Indian Journal of Image Processing and Recognition (IJIPR)
ISSN: 2582-8037(Online), Volume-1 Issue-1, December 2020

Adopted Image Matching Techniques for Aiding Indoor Navigation

Mohamed Ramadan, Mohamed El-Tokhey, Ayman Ragab, Tamer Fathy, Ahmed Ragheb

Abstract*: Fast and accurate image matching is a very important task with various applications in computer vision and robotics. In this research, we compare the performance of all available feature detection techniques (HARRIS, GFTT, SIFT, SURF, STAR, FAST, ORB, MSER, Dense, and SimpleBlob), feature description techniques (SIFT, SURF, BRIEF, and ORB), and image matching techniques (BruteForce, BruteForce-L1, BruteForce-Hamming, BruteForce-HammingLUT, and FlannBased) against different kinds of geometric distortions and deformations such as scaling, rotation, fish-eye distortion, and shearing. To perform this task, we manually apply different types of transformations on original images and compute the matching evaluation parameters such as the number of keypoints in images, the processing time, and the matching accuracies for each algorithm and we will show that which algorithm is the best more robust against these distortions.

Keywords*: feature detector, feature descriptor, image matching, indoor navigation, OpenCV.

I. INTRODUCTION

Navigation systems have important roles in many vital disciplines. Determining the location of a user relative to the physical environment (e.g. roadway, intersections, and buildings) is an important part of transportation services such as in-vehicle navigation, fleet management, and infrastructure maintenance. In addition, other navigation services are required for locating the position of a user in an indoor physical environment (e.g. airports, shopping malls, public buildings, university campuses). This indoor navigation can support several applications such as user navigation, emergency services (e.g. ambulance, police, etc.), law enforcement, marketing services. Both indoor and outdoor navigation applications require a reliable, trustful, and continuous navigation solution [1].

These applications usually use the Global Navigation Satellite Systems (GNSS). Although these systems usually provide accurate positioning information, they can only do that in GPS-friendly environments. Navigation in GPS-denied environments, such as indoor centers, usually has difficulty maintaining accurate positioning information due to GPS signal blockage and multipath [1].

In the last decade, modern smart devices have multiple sensors (e.g. magnetometer, barometer, gyroscope, accelerometer, camera, Wi-Fi, etc.) which can be used to get an indoor positioning result. The great advantage of using smart devices sensors is their low cost. On the other hand, the main disadvantage of these sensors is their low accuracy [2].

Several researchers have addressed possible methodologies and techniques for combining various sensors results in a more homogenous and more reliable outcome in order to make indoor navigation more precise [2].

Nowadays, cameras can be seen everywhere, from the smartphone in our pocket to the surveillance cameras in buildings to the scientific microscopic cameras and so on. So, the field of computer vision has seen a rapid rise in the recent past, with the development of a wide variety of techniques to accomplish certain tasks [3].

Feature detection, description, and matching play an important role in several computer vision applications. Thus, they have received considerable attention in the last decades. Several feature detectors and descriptors have been addressed in the literature with a variety of definitions for what kind of interesting points [4].

II. IMAGE FEATURES DETECTION, DESCRIPTION, AND MATCHING

This section introduces brief notes about the basic algorithms and mathematical concepts for detecting and describing image features.

A. Overview of detection techniques

Features detection is the first processing operation in the computer vision that extract the interest points required for the next processing steps [5]. An interesting point in an image should be clear, well-defined, stable under local and global perturbations in the image domain [6].

- Harris corner detector is rotation-invariant, which means, even if the image is rotated, we can find the same corners. But if the image is scaled the corners may not be detected as corners. So, the Harris corner detector is not scale-invariant [7].
- GFTT (Good Features to Track) detector was made with a small modification to Harris corner detector [8]. As the Harris corner detector, the GFTT method is also rotation-invariant, but not scale-invariant [7].
- SIFT (Scale Invariant Feature Transform) is one of the most used detectors. This method tests each pixel in the image with its eight neighbors as well as nine pixels in the scale above and below. This method is rotation-invariant, and scale-invariant [9].

Manuscript received on 27 November 2020 | Revised Manuscript received on 03 December 2020 | Manuscript Accepted on 15 December 2020 | Manuscript published on 30 December 2020.

***** Correspondence Author

Mohamed Ramadn, Assistant Lecturer, Public Works Department, Ain Shams University/ Faculty of Engineering/ Cairo, Egypt.

Mohamed El-Tokhey, Professors, Public Works Department, Ain Shams University/ Faculty of Engineering/ Cairo, Egypt.

Ayman Ragab, Professors, Public Works Department, Ain Shams University/ Faculty of Engineering/ Cairo, Egypt.

Tamer Fathy, Professors, Public Works Department, Ain Shams University/ Faculty of Engineering/ Cairo, Egypt.

Ahmed Ragheb, Associate Professor, Public Works Department, Ain Shams University/ Faculty of Engineering/ Cairo, Egypt.

- The SIFT algorithm was comparatively slow and the recent applications needed a more speeded version. The SURF (Speeded Up Robust Features) algorithm is based on the same principles as SIFT but with some approximations to make the process much faster [10]. Like the SIFT technique, the method is rotation-invariant, and scale-invariant [11].
  - STAR feature detector is derived from the CenSurE (Center Surrounded Extrema) detector. STAR is like SIFT and SURF, but with some differences. STAR uses of bilevel center-surround filters to make the computations simpler [12].
  - FAST (Features from Accelerated Segment Test) algorithm's great advantage is that it is faster than many other well-known feature extraction methods [7].
  - ORB (Oriented FAST and rotated BRIEF) detector is a fast feature detector like the FAST detector with many modifications to enhance the performance [13]. The basic idea of this technique is that if a pixel is significantly different from the neighborhood pixels then it is marked to be a corner point [14].
  - The concept of MSER (Maximally Stable Extremal Regions) detector is based on analyzing the intensity of the original image at different scales. So, this method is rotation-invariant, scale-invariant, and lighting-invariant [15].
  - The Dense algorithm chooses the interest point locations on the nodes of a regular grid on the image [16]. The dense method gives better coverage of the image, a constant number of features per image area, and simple spatial relations between features. But the detected points may not be relevant [17].
  - SimpleBlob algorithm converts the original image into several binarized images by thresholding the source image with different thresholds. In each binary image, connected white pixels are grouped and the centers of these areas are the detected interest points [16].

