University of Nottingham
UK | CHINA | MALAYSIA

Department of Electrical and Electronic Engineering

H54IOP / H53PJE / H53PJ3

(delete whichever of these is inappropriate)

Final Year Project Thesis

Image Processing of Motorway CCTV Cameras

AUTHOR: **Ms Shuqi WU**

ID NUMBER: **16521800**

SUPERVISOR: **Dr. Ed Morris**

MODERATOR: **Dr. James Bonnyman**

DATE: **14 May 2020**

Third *(Fourth if applicable)* year project report is submitted in part fulfilment of the requirements of the degree of Bachelor *(Master if applicable)* of Engineering.

Table of Contents

Abstract

There are more and more data from multiple sensors used to observe the earth process, so it is becoming significant to develop an image processing technique which is appropriate to a problem solution. This thesis presents examples of the basic background knowledge and specific implementation of cross-correlation algorithm and ORB feature-based algorithm to process images. The two algorithms are used to process motorway CCTV images for further study such as observing the changes of greenness throughout the year. Details of two algorithms are given in the thesis, and the advantages and disadvantage of each method are discussed later.

Chapter 1: Introduction

1.1 Background

In reality, digital images can be captured by digital cameras, sensors, etc. Image processing is a pre-stage where extracts information and improves visualization for further study by processing data and converting original image to expected result [4][5]. Depending on specific objectives and purposes, the appropriate processing methodology and techniques will be developed and applied to get the best result in different applications [5].

Imaging sensors such as satellite remote sensing system, which gives a uniform and repeated view of the earth, and the applications are widely used in such as crop monitoring, deforestation, urban growth, etc. [5]. In this project, image processing techniques are used for phenology analysis, which is an environmental science to monitor life cycle events of species with global climate change, and plant phenological event is important aspect for analysing ecosystems process such as patterns [34].

The satellite remote sensing is an important and popular way to monitor the changes of earth system [33], but high spatial resolution images taken by satellites are expensive and sensitive to effects of clouds and atmospheric conditions. Considering the cost and resolution conditions of images, images taken by CCTV camera with acceptable revolution is chosen as analysis object for this project [1]. CCTV cameras are used for monitoring traffic at the same time the images can be used as a cost-effective way for vegetation observation analysis.

The 3D geometric features in the world are captured into 2D features shown as discrete color or intensity values in images, and the values are influenced by ambient illumination and sensors properties [4]. In addition to the two aspects, because images are taken at different time, the different viewpoint of camera and varied weather conditions are also important factors when implement image processing algorithms in this project [1].

There are various techniques such as image pre-processing, image enhancement and image classification, etc. used in image processing in order to keep versatility and maintain precision of data in original images [5][6]. For example, image enhancement used in image processing is to filter kinds of noise caused by illumination conditions or imaging systems so that the important image characteristics can be emphasized without change intrinsic information of images [6]. Some essentially mathematical operations related to algorithms of image processing are such as gradient calculation, edge detection, filtering and image transformation [7]. More details about techniques and operations used in this project are claimed in the part of Literature Review.

The fundamentals of image processing methods are based on numerical analysis like the least squares, matrix theory and functional transforms [9]. As the development of computer technology, a variety of applications are developed in a wide range of fields in science in the last five years. Medical field (medical diagnosis), video processing, computer vision by using image processing methods match images and find correspondence between them [10] and object recognition [21], etc.

1.2 Aim of the project

Because of limitations of other observations methods such as satellite remote sensing, the motorway CCTV image is proposed to be an inexpensive and acceptable dataset for phenology or related areas analysis. The aim of image processing is to extract useful information by improving the quality of image dataset, for specifically, there are examples shown in Fig.1.1.

There are lots of blurred images in dataset due to rain or fog shown in Fig.1.1 as Case 2, which is clearly not suitable for analysis and should be filtered. The other example is shown as Case 4 image, which is taken by totally different viewpoint even different camera, if people want to extract more information based on scene of Case 1 image, then the Case 4 image is insignificant can should be discarded. Another situation is that based on information of Case 1 image, it is clearly Case 3 image, taken from different viewpoint because of strong winds, will be considered as analysis object for extra information of the overlapping part. In addition to the influence of rains, winds and maybe image storage errors, other factors also can cause loss of data such as camera malfunction, disconnection of camera, etc.

			
Case 1	Case 2	Case 3	Case 4

Figure 1.1: Four images in database

Considering improvement of utilization ratio of image dataset and reduction of manpower expense, the aim of this project is finding optimized algorithms to filter unnecessary images and align wanted images automatically.

In order to achieve this goal, the most important stage in this project is implementing algorithms shown in Fig.1.2, and this stage can be divided into two small tasks, which are image categorization and image alignment, so the first task filters bad images such as blurred images because of rains and images taken by totally different viewpoints, after filtering, the remaining images are aligned based on algorithms.

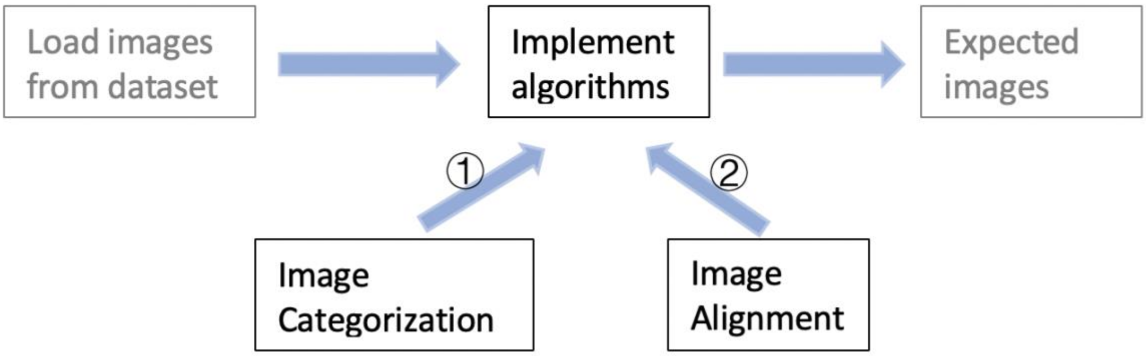


Figure 1.2: basic steps of the project

1.3 Objective of the project

There are three main parts for the objective of the project. First part is learning basic theory of tools and methods used in this project. Secondly, depending on background research, two methods are implemented to process images. The final stage of the project is evaluating the performance of methods respectively according to matching accuracy rate.

