduced resolution of the example images. Outlier matches (false positives), on the other hand, are also discovered. since matches in Algorithm 1 are ordered by quality, a slightly improved visualisation could be gained by reducing the number of matches revealed. This is carried out in a case of SURF algorithm, in which 47 inlier matches have been displayed in Fig. 3 in 0.395 seconds with a 0.152 mean error.

B. Example 2

In this case study, matching between 2 textual images is displayed. The resolutions regarding the query image and the train image are 101x41 and 1662x786, respectively. The word "see" is represented by the query (object) image. This example presents a word because the train (scene) image has numerous 'e' and 's' letters, including the word 'seemed'. Table III lists the outcomes of the suggested methodology. In this situation, binary descriptor-based algorithms for extraction and detection perform poorly. Two inlier matches are found in the ORB results, yet the homography matrix is empty because a minimum of 4 inliers are needed. BRISK generates 4 matches, yet only 1 is an inlier, and the homography matrix calculation is precise. For the purpose of object detection, the SURF and SIFT algorithms in conjunction with brute-force matching display encouraging results. Six of the 16 matches that were found with the use of SIFT for the process of feature detection and extraction are inliers. Thirty matches are returned by the SURF, of which 13 are inliers. For the accurate homography matrix estimate ($4 < 6$, $4 < 13$) and accurate object detection, the number is sufficient in both scenarios. Due to the object image's low resolution, the example shows a minimal number of matches. However, because the scene image has a high quality, all algorithms' execution times are slightly longer (Table III).

DOI : 10.31695/IJERAT.2023.9.11.1

Table III. Proposed Method Result data

Detection algorithms	NOIM	S (NOIM/sec)	ET (sec)	MEM
ORB	2	0.9470	2.1130	0.0
SIFT	16	5.9840	2.6740	68.3640
BRISK	4	2.6860	1.4890	50.750
SURF	30	11.9520	2.510	0.0520

5. CONCLUSIONS

This study proposes a framework for feature matching and object detection. A total of 4 algorithms of feature detection and extraction have been built in the first stage of the framework design. After that, with specific settings and optimizations, brute-force matchers are utilized for feature matching. The estimation of homography matrix and a suitable perspective transform serve as foundation for the object detection. Two exemplary instances are used to evaluate the suggested framework's efficacy and performance. Various metrics parameters are used for a large variety of images, each with a different resolution and set of features. Those three factors are the processing time, the number of inlier matches, and average matching error. Each of the four-feature extraction and detection algorithms could typically produce results that are adequate, however it is demonstrated through considered examples that SURF is the most accurate and effective algorithm. By combining algorithms for the processes of feature detection, extraction, and brute-force matching, this study makes contributions to object detection in terms of diverse methodologies. Although the existence of outliers is inevitable, the suggested algorithm reduces their number and enables their omission from the visualization section.

REFERENCES

[1]R. Szeliski, *Computer Vision: Algorithms and Applications*. London: Springer-Verlag, UK, 2010.

[2]D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004. [3]D. G. Lowe, "Object recognitionfrom local scale-invariant features," in Proc. International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1150-1157.

[4]H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, pp. 346-359, 2008.

[5]H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded up robust features," in Proc. European Conference on Computer Vision, Graz, Austria, 2006, pp. 404-417.

[6]S. Leutenegger, M. Chli and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in Proc. IEEE International Conference on Computer vision, Barcelona, Spain, 2011, pp. 2548-2555.

[7]E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "Orb: An efficient alternative to sift or surf," in Proc. IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564-2571.

[8]OpenCV modules. http://docs.opencv.org/3.1.0. Accessed: 2017-04-19. [9]M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,", *Communications of the ACM*, vol. 24, pp. 381- 395, 1981.

[10]X. Li and Z. Hu, "Rejecting mismatches by correspondence function," *International Journal of Computer Vision*, vol. 89, pp. 1-17. 2010. [11]OpenCV library. http://opencv.org. Accessed: 2017-05-12.

[12]M. Beyeler, *OpenCV with Python Blueprints*. Birmingham: Packt Pub- lishing Ltd, UK, 2015.

[13]I. Zoghlami, O. Faugeras and R. Deriche, "Using geometric corners to build a 2D mosaic from a set of images", in Proc. EEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, 1997, pp. 420-425.

[14]R. Urtasun, "Computer vision: Image features", Lecture, TTI Chicago, January 2013

DOI : 10.31695/IJERAT.2023.9.11.1