B. Overview of description techniques

Once features have been detected, a local image patch around the feature will be extracted. This extraction needs to be encoded as a local histogram in a suitable descriptor. The result is known as a feature descriptor or feature vector [5].

- SIFT algorithm is based on taking neighborhood pixels around the keypoint. Then, this area is divided into sub-blocks and an orientation histogram for each sub-block is created. The grouping of these orientation histograms forms the total descriptor [7].
- SURF descriptor uses the same steps as the SIFT descriptor but with few modifications [10].
- The BRIEF (Binary Robust Independent Elementary Features) algorithm selects a set of location pairs in a unique way. Then, the pixel intensities are compared on each one of the selected location pairs to determine the binary values of this descriptor [18].
- ORB is basically like the BRIEF descriptor with some modifications to enhance the performance [13].

C. Overview of matching techniques

Once the features and their descriptors have been extracted from two or more images, the next step is to establish some feature matches between these images. Features matching is the task of establishing correspondences between two images of the same scene by comparing the extracted descriptors of the detected features in these images [4].

- The BruteForce matcher is simple. It uses the descriptor of each detected feature in the first set and matching it with all other features in the second set using Euclidean distance calculation and the closest one is will be considered its pair [7].
- The basic idea of the BruteForce-L1 matcher is the same as the Brute-Force matcher. Instead of calculating the Euclidean distance, this method is based on calculating distance in each dimension and then the summation of all these distances [7].
- BruteForce-Hamming is the same as the Brute-Force matcher but using Humming distance instead of using Euclidean distance. This method can be used by binary descriptors [16].
- Also, BruteForce-HammingLUT is used by binary descriptors and uses two bits rather than one bit in XOR operations compared to the previous algorithm. The LUT version is usually faster [16].
- FlannBased matcher is a fast library for approximate nearest neighbors in high dimensional spaces. FlannBased matcher is a big toolbox, it knows how to choose the right tools for fast nearest neighbor search [19].

III. IMAGE ANALYTICAL GEOMETRY

As cartesian coordinates which are used in Euclidean geometry, homogeneous coordinates, or projective coordinates are a system of coordinates that can be used in projective geometry. The number of required coordinates is one more than the dimension of the projective space being considered. The derived formulas based on homogeneous coordinates are often simpler than their cartesian coordinates. Homogeneous coordinates are used nowadays in several applications, including computer graphics and computer vision. They allow projective transformations to be easily represented by matrices [20].

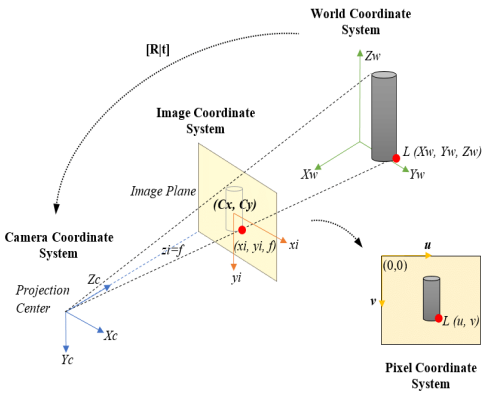


Fig. 1. Ideal projection of a three-dimensional object [18].

In the next part, the relationship between the real-world three-dimensional coordinates of features and the two-dimensional pixel coordinates of these features will be discussed.

A. World coordinates to camera coordinates

The world coordinate system should be aligned with the camera where the X_c and Y_c axes are aligned with the x_i and y_i axes of the image and the positive Z_c axis is aligned with the optical axis of the camera. This relationship includes a rotation matrix and translation between the camera coordinate system and the world coordinate system [22].

$$P_c = R(P_w - C) \tag{1}$$

$$\therefore \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \left(\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} - \begin{bmatrix} C_x \\ C_y \\ C_z \\ 1 \end{bmatrix} \right) \tag{2}$$

Where,

$P_w = [X_w \quad Y_w \quad Z_w \quad 1]$ are the homogenous world coordinates.

$P_c = [X_c \quad Y_c \quad Z_c \quad 1]$ are the homogenous camera coordinates.

$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ is a rotation matrix from the world coordinate system to the camera coordinate system.

$C = [C_x \quad C_y \quad C_z \quad 1]$ are the homogenous coordinates of the camera center in the world coordinate system.

B. Camera coordinates to image coordinates

The projection from the camera frame to the image frame is a perspective projection. The units of this image plane and the camera coordinate system are the. Furthermore, the depth of this plane in the camera coordinate system is the focal length [22].

$$x_i = f\frac{X_c}{Z_c} \quad , \quad y_i = f\frac{Y_c}{Z_c} \tag{3}$$

These equations can be written in a matrix form as follows.

$$\begin{bmatrix} x_i' \\ y_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{4}$$

$$x_i = \frac{x_i'}{z_i'} \quad , \quad y_i = \frac{y_i'}{z_i'} \tag{5}$$

Where, $[x_i \quad y_i]$ are the image coordinates and f is the focal length of the used camera.

C. Image coordinates to pixel coordinates

Finally, the image coordinates can be projected to the pixel coordinates using the following relationship. Scaling factors for pixels in each image direction are used [22].

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & O_x \\ 0 & 1 & O_x \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & S & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{6}$$

Where,

S is the skew coefficient of the sensor which is non-zero if the image axes are not perpendicular.

S_x , S_y are the number of pixels per world units in both x and y directions.

$[O_x , O_y]$ are the coordinates of the principal point in pixels.

D. Camera lens distortion

The camera matrix does not include lens distortions because an ideal pinhole camera does not have a lens. To precisely represent a real camera, the camera model should include both radial and tangential lens distortions [23].

Radial distortion occurs when light rays near the edges of the lens bend more than other light rays near the optical center. The value of the distortion depends on the size of the lens [24].

$$u_{distorted} = u(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) \tag{7}$$

$$v_{distorted} = v(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) \tag{8}$$

Where,

$[u_{distorted} \quad v_{distorted}]$ are the distorted pixel locations.

$[u \quad v]$ are the undistorted point locations. These locations are normalized coordinates by translating them to the optical center and dividing by the focal length.

$[k_1 \quad k_2 \quad k_3]$ are the radial distortion coefficients of the camera lens.

$r = \sqrt{u^2 + v^2}$ is the radial distance of the point.