Research:

- Learn to use Python and MATLAB to implement image processing algorithm.
- Learn image processing techniques.
- Learn open source library OpenCV.
- Learn theory of cross-correlation algorithm including basic steps such as noise filtering and similarity calculation methods.
- Learn theory of feature-based methods including basic steps such as feature detection and feature matching methods.

Development:

- Write a test program based on two images by using cross-correlation algorithm.
- Process a thousand images and adjust threshold value depending on the performance of results.
- Write a test program based on two images by using ORB feature-based algorithm.
- Process nearly a thousand images and adjust three parameters, which are the number of feature points, threshold values of transformation matrix and distance of matching points.

Evaluation:

- Evaluate pros and cons of each method used in this project.

Chapter 2: Literature overview

2.1 Image registration

Image registration is an alignment making two images overlapped by adjusting the pixels of test image to the coordinate system of reference image [3][5]. Techniques, which can be concluded into four basic steps including feature identification, feature matching, spatial transformation and interpolation, are used for automatic image registration [5].

2.2 Cross-correlation algorithm

The matching comparison progress of 2D edge feature detection algorithms can use (Normalized) Cross-Correlation algorithm with basic image processing techniques to find disparity showing the distance of different position results of images by overlapping two images (objects) [26].

The basic mathematical calculation is shown as below:
There are two matrices, $n \times n$ M1 and $m \times m$ M2, so the size of cross-correlation matrix is $(m+n-1) \times (m+n-1)$. For example, $n = 5$ and $m = 3$, then the result matrix is 7×7:

$$C = \begin{pmatrix} c_{-2,-2} & c_{-2,-1} & c_{-2,0} & c_{-2,1} & c_{-2,2} & c_{-2,3} & c_{-2,4} \\ c_{-1,-2} & c_{-1,-1} & c_{-1,0} & c_{-1,1} & c_{-1,2} & c_{-1,3} & c_{-1,4} \\ c_{0,-2} & c_{0,-1} & c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} & c_{0,4} \\ c_{1,-2} & c_{1,-1} & c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,-2} & c_{2,-1} & c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,-2} & c_{3,-1} & c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \\ c_{4,-2} & c_{4,-1} & c_{4,0} & c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{pmatrix}.$$

Figure 2.1: the example of matrix C [35]

For example, when calculate value of $c_{0,2}$, the procedure shown in Fig.2.2,

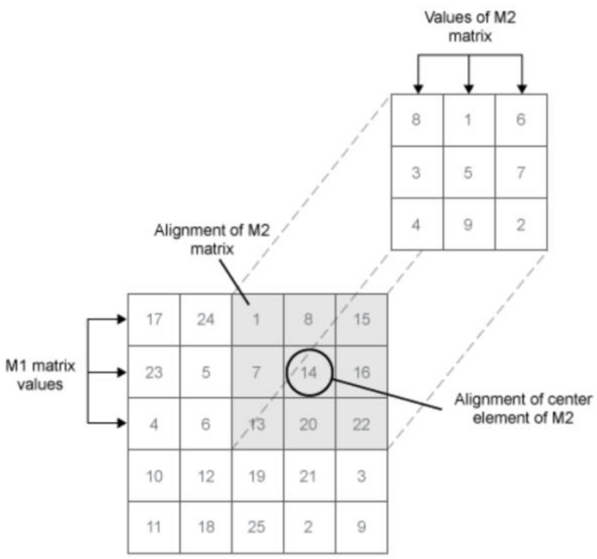


Figure 2.2: the process of calculation [35]

Then M2 is corresponding to the (1:3,3;5) part of M1 shown in Fig.2.2.
The calculation is: $1 \times 8 + 7 \times 3 + 13 \times 4 + 8 \times 1 + 14 \times 5 + 20 \times 9 + 15 \times 6 + 16 \times 7 + 22 \times 2 = 585$

2.2.1 Edge detection:

The main information of an image can be obtained from its edge, where the grayscale change is sharp, i.e. the brightness of the image changes dramatically [29][30].

In the classical edge detection method, the edge detection operator is constructed based on its small neighbourhood of pixels in original image [31].

Canny operator edge detection algorithm:

Canny edge detection proposed by John Canny in 1986 is one of most popular edge detection method, and the steps of this method is discussed as below [29]:

1. Gaussian filtering:

This step is to smooth the image and filter noise from the image by using 2D Gauss function and related mathematical calculation, and in this step, the selection of sigma parameter is related to accuracy of image edge.

2. Calculate intensity gradient:

Canny algorithm calculates vertical, horizontal and diagonal directions of gradient, so the peak of the derivative of a curve fitting the gradient between light and dark areas in an image is assumed as the location of an edge pixel, and the amplitude and direction of the gradient of the pixel are calculated.

3. Non-maximum suppression:

Depending on the amplitude and direction of the gradient of the potential edge pixel, if the value is greater than the adjacent points, which are in the direction of the gradient of the potential edge pixel, then the potential edge pixel will be considered as edge point. Otherwise, the potential edge pixel will be set to 0 and discarded.

4. Double threshold to filter edge:

After non-maximum suppression stage, there are still some detected edge pixels from noise, so weak gradient edge pixels should be filtered. The double threshold method is to set upper and lower threshold values to filter edge pixels, and there are three different situations to determine if a pixel is an edge point or not. Case one is that a pixel can be verified as edge point if its gradient is larger than upper threshold value; case two is if the gradient is smaller than lower threshold value, the pixel can be verified as not an edge point and will be discarded; case three is the pixel gradient is within the range of upper and lower threshold values, then the pixel is considered as week edge pixel.

5. Hysteresis to track edge:

This is final stage to determine if the weak edge pixel is real edge pixel or not, so the eight adjacent points of the weak edge pixel are examined to see if there is a pixel whose gradient is larger than upper threshold value, if it is, then the weak edge pixel will be verified as real edge pixel.

2.3 Feature-based Algorithms

There are basic three stages of feature-based algorithm. The first stage is detecting keypoints of images, then find description to all the keypoints for distinguishing them. The next stage is using image matching algorithms depending on different image processing purposes to compare features of reference image to a test image. Finally, kinds of image transformation methods based on calculated transformation matrix are applied to get an expected result.

One of the most important factors that affects the choice of image matching method is the variations in images, such as different scale, illumination and object to be matched [10]. Depending on different conditions of image problems, a variety of methods are tested to evaluate the performance of each image matching algorithms.

2.3.1 Scale Invariant Feature Transform (SIFT)

SIFT is a remarkably successful algorithm, which can abstract diverse unchangeable features from images, performing reliably on scenarios such as variant image scale and rotation, illumination and camera viewpoint change and affine distortion [11].

There four main stages of SIFT algorithm given by Lowe in 2004 to produce image features:

1. Scale-space extrema detection:

A difference of Gaussian function is implemented to acquire possible feature points which are not variant with image scale and orientation by searching all image locations.

2. Keypoint localization:

Depending on the stability of possible feature points, location and scale of keypoints are determined by a detailed model.

3. Orientation assignment:

The orientations of each keypoint, which is one of important factors distributing transformation operations, can be found based on gradient directions of image.

4. Keypoint descriptor:

Keypoint descriptor, which is invariant to shape distortion and illumination change, is a representation of local image gradients in the region around each keypoint. A most important aspect of keypoint descriptor is that they are greatly different to each other, so there is a high accuracy of a set of matches from database of features. In a cluttered image, the accuracy of matching points may decrease because some features from background cannot find any correct match in the database.

The procedure of using SIFT algorithm for image matching is that, firstly, features of a reference image are extracted and stored in a database, then features of another image are compared individually to the features in the previous database, finally, depending on the matching algorithms, the potential matching features can be found [11].

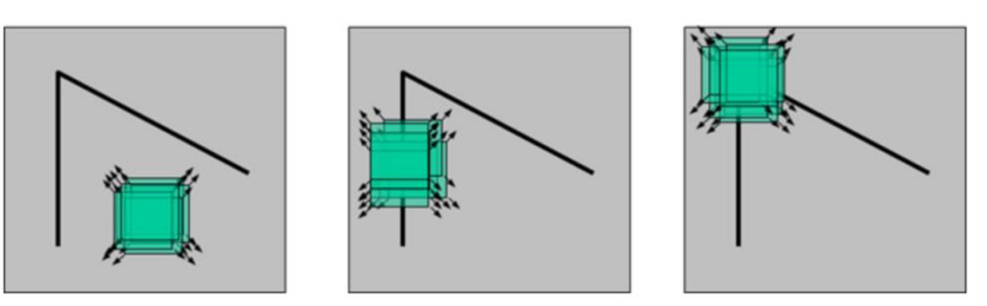
There are lots of applications including object recognition, image stitching [14], visual mapping [15], etc. prove the success of SIFT algorithm. Using highly distinctive feature descriptors is an advantage of SIFT algorithm to get good accuracy of matching features, but it is also the reason why SIFT algorithm has high computation cost [11].

2.3.2 Harris

Harris corner detector:

There are three cases of detecting corners are shown in Fig.2.3. The window shifts a small amount in various directions can determine average changes of image intensity.

In Fig.2.3, the one called ‘flat’ is because the window shifts a small amount in various directions and the intensity is approximately constant or only a small change; the second one is that if the window on an edge, then the window shifts along the edge will get a small change, but when the shift perpendicular to the edge will result in a large change; the last one is that if the window on a corner, then all the shifts will cause a large change in intensity, so the corner can thus be detected [28].



“Flat” “Edge” “Corner”

Figure 2.3: Three different cases of detecting corners [28]

The useful feature information in an image is the corner points, which can determine important information in the image, one of the simplest corner detectors is Harris corner detector, which is mathematical way to find interest points, and the region around the points shows edges in more than one direction [28] [32].

The basic idea of the algorithm is using difference in intensity to calculate a displacement of (u, v) in all directions, the calculation is shown as below:

$$E(u, v) = \sum_{x,y} \mathrm{w}\,(x, y)\,[I(x + u, y + v) - I(x, y)]^2 \qquad Eqn. 1$$

Where, w (x, y) is window function, which is a rectangular window or a gaussian window giving weights to pixel (x, y). $I(x+u, y+v)$ is shifted intensity and $I(x, y)$ is intensity. I_x and I_y represent image derivatives in x and y directions respectively.