Also, the tangential distortion occurs when the lens be not parallel to the image plane [24].

$$u_{distorted} = u + [2 * p_1 * u * v + p_2 * (r^2 + 2 * u^2)] \tag{9}$$

$$v_{distorted} = v + [2 * p_2 * u * v + p_1 * (r^2 + 2 * v^2)] \tag{10}$$

Where,

$[p_1 \quad p_2]$ are the tangential distortion coefficients of the camera lens.

E. Summary

Briefly, the projection of the world coordinates of a point to its pixel coordinates can be done using one matrix. This matrix includes several parameters: the camera position coordinates $[C_x \quad C_y \quad C_z \quad 1]$; the camera attitude angles which are represented by the rotation matrix R; the focal length of the used camera f; the skew coefficient S; the number of pixels per world units $[S_x, S_y]$; the coordinates of the principal point $[O_x, O_y]$; and the camera lens distortion coefficients $[k_1 \quad k_2 \quad k_3 \quad p_1 \quad p_2]$.

IV. EXPERIMENTS

In the following section, we will compare the performances of each algorithm that has been discussed before. This testing is performed on various groups of images. MATLAB-2017a with OpenCV 3.0 has been used for performing the experiments presented in this thesis. Specifications of the computer system used are Intel(R) Core (TM) i7-5500CPU @ 2.40 GHz, 3 MB Cache, and 8.00 GB RAM. All the used parameters are set as OpenCV's default.

A. Training Image

Training images dataset plays an essential role when testing algorithms of computer vision. However, using real images is usually not good enough due to insufficient variety. Also, the determination of the parameters of the real photos is difficult and time-consuming [25]. In this research, virtual images were constructed based on different camera parameters to be used later during the next experiments.

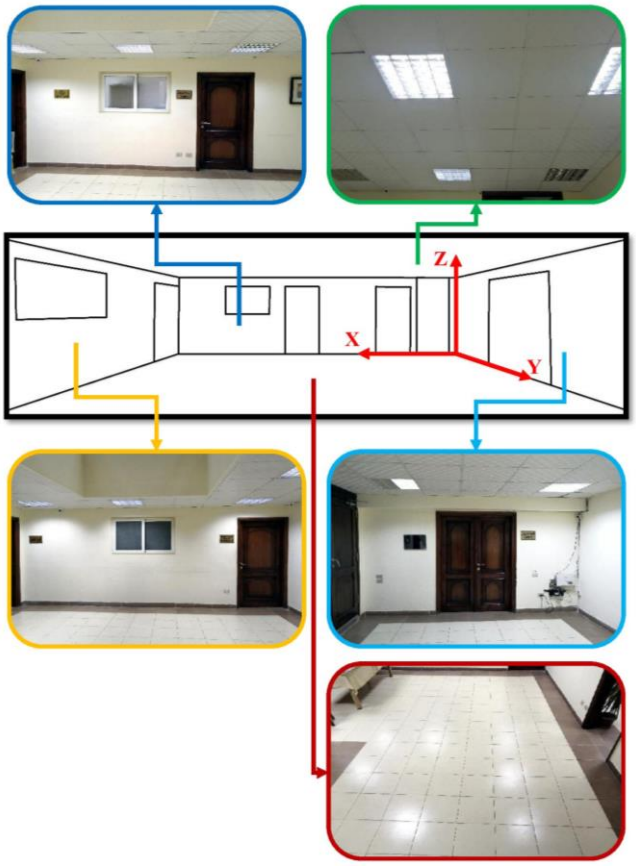


Fig. 2. Isometry of the used training area.

At first, the lobby of the surveying laboratory was chosen as a training area in these experiments. To simulate the layout of this area, terrestrial surveying was carried out based on an arbitrary coordinate system using the traditional surveying instruments (total station and steel tapes). Then, several photos were taken, using a high-resolution camera, for each side (each wall, floor, and ceiling) of this lobby. The three-dimensional isometry of the lobby and some of the taken photos are shown in the shown figure.

Based on the surveyed coordinates, each photo was manually georeferenced using ArcMap software. By assigning coordinates to some points (especially the corners of the surveyed features) in each photo, the image is warped automatically to fit these coordinates. So, real-world coordinates, related to the assumed arbitrary ground coordinates system, were assigned to each pixel in each photo.

After georeferencing the images, each photo was cropped and presented in its correct position. Besides, image enhancement was created to reduce the illumination variations between photos.

A uniformly spaced grid 1cm*1cm was constructed on each side of the lobby and the color intensity values (red band, green band, and blue band) at each point in this grid were determined, using ArcMap software.

As discussed before, any real-world coordinates can be projected into pixel coordinates based on theoretical equations. Furthermore, these equations depend on several parameters. Using these equations of imaging geometry, each point in the grid that formed before could be projected to its pixel coordinates then its color intensities, determined before, is recorded in this calculated pixel position. So, several training images could be constructed using different parameters.

generally, the training images construction depends not only on the parameters listed in the previous section but also, on the part of the world that be observed. The extent of the observable part of the world that be seen at a moment can be determined by the camera field of view.

The camera field of view is the solid angle through which a detector is sensitive to electromagnetic radiation [23].

$$DFOV = 2 * \tan^{-1} \frac{d/2}{f} \tag{11}$$

Where f is the camera focal length and d is the image diagonal length.

Also, there is a mathematical relation between the horizontal $HFOV$, vertical $VFOV$, and diagonal $DFOV$ fields of view [26].

$$DFOV = \sqrt{HFOV^2 + VFOV^2} \tag{12}$$

$$HFOV = 2 * \tan^{-1}\frac{\frac{w}{2}}{f} \quad , \quad VFOV = 2 * \tan^{-1}\frac{h/2}{f} \qquad (13)$$

Where, w is the width of the image and h is the height of the image.

By knowing the aspect ratio of the image, which set to 2:3 for all images, the horizontal and the vertical field of views could be determined from the diagonal field of view. Then, the image dimensions could be calculated.

When assuming the camera resolution, and by knowing the image dimensions, the number of pixels per world unit could $[S_x, S_y]$ be calculated.

Finally, for each constructed image, some parameters must be determined to build this image: the camera position; the camera attitude angles; the camera focal length; the skew coefficient; the field of view; the image resolution; and the lens distortion coefficients. Always, the coordinates of the principal point $[O_x, O_y]$ were taken half the image dimension. The limits of the parameters used to build images are shown in the next table.

Table- I: Used limits of each parameter used when building the training images.