After mathematical calculations based on previous equation, the corner response value R is used to determine corners. The figure below shows the several situations of different R values:

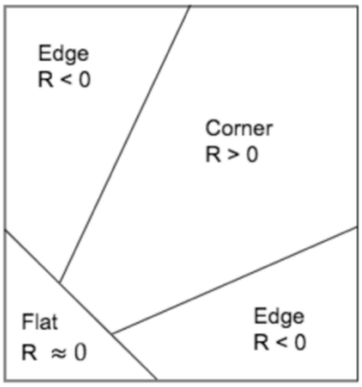


Figure 2.4: Harris Response R [28]

<u>Harris corner detector algorithm:</u>

The procedures of Harris Corner Detection algorithm are:
1. Convert color image to grayscale;
2. Compute derivatives
3. Define matrix of each pixel point
4. Calculate Harris response R
5. Using R value to find edges and corners

2.3.3 Features from Accelerated Segment Test (FAST)

<u>FAST feature detector:</u>

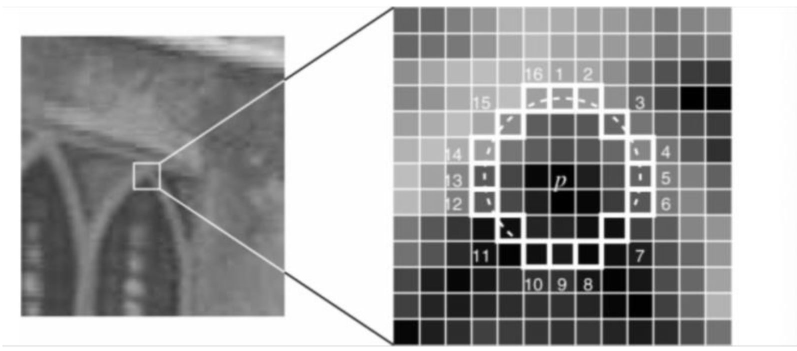


Figure 2.5 An example of feature point. [16]

The figure above shows p around with a circle of 16 pixels. If the corner detector detects all pixels in the circle are brighter than $(I_p + t)$ or darker than $(I_p - t)$, where I_p is the intensity of point p and t is a threshold value, then p will be classified as a corner [19][20]. It is inefficient to evaluate all 16 pixels on the circle, so there is a high-speed test only checking pixel 1, 9, 5 and 13 to filter a large number of pixels which are not actual corner points. The high-speed test checks point 1 and 9, if both be brighter or darker than t, then keep checking pixel 5 and pixel 13, if the pixel p is a corner, then among point 1, 9, 5 and 13 at least three of them are brighter than$(I_p + t)$ or darker than $(I_p - t)$, an opposite situation is that neither of the condition is satisfied, then pixel p will be considered as not a corner point [16]. The same condition test is applied to all the pixels, and the pixels satisfying the condition test become candidate pixels, finally, all the pixels in the 16-pixel-circle of these candidate pixels $(p_1, p_2, \ldots)$ are examined.

Although the detector has high performance, it still has some vulnerabilities [16]:

1. if N is smaller than 12, then the high-speed test is not available anymore, so if among four pixels (the same relationship as pixel 1 and 9, pixel 5 and 13), two of them are both significantly brighter or darker than pixel p then the pixel p can be considered as corner point.

2. many features are detected adjacent to one another, it is not beneficial to match whole part of images.

The result of an example is shown as below:

Figure 2.6: An example shows detected feature points [16]

FAST algorithm is faster than the other known corner detector, but the disadvantages of it is that a lot of noise can influence the robust of the algorithm and the result of the algorithm really depends on the threshold values t, when there is no orientation information, the algorithm is no longer invariant to rotation [16].

2.3.4 Binary Robust Independent Elementary Features (BRIEF)

BRIEF feature point descriptor:

SIFT algorithm is an efficient way to detect and match features with 128-vector highly discriminant feature descriptors, but the computational cost is really high [21]. A solution to speed up matching process and decrease memory consumption is the use of a short descriptor, few bits without loss of performance can be used to replace floating-point SIFT descriptor [22][23]. A dimensionality reduction method can be applied for reducing SIFT descriptors to binary strings [24], then an algorithm called BRIEF was developed. BRIEF has good speed and efficient storage comparing to SIFT according to result of standard benchmarks [21].

The binary strings used in BRIEF algorithm is considered as an efficient feature point descriptor. The mathematical way of producing binary strings is shown as below:

BRIEF algorithm chooses a patch around a keypoint p from a smoothed image, and then a set of $n_d(x, y)$ pairs of pixel points are chosen, after that, the pixel intensity of each pair of pixel point in the set of n_d is compared in order to produce binary values, the condition is shown as below:

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} ,$$ *Eqn. 2 [21]*

The mathematical comparison of patch p is shown as above, where p(x) is the pixel intensity in a smoothed image [21].

Then the BRIEF descriptor becomes a n_d binary string:

Eqn. 3 [21]

$$f_{n_d}(\mathbf{p}) := \sum_{1 \leq i \leq n_d} 2^{i-1} \, \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i) \ .$$

The values of n_d can be 128, 256 and 512, all the three parameter values can be supplied in OpenCV.

The most important aspect is that the BRIEF descriptor with high recognition rates gives much faster computation than SIFT descriptor by using bit string as feature descriptor, but the algorithm is only designed to get the efficient feature descriptor without providing function of feature detection, it should be used together with other algorithms such as FAST to detect feature points first [21].

2.3.5 Oriented FAST and Rotated BRIEF (ORB)

the theory of ORB algorithm:

ORB is an algorithm shown in Fig.2.7 computationally faster than SIFT with similar matching performance combining descriptor based on rBRIEF and oFAST keypoint detector [12]. rBRIEF and oFAST are improved algorithms of BRIEF and FAST respectively. More details about rBRIEF and oFAST are discussed later.

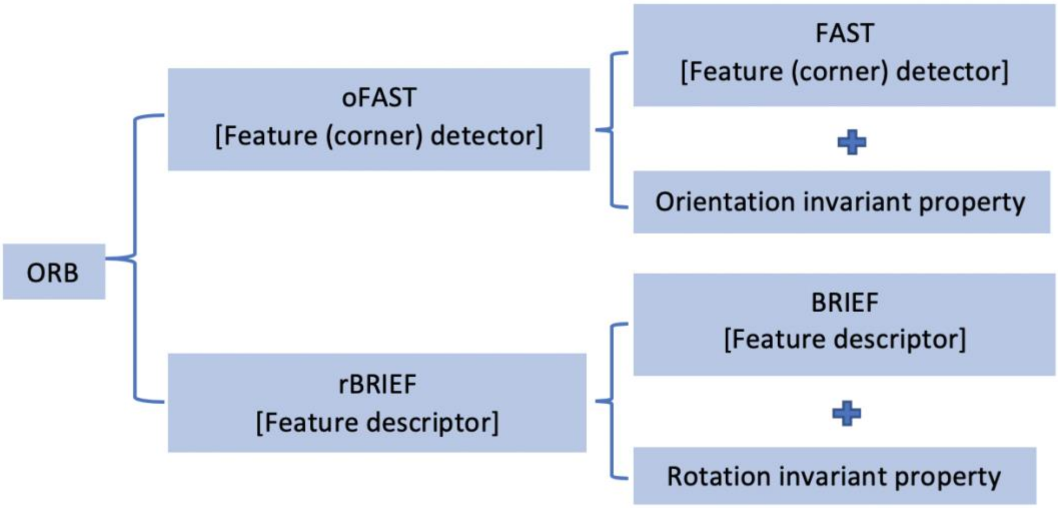


Figure 2.7: What is ORB algorithm

The ORB algorithm computes the oFAST keypoints and rBRIEF features for every image in the dataset, then a brute force matching method is used to find correspondences. Finally, the results can be obtained with the percentage of correct matches, so in theory, the percentage of correct matches larger than 75% can be used in image alignment [12].

The noise impact test results of comparison experiment between ORB and SIFT ran by Rublee et al. (2011), it can be obviously seen that image noise has much less impact to ORB, and although ORB is 100 times faster than SIFT, its error rate is higher than SIFT because of bit strings as feature descriptor [12].

<u>the invariant to orientation and scale for ORB:</u>

FAST algorithm is used to detect features in images, and the only parameter value should be determined is the intensity threshold of pixel circle [12]. FAST does not provide criterion to sort feature points, so Harris corner detector, which is a mathematical way with R values, is applied to sort the FAST keypoints [12]. For example, 100 should be the target number of keypoints, so setting a lower threshold value can remain more than 100 keypoints, maybe 150 keypoints can be obtained, then sort them according to the Harris response R values and pick the top 100 points [12]. FAST not produces multi-scale features, using Harris corner detector algorithm to quantize keypoints detected by FAST do provide a way to scale keypoints for ORB.

The approach to measure the orientation of keypoints is using the intensity centroid, which is the intensity centroid, the intensity of a corner is assumed as offset from its centre, and the vector can be used to compute orientation [25].

the invariant to rotation for ORB:

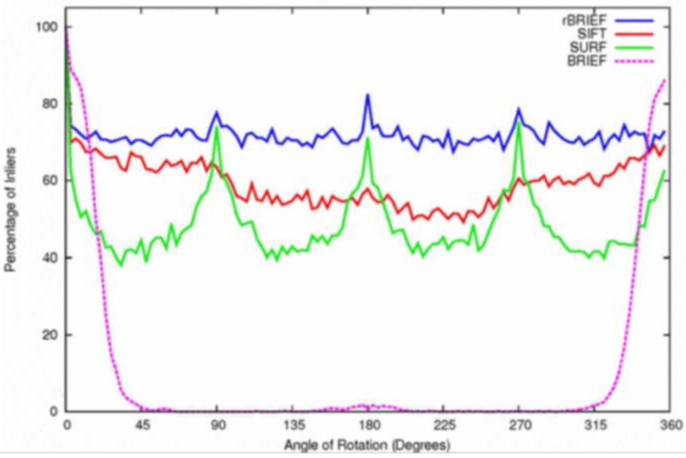


Figure 2.8: Matching performance with different rotation angle [12]

A vulnerability of BRIEF can be seen from the Fig.2.8 is that the matching performance of BRIEF is extremely instable when the angle of rotation increases, so an efficient way [12] is used in ORB so that BRIEF is expected to be invariant to rotation. The details about producing the bit string descriptor of BRIEF algorithm is shown as below:

For any feature image patch, depending on location (x_i, y_i), a $2 \times n$ matrix can be defined:

$$S = \begin{pmatrix} x_1, & \cdots, & x_n \\ y_1, & \cdots, & y_n \end{pmatrix} \quad \textit{Eqn. 4 [12]}$$

Then use:

$$S_\theta = R_\theta S, \quad \textit{Eqn. 5 [12]}$$

Where θ is patch orientation and R_θ is corresponding rotation matrix:

$$R_\theta = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \quad \textit{Eqn. 6 [12]}$$

The θ is the rotation angle. The angle can be discretized by 12, each one of them is 30 degrees and corresponding to one S_θ, then construct a lookup table of precomputed patterns of BRIEF, in the end a list of points S_θ is used to compute its descriptor.