Parameter		Minimum	Average	Maximum
Camera position coordinates	x	3.7 m	4.5 m	5.3 m
	y	3.2 m	4.0 m	4.8 m
	z	0.7 m	1.5 m	2.3 m
Camera attitude angles	ω	50 °	90 °	130 °
	$\emptyset$	-40 °	00 °	40 °
	κ	-40 °	00 °	40 °
Focal length	f	20 mm	60 mm	100 mm
Skew	S	-8 °	00 °	8 °
Field of view		60 °	100 °	140 °
Resolution		2 megapixels	6 megapixels	10 megapixels
Radial lens distortion	k_1	-0.5	0	0.5
	k_2	-0.5	0	0.5
	k_3	-0.5	0	0.5
Tangential lens distortion	p_1	-0.5	0	0.5
	p_2	-0.5	0	0.5

B. Comparison between all available feature detection algorithms

Using the average values in the previous table, an image was built to be the basic image for all the next experiments.

In this experiment, the interesting features were detected using all detectors available in the OpenCV library. This process was done serval times using the basic image but with different resolutions. The keypoints detected by each detector are shown in the next figures. Also, the number of keypoints detected by each detector are shown in the next table.

Fig. 3.The basic training image.

Table- II: Used limits of each parameter used when building the training images.

Detector	Image resolution in megapixels				
	2	4	6	8	10
FAST	1617	1920	2347	3081	3529
STAR	135	154	150	143	145
SIFT	623	772	875	895	883
SURF	2823	4502	5055	5702	6243
ORB	500	500	500	500	500
MSER	620	1121	1540	2007	2318
GFTT	1000	1000	1000	1000	1000
HARRIS	129	118	97	116	101
Dense	55777	111248	167000	222530	278426
SimpleBlob	9	7	8	9	6

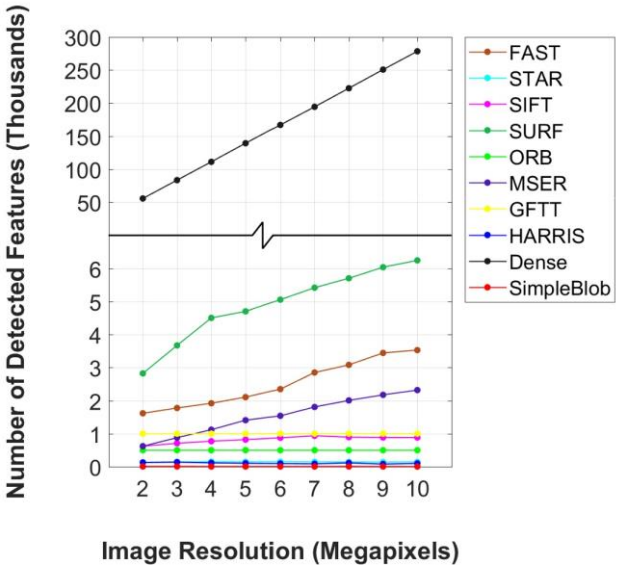


Fig. 4.The relation between the detected keypoints and the image resolutions for the available OpenCV detectors.

The results show that the OpenCV available detector techniques can be divided into two groups. The first group includes STAR, ORB, GFTT, HARRIS, and SimpleBlob and the second one includes FAST, SIFT, SURF, MSER, and Dense.

The first group methods detect constant or nearly constant numbers of keypoints even if the image resolution changes. While the numbers of keypoints detected by the second group methods increase with increasing the image resolution. Also, the numbers of keypoints detected by Dense method are huge comparing to other methods.

C. The methodology of choosing the best detector, descriptor, and matcher for different geometric image distortion

OpenCV library offers ten detectors, four descriptors, and five matchers. As discussed before, only three matchers (BruteForce, BruteForce-L1, FlannBased) could be used with all available descriptors, while the remaining two techniques (BruteForce-Hamming and BruteForce-HammingLUT) were designed to be used with binary descriptors (BRIEF, ORB) only. So, 160 methods ($10_{detectors} * [4_{descriptors} * 3_{matchers} + 2_{descriptors} * 2_{matchers}]$) can be used later in the remaining part of this test. Each used method includes three steps: detecting interest points in each image using a detector; describing these detected points using a descriptor; matching features in one image with other features in the second image using a matcher.

When comparing any two training images, the OpenCV algorithms were run and the matching list between these two images was produced. This matching list shows the pixel coordinates of features detected in the first image and their corresponding pixel coordinates in the second image. As stated before, the used training images were built using the world coordinates of the training area. So, the world coordinates of each pixel are known.

The pixel coordinates of features may change from image to another, but the world coordinates of these features must be the same. So, the evaluation of each method was based on converting all pixel coordinates in the matching list to their world coordinates and comparing the world coordinated of each pair of matching points in this matching list. The root mean squares (RMS) for each method was calculated using the next equation.

$$RMS = \frac{1}{n}\sum_{1=1}^{1=n}\sqrt{(X_i - X_i')^2 + (Y_i - Y_i')^2 + (Z_i - Z_i')^2} \quad (14)$$

Where $[X_i \quad Y_i \quad Z_i]$ are the world coordinates of a point in the first image and $[X_i' \quad Y_i' \quad Z_i']$ are the world coordinates of the matching point in the other image.

In addition, the resulting matching list produced from each method can be rearranged and sorted according to the matching distance between each pair of matching points. This matching distance between each pair of matching points could be taken as an indication of the accuracy of matching between this pair of points. In our analysis, the root mean squares (RMS) was calculated using the whole matching list and using the most accurate pairs only.

As stated before, the images are affected by several geometric distortions stated before. Under these geometric distortions, the matching methods were assessed in order to choose the best method to be used later.

Briefly, the basic image was matched to each image using the different methods discussed before. The results from all images were averaged and shown in the tables below. Furthermore, some graphs were drawn to show the relations between the time and RMS of each method. These relations were drawn as a polar coordinates system where the RMS of each method was the radial distance and the time of each method is the angle from the north in a clockwise direction.

D. Choosing the best detector, descriptor, and matcher when using a calibrated camera

When using a calibrated camera, all parameters may change except the camera lens distortion coefficients which are known before. In this part, the basic image was compared to each image in a group contains 100 images. The images used in this experiment were built as the basic image but all parameters, except the camera lens distortion coefficients, changed randomly from image to another.

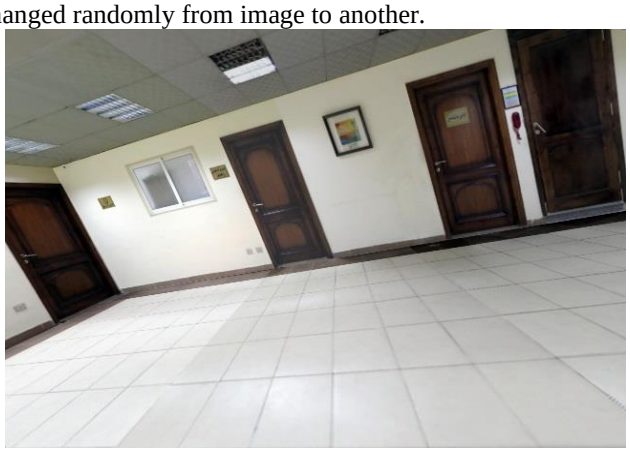


Fig. 5.Sample of the images used in this experiment (the camera was moved 0.5 meters in each direction and rotated 20 degrees around each axis, the camera focal length was changed to be 100 millimeters, the image skew coefficient was changed to be 10 degrees, the camera field of view was changed to be 120 degrees, and the image resolution was changed to be 10 megapixels.

Table- III: Time of each method in seconds (calibrated camera).

Detector	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob
	BruteForce										BruteForce-L1									
SIFT	2.7	3.1	12.1	23.0	3.5	145.9	3.6	3.3	1071.6	2.8	2.7	3.1	12.1	22.9	3.5	146.0	3.5	3.3	1309.5	89.4
SURF	0.8	1.5	8.0	7.6	1.0	19.9	2.1	2.1	659.9	1.5	0.8	1.5	8.0	7.6	1.1	19.8	2.2	2.1	590.6	1.6
BRIEF	0.7	1.4	7.8	4.8	0.9	12.0	2.1	2.1	412.8	1.6	0.6	1.4	7.8	4.8	0.8	11.9	2.1	2.1	340.8	1.5
ORB	0.6	1.4	7.7	4.7	0.6	11.9	2.1	2.1	1398.8	1.5	0.7	1.4	7.7	4.6	0.7	11.9	2.1	2.1	347.3	1.5
	BruteForce-Hamming										BruteForce-HammingLUT									
SIFT	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
SURF	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
BRIEF	0.7	1.4	7.8	4.9	0.9	11.9	2.1	2.0	529.8	1.5	0.7	1.4	7.8	4.9	0.8	11.9	2.1	2.1	531.6	1.5
ORB	0.7	1.4	7.7	4.8	0.7	11.9	2.1	2.1	541.7	1.5	0.7	1.4	7.7	4.8	0.7	11.9	2.1	2.1	544.8	1.5
	FlannBased																			
SIFT	2.7	3.2	12.1	22.6	3.5	146.1	3.4	3.4	25.9	3.7										
SURF	0.7	1.5	8.0	7.4	1.1	19.9	2.2	2.1	20.0	1.5										
BRIEF	0.6	1.5	7.8	4.7	0.9	11.9	2.1	2.1	15.7	1.6										
ORB	0.6	1.4	7.7	4.5	0.7	11.9	2.1	2.1	15.9	1.5										

Table- IV: RMS of each method in meters (calibrated camera).

Detector	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob
	BruteForce										BruteForce-L1									
SIFT	2.5	2.2	2.8	2.3	2.2	2.7	2.9	2.9	3.2	2.1	2.4	2.2	2.8	2.2	2.2	2.6	3.0	2.9	3.2	2.1
SURF	3.1	2.6	2.9	2.8	2.3	3.4	3.3	3.2	3.9	2.8	3.0	2.6	2.9	2.7	2.3	3.3	3.3	3.2	3.9	2.8
BRIEF	3.3	3.0	3.4	3.5	2.4	3.7	3.5	3.3	3.1	2.6	3.2	2.9	3.4	3.5	2.3	3.7	3.5	3.3	3.1	2.6
ORB	3.2	2.7	3.3	3.5	2.4	3.7	3.4	3.3	3.1	2.5	3.2	2.7	3.3	3.5	2.3	3.7	3.4	3.3	3.1	2.4
	BruteForce-Hamming										BruteForce-HammingLUT									
SIFT	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
SURF	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
BRIEF	3.2	2.8	3.4	3.5	2.2	3.7	3.5	3.3	3.1	2.4	3.2	2.8	3.4	3.5	2.2	3.7	3.5	3.3	3.1	2.4
ORB	3.2	2.6	3.3	3.5	2.2	3.7	3.4	3.2	3.0	2.2	3.2	2.6	3.3	3.5	2.2	3.7	3.4	3.2	3.0	2.2
	FlannBased																			
SIFT	2.5	2.2	2.8	2.4	2.2	2.7	2.9	2.9	3.2	2.1										
SURF	3.1	2.6	2.9	2.8	2.3	3.4	3.3	3.2	3.4	2.8										
BRIEF	3.3	3.0	3.4	3.5	2.5	3.7	3.5	3.3	3.1	2.6										
ORB	3.2	2.8	3.3	3.5	2.4	3.7	3.4	3.3	3.1	2.5										

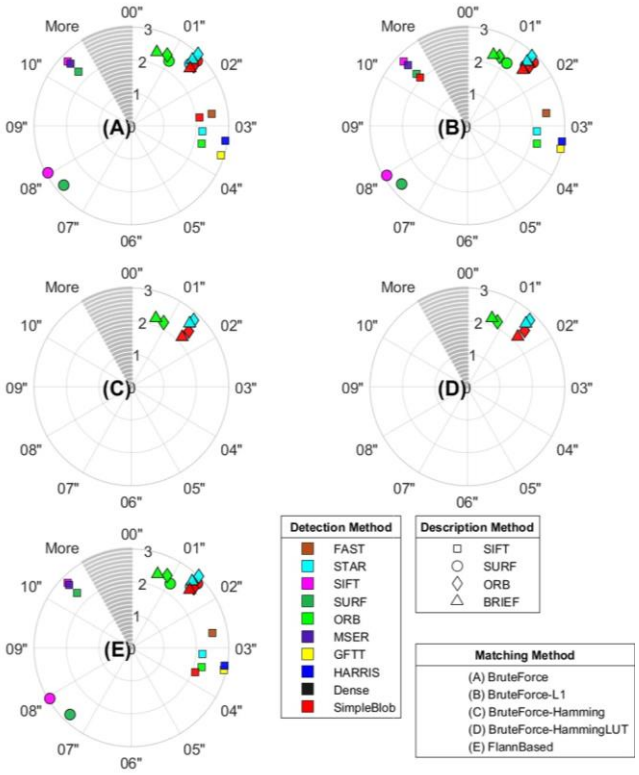
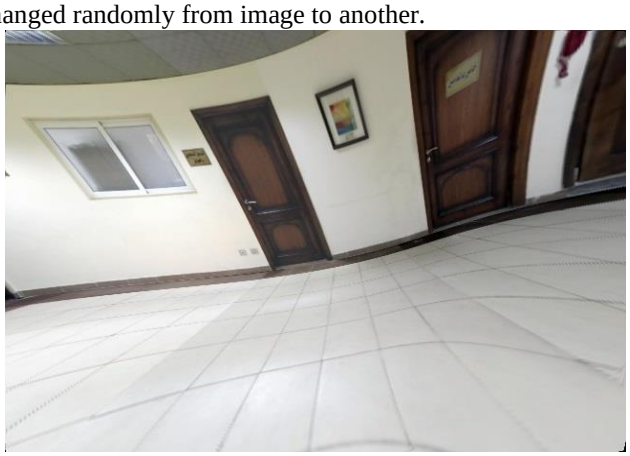