2.4 Transformation operators

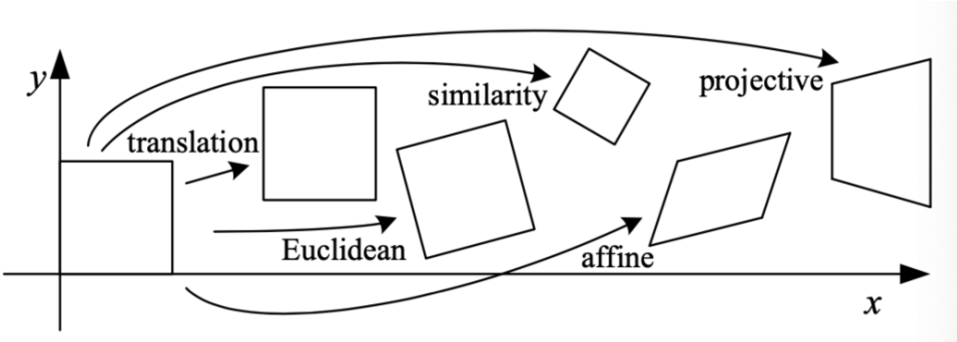


Figure 2.9: Basic transformations [4]

There are several transformation forms shown in Fig.2.9, these are basic image transformations, and for each transformation, there is a mathematical relationship between two images [4].

Chapter 3: Results

In this project, the analysis object is the motorway CCTV images, and all images tested in the experiment are from the database of the same CCTV camera.

Two methods are implemented to align these images so the aims and objectives of this project can be achieved. The performance of both methods is quantified by two factors, one is categorization result which is the number of images is correctly filtered, the other one is alignment performance.

A reference image I_{ref}, which is basically a clear background of a motorway scene without any vehicles and in a good illumination condition, is chosen as the template data in order to preserve important feature information, otherwise some images with a big truck or blurring because of rains on cameras may influence the performance of methods used in the project. The aim of this project is to find an optimized image processing method, which should be an automated way to process images, which means all unnecessary human involvement should be avoided or reduced, but in my test experiment, choosing a reference image needs little human involvement, actually in practical application, the methods should be applied to image database without knowing which one is the reference image, and depending on a random reference image, all the images in the database can be well classified and aligned, the reason I did in this way (choosing a reference image) is try to test best performance of two methods as possible as I can, so the test results show the best accuracy I can achieve.

A method proposed by Fawcett in 2006 by using four types of definitions shown as below to calculate the number of true and false matches and match failures:

1. TP: true positives, images should not be discarded and correctly classified;
2. FN: false negatives, images should not be discarded but incorrectly classified;
3. FP: false positives, images should be discarded but incorrectly classified and images should not be discarded but with bad alignment result;

4. TN: true negatives, images should be discarded and correctly classified.

The performance can be analysed by the following quantities [27]:

1. Precision (positive predictive value):

$$PPV = \frac{TP}{TP + FP} \qquad Eqn. 7\ [27]$$

2. Accuracy:

$$ACC = \frac{TP + TN}{TP + TN + FN + FP} \qquad Eqn. 8\ [27]$$

3.1 Cross-correlation

The first method using cross-correlation algorithm with MATLAB programming environment was implemented. The basic steps to implement this method are:

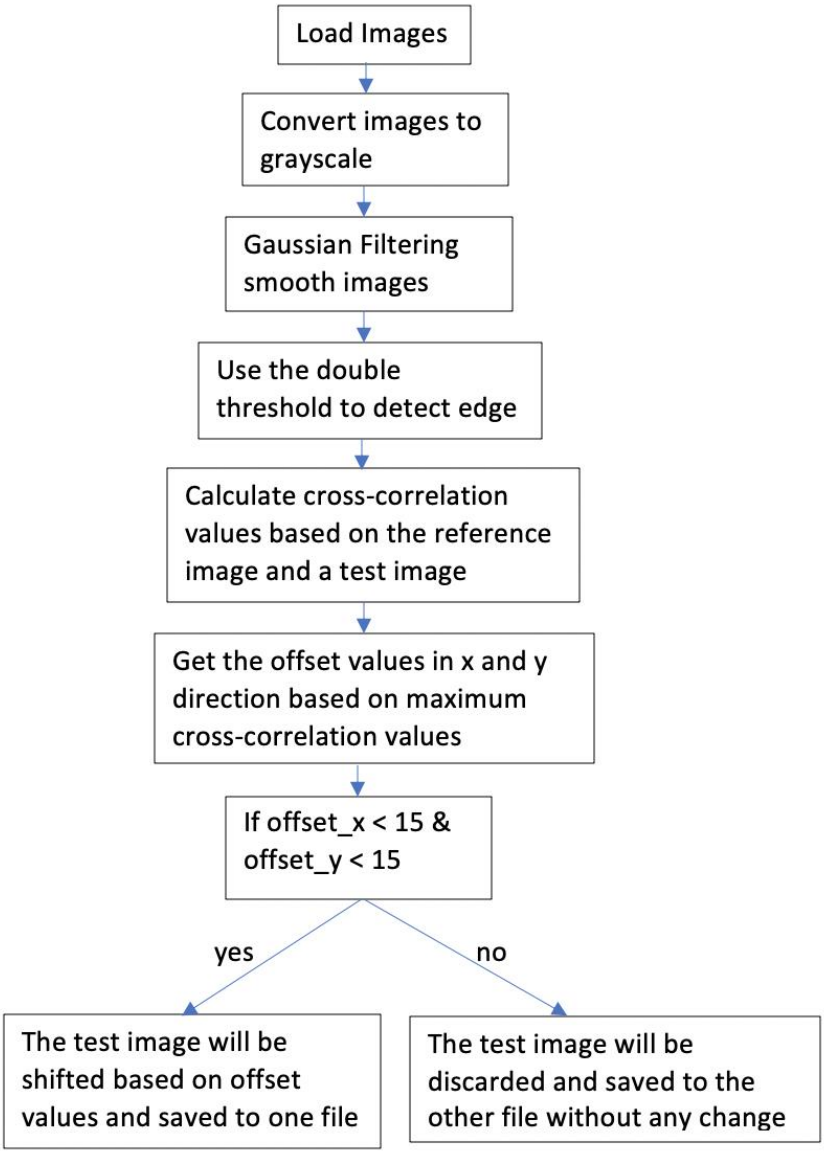


Figure 3.1: Procedures of implementing cross-correlation

Before feeding images into cross-correlation algorithm, the RGB images should be converted into grayscale images, so for each pixel, the value is in the range of 0 to 255, which is the first step for edge detection. Then Gaussian filter is used to filter noise by smoothing images, when implement Gaussian filtering in MATLAB, the Gaussian filter size is calculated by the chosen value of sigma, which is standard derivation of Gaussian distribution, to smooth images. In this test, the 3 * 3 default filter size is used in this test when the sigma equals to 0.5.

In other programming languages, upper and lower threshold values should be determined to detect edges, but for MATLAB, it chooses low and high threshold values automatically if the threshold values are not specified. In this test, it is better to choose low and high threshold values automatically, the grey levels of images are varying because of the influence of illumination condition, so fixed threshold values cannot be determined to detect edges.

The cross-correlation algorithm is used to measure the similarity between reference image I_{ref} and a test image. When calculate cross-correlation similarity of two images, the following figure shows the output of 2-D cross-correlation:

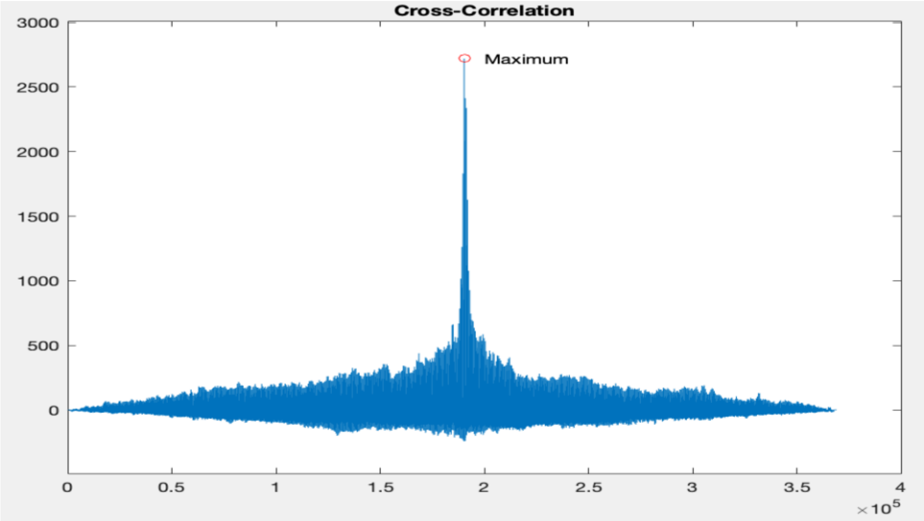


Figure 3.2: the distribution diagram of test result

The largest spike is the maximum of cross-correlation, which can give us the two-dimensional location of images, and we can use the location to calculate how many pixels the test image should shift to overlap the reference image completely.

The size of all tested images is 266 * 348 pixels.

The result of two image alignment is shown as below, in order to easily see the performance of cross-correlation method, two images overlapped, so the differences between them can be easily seen from Fig.3.3 shown as below:

two images before alignment after alignment

Figure 3.3: The performance of cross-correlation

After testing cross-correlation algorithm on more than 1000 images from database, the result can be seen in Table 1:

No. of images					Percent (%)	
TOTAL	TP	FP	FN	TN	Precision	Accuracy
1063	858	46	79	80	94.91	88.24

Table 1. The test results

It can be seen from Table 1, that the accuracy is 88.24%, which is calculated based on the number of correctly classified images, and the precision is 94.91%, which is calculated based on the number of good matching images, i.e., good performance of image alignment.

In this method, threshold value is the only parameter should be determined, and the value directly influences the result of the method.

The Fig.3.4 is the distribution of TP, FP, FN and TN as a function of threshold d which is the shift distance in theory [4], and the figure is only as a reference for choosing threshold value, I did not get the actual distribution function of threshold, because every point in the function is obtained by checking results of 1000 images manually, if I have more time, a practical distribution function can be computed.

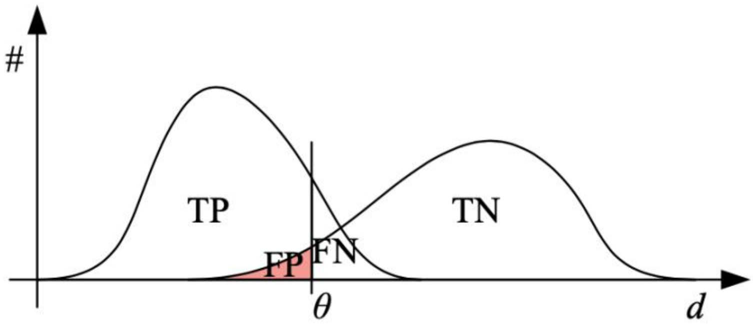


Figure 3.4: The relationship between positives, negatives and threshold [4]

From the Fig.3.4, it can be seen that when threshold value d increases, Although the number of TP (image should not be discarded and classified correctly) will increase, the number of FP (the images should be discarded in theory but classified incorrectly) will increase as well. Also, the number of TN (images should be discarded and classified correctly) will decrease and so the FN (images should not be discarded and classified incorrectly) as well.