Fig. 6. The relation between RMS and time of each method (calibrated camera).

E. Choosing the best detector, descriptor, and matcher when using any camera

When using any camera, all parameters may change. In this part, the basic image was compared to each image in a group contains 100 images. The images used in this experiment were built as the basic image but all parameters changed randomly from image to another.



Sample of the images used in this experiment (the camera was moved 0.5 meters in each direction and rotated 20 degrees around each axis, the camera focal length was changed to be 100 millimeters, the image skew coefficient was changed to be 10 degrees, the camera field of view was changed to be 120 degrees, the image resolution was changed to be 10 megapixels, and the camera lens distortion coefficients were was changed to be 0.1).

Table- V: Time of each method in seconds (Any camera).

Detector	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob
	BruteForce										BruteForce-L1									
SIFT	2.6	3.0	11.7	20.9	3.3	225.8	3.3	3.1	846.6	2.6	2.6	3.0	11.7	20.8	3.3	148.0	3.2	3.1	806.3	2.6
SURF	0.7	1.4	7.8	7.0	1.0	43.6	2.0	1.9	518.4	1.4	0.7	1.3	7.8	6.9	1.0	20.1	2.0	1.9	464.7	1.4
BRIEF	0.6	1.3	7.6	4.5	0.8	12.1	1.9	1.9	316.6	1.4	0.6	1.3	7.6	4.4	0.8	12.1	2.0	1.9	268.7	1.4
ORB	0.6	1.3	7.5	4.4	0.6	12.1	2.0	1.9	321.9	1.4	0.6	1.3	7.5	4.3	0.6	12.1	2.0	1.9	273.6	1.4
	BruteForce-Hamming										BruteForce-HammingLUT									
SIFT	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
SURF	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
BRIEF	0.6	1.3	7.6	4.6	0.8	12.1	2.0	1.9	429.9	1.4	0.6	1.3	7.6	4.6	0.8	12.1	2.0	1.9	430.0	1.4
ORB	0.6	1.3	7.5	4.4	0.6	12.1	2.0	1.9	439.2	1.4	0.6	1.3	7.5	4.4	0.6	12.1	2.0	1.9	445.3	1.4
	FlannBased																			
SIFT	2.5	3.0	11.7	20.6	3.3	146.5	3.2	3.1	23.3	2.6										
SURF	0.7	1.4	7.8	6.8	1.0	20.2	2.0	1.9	18.8	1.4										
BRIEF	0.6	1.3	7.6	4.4	0.8	12.1	2.0	1.9	15.2	1.4										
ORB	0.6	1.3	7.5	4.2	0.6	12.3	18.1	1.9	16.9	1.4										

Table- VI: RMS of each method in meters (Any camera).

Detector	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob
	BruteForce										BruteForce-L1									
SIFT	2.7	2.3	2.8	2.5	2.4	2.7	3.0	2.9	3.1	2.3	2.7	2.3	2.8	2.4	2.4	2.7	3.0	3.0	3.1	2.3
SURF	3.1	2.7	3.0	2.8	2.5	3.3	3.3	3.0	3.3	2.9	3.0	2.7	3.0	2.7	2.4	3.2	3.3	3.0	3.3	3.0
BRIEF	3.2	3.0	3.3	3.4	2.6	3.6	3.4	3.2	3.1	2.7	3.2	2.9	3.3	3.4	2.5	3.6	3.4	3.2	3.0	2.7
ORB	3.2	2.8	3.2	3.4	2.5	3.6	3.4	3.2	3.1	2.5	3.2	2.7	3.2	3.4	2.4	3.6	3.4	3.2	3.1	2.4
	BruteForce-Hamming										BruteForce-HammingLUT									
SIFT	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
SURF	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
BRIEF	3.2	2.8	3.3	3.4	2.3	3.6	3.4	3.2	3.0	2.6	3.2	2.8	3.3	3.4	2.3	3.6	3.4	3.2	3.0	2.6
ORB	3.2	2.7	3.2	3.4	2.3	3.6	3.4	3.2	3.0	2.4	3.2	2.7	3.2	3.4	2.3	3.6	3.4	3.2	3.0	2.4
	FlannBased																			
SIFT	2.7	2.4	2.8	2.5	2.4	2.8	3.0	2.9	3.1	2.3										
SURF	3.1	2.7	3.0	2.8	2.5	3.3	3.3	3.0	3.4	2.9										
BRIEF	3.2	3.0	3.3	3.4	2.6	3.6	3.4	3.2	3.1	2.7										
ORB	3.2	2.8	3.3	3.4	2.5	3.6	3.4	3.2	3.1	2.5										

Table- VII: RMS of each method in meters using the most accurate 50% of the matching pairs of points (Any camera).

Detector	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob
	BruteForce										BruteForce-L1									
SIFT	2.5	1.4	2.5	2.0	1.4	2.5	3.0	3.1	3.1	1.5	2.5	1.4	2.6	1.9	1.2	2.4	3.1	3.2	3.1	1.4
SURF	3.0	2.3	2.8	2.7	1.7	3.1	3.2	3.1	2.7	2.7	2.9	2.2	2.8	2.6	1.6	3.1	3.2	3.1	2.7	2.7
BRIEF	3.2	2.7	3.4	3.4	2.0	3.6	3.4	3.2	3.1	2.5	3.2	2.5	3.4	3.4	1.9	3.6	3.4	3.3	3.1	2.5
ORB	3.2	2.4	3.3	3.4	2.1	3.6	3.4	3.2	3.1	2.1	3.2	2.3	3.3	3.4	2.0	3.6	3.4	3.2	3.1	1.9
	BruteForce-Hamming										BruteForce-HammingLUT									
SIFT	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
SURF	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
BRIEF	3.2	2.4	3.4	3.4	1.6	3.6	3.4	3.2	3.0	2.3	3.2	2.4	3.4	3.4	1.6	3.6	3.4	3.2	3.0	2.3
ORB	3.2	2.1	3.3	3.4	1.9	3.6	3.4	3.2	3.0	1.9	3.2	2.1	3.3	3.4	1.9	3.6	3.4	3.2	3.0	1.9
	FlannBased																			
SIFT	2.5	1.4	2.6	2.0	1.4	2.5	3.0	3.0	3.1	1.5										
SURF	3.0	2.3	2.8	2.7	1.7	3.1	3.2	3.1	3.8	2.7										
BRIEF	3.2	2.7	3.4	3.4	2.0	3.6	3.4	3.2	3.1	2.5										
ORB	3.2	2.4	3.3	3.4	2.1	3.6	3.4	3.2	3.1	2.1										

Table- VIII: RMS of each method in meters using the most accurate 10% of the matching pairs of points (Any camera).

Detector	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob	FAST	STAR	SIFT	SURF	ORB	MSER	GFTT	HARRIS	Dense	SimpleBlob
	BruteForce										BruteForce-L1									
SIFT	3.2	0.7	2.6	1.2	0.6	1.7	3.2	2.9	2.8	0.4	3.2	1.0	2.7	1.1	0.6	1.6	3.3	2.9	2.8	0.0
SURF	2.9	1.5	2.0	2.3	1.2	2.8	3.1	2.7	3.2	1.4	2.9	1.6	1.9	2.2	1.2	2.8	3.0	2.7	3.2	1.5
BRIEF	3.2	2.3	3.4	3.4	1.2	3.6	3.4	3.3	3.1	2.0	3.3	2.1	3.4	3.4	0.9	3.6	3.4	3.3	3.1	2.0
ORB	3.2	1.7	3.4	3.4	1.8	3.5	3.4	3.2	3.1	1.6	3.3	1.6	3.3	3.4	1.7	3.5	3.4	3.2	3.1	1.5
	BruteForce-Hamming										BruteForce-HammingLUT									
SIFT	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
SURF	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
BRIEF	3.3	1.9	3.4	3.4	0.8	3.5	3.4	3.3	3.0	1.8	3.3	1.9	3.4	3.4	0.8	3.5	3.4	3.3	3.0	1.8
ORB	3.3	1.4	3.3	3.4	1.5	3.5	3.5	3.2	3.1	1.4	3.3	1.4	3.3	3.4	1.5	3.5	3.5	3.2	3.1	1.4
	FlannBased																			
SIFT	3.2	0.7	2.6	1.2	0.6	1.7	3.2	2.9	2.9	0.4										
SURF	2.9	1.5	2.0	2.3	1.2	2.8	3.1	2.7	4.8	1.4										
BRIEF	3.2	2.3	3.4	3.4	1.2	3.6	3.4	3.3	3.1	2.0										
ORB	3.2	1.7	3.3	3.4	1.8	3.5	3.4	3.2	3.2	1.6										

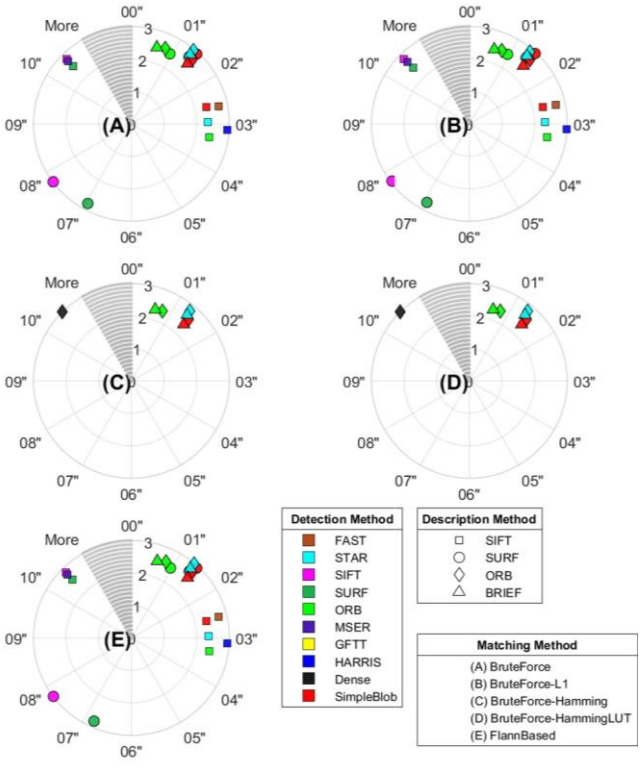


Fig. 7. The relation between RMS and time of each method (any camera).

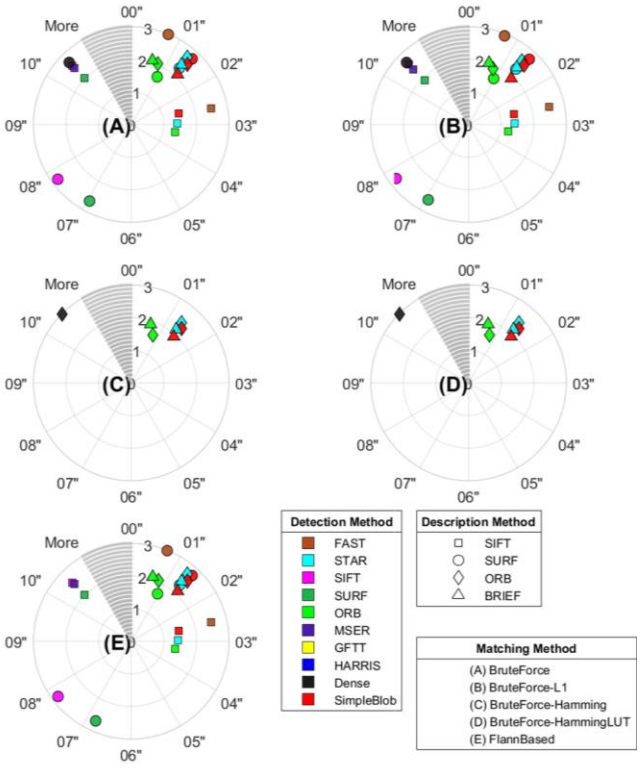


Fig. 8. The relation between RMS and time of each method using the most accurate 50% of the matching pairs of points (any camera).

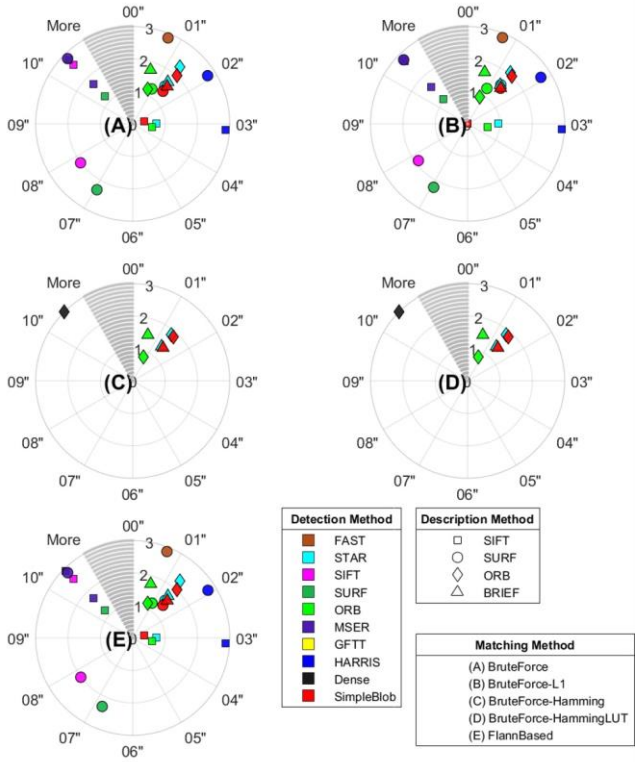


Fig. 9.The relation between RMS and time of each method using the most accurate 10% of the matching pairs of points (any camera).

V. ANALYSIS AND CONCLUSIONS

Based on all the above experiments, there are several important notes. Because this research concern with real-time navigation, not only the accuracy of each method was assessed but also the time of each method must be taken into considerations.

At first, the SimpleBlob detector should not be used in indoor environments due to its little detected points that cannot support the next navigation stages. Also, the Dense detector should be not used due to its large number of detected points that increase the processing time more than usual without any significant accuracy improvement comparing to the rest detectors.

Based on the processing time, the rest detectors can be divided into four groups. The fastest group contains the FAST, the STAR, and the ORB detectors. The GFTT and the HARRIS detectors are slower than the first group but by a small second. The third group contains the SIFT and the SURF detectors, while the MSER is the slowest detector. Also, the resulting accuracies from all detectors were nearly the same except the ORB and the STAR which are more accurate detectors.

The determined accuracies from all descriptors were nearly the same except the accuracy when using the SIFT descriptor which is nearly more accurate. Besides, the ORB and the BRIEF descriptors are faster than others.

The resulting accuracies from all matchers were nearly the same. While the FlannBased matcher was faster than other matchers.

In conclusion, when the accuracy is the most important target, the ORB detector, the SIFT descriptor, and the FlannBased matcher are preferred to be used. While for a faster solution with less amount of accuracy, the SIFT descriptor should be replaced by the ORB descriptor.

Because the ORB detector gives the same number of interesting points for all image sizes, it is recommended to reduce the image size before any calculations to reduce the processing time.

The camera calibration process could increase the accuracy by a very small amount. So, this step could be exceeded. In addition, only the most accurate 10% of the matching pairs of points can be used in the next stages to increase the resulting accuracy.

REFERENCES

1. M. Attia, "Map Aided Indoor and Outdoor Navigation Applications," 2013.
2. Infsoft, 2020. [Online]. Available: www.indoornavigation.com.
3. N. Jayanthi and S. Indu, "Comparison of Image Matching Techniques," 2018.
4. M. Hassaballah, A. A. Abdelmgeid and H. ,. Alshazly, "Image Features Detection, Description and Matching," 2016.
5. B. R. Naidu, M. S. P. Babu, P. L. Rao and K. V. L. Bhavani, "Efficient Case Study for Image Edge Gradient Based Detectors-Sobel, Robert Cross, Prewitt, and Canny," 2012.
6. T. Lindeberg, "Scale Selection Properties of Generalized Scale-Space Interest Point Detectors," 2013.
7. Doxygen, 2020. [Online]. Available: https://docs.opencv.org/master/d9/df8/tutorial_root.html.
8. J. Shi and C. Tomasi, "Good Features to Track," 1994.
9. D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, 2004.
10. H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, Speeded-Up Robust Features (SURF), 2008.
11. S. Birchfield, "SURF Detectors and Descriptors," 2011.
12. S. Krig, "Interest Point Detector and Feature Descriptor Survey," 2014.
13. E. Rublee, V. Rabaud, K. Konolige and G. Bradsk, "ORB: an Efficient Alternative to SIFT or SURF," 2011.
14. C. Luo, "Overview of Image Matching Based on ORB Algorithm," 2019.
15. J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," 2002.
16. G. M. Moura and R. L. D. Da Silva, "Analysis and Evaluation of Feature Detection And tracking Techniques Using OpenCV With Focus on Markerless Augmented Reality Applications," 2017.
17. T. Tuytelaars, "Dense Interest Points," 2010.
18. M. Calonder, V. Lepetit, C. Strecha and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," 2010.
19. J. Minichino and J. Howse, Learning OpenCV 3 Computer Vision with Python, 2020.
20. M. Böcher, "Introduction to Higher Algebra," 1907.
21. L. Fernandez, V. Avila and L. Goncalves, "A Generic Approach for Error Estimation of Depth Data from (Stereo and RGB-D) 3D Sensors," 2017.
22. E. Cristofalo, " Final Project - 3D Scene Reconstruction From Video," 2014.
23. Z. Zhang, "A Flexible New Technique for Camera Calibration," 1998.
24. MathWorks, "What Is Camera Calibration?," 2020.
25. Y. Tian, X. Li and K. Wang, "Training and Testing Object Detectors with Virtual Images," 2017.
26. T. Ngo, A. Abdukhakimov and D. Kim, "Long-range Wireless Tethering Selfie Camera System using Wireless Sensor Networks," 2019.