The expected result is to let TP and TN as many as possible (the accuracy reaches maximum), and decrease the number of FP and FN, so there is a trade-off between the number of (TP + TN) and (FP + FN), and this is the reason why the threshold value is important for the result accuracy rate.

3.2 ORB

The other method I used in the project to process CCTV images is ORB algorithm and OpenCV2 library with Python. Compared to SIFT algorithm, ORB algorithm is a computationally efficient replacement to SIFT, and the other important reason is that SIFT algorithm is not free for use.

The steps of feature-based method are shown as below:

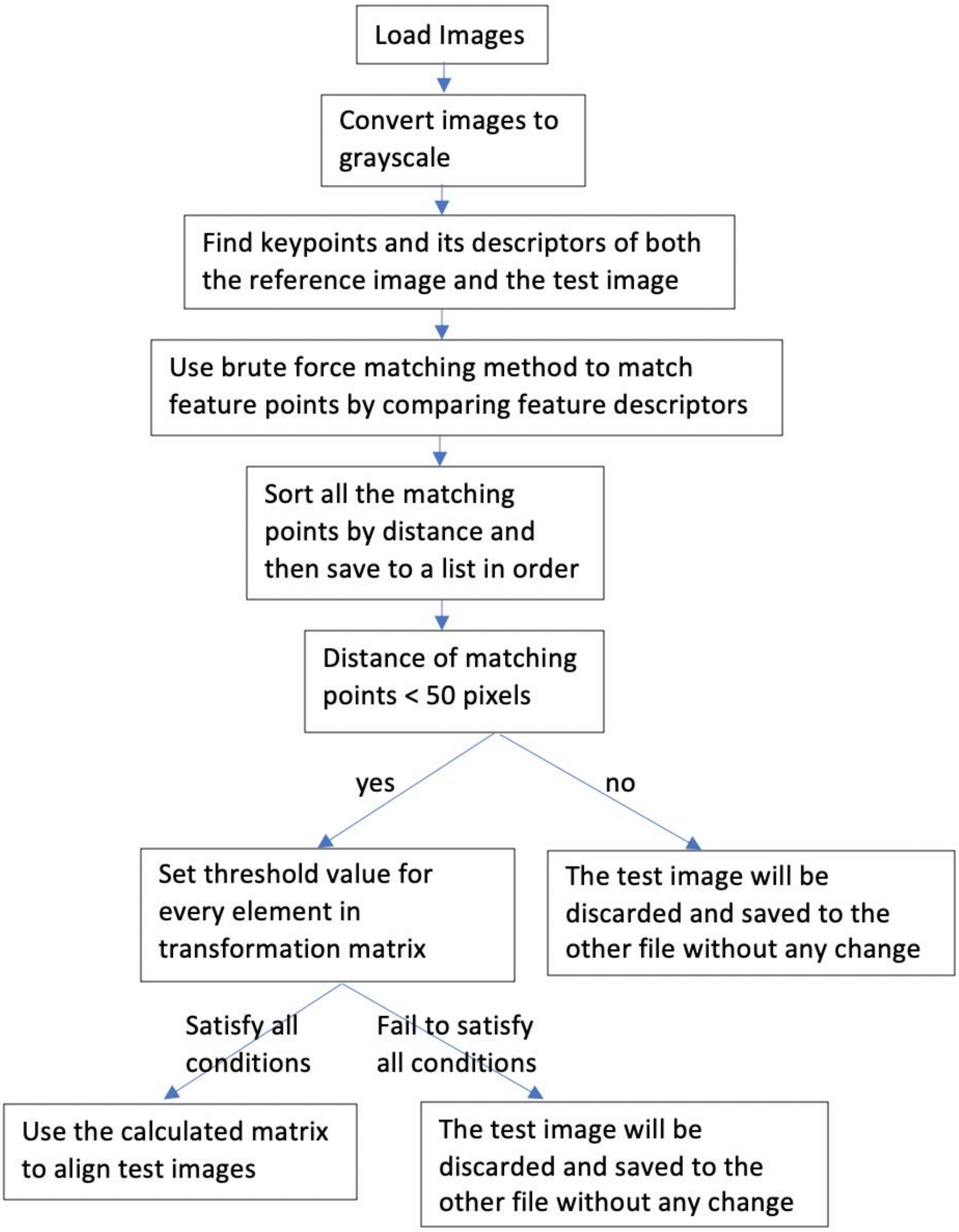


Figure 3.5: The process of implementing ORB

In this test, there are two filtering methods to improve the accuracy of matching points. Firstly, the distance threshold is set as 50 pixels, which can expand the scope of the method. Secondly, a transformation matrix, which is a key mathematical relationship between feature points of reference image and the test image, is calculated. The size of the matrix is 3*3 and the last element of the matrix is always 1, so there will be 8 parameters to align image. Because most of images' viewpoints are just changed a little, basically, the transformation matrix values should be in a certain range, so for each element in the matrix, threshold values can be used to filter matching incorrectly images in order to improve accuracy rate.

There is an example shown in Fig.3.6:

two images before alignment after alignment

Figure 3.6: The performance of ORB

The pairs of matching points can be seen as below:

Figure 3.7: An example of correspondences

Based on the same image database used previously to test performance of cross-correlation method, the threshold values for geometric transformation matrix are determined, the details about individual eight threshold values of calculated matrix are shown in the following figures.

When determined threshold values for transformation matrix, all the 8 elements in the matrix are tested by the same procedure except the last one is always 1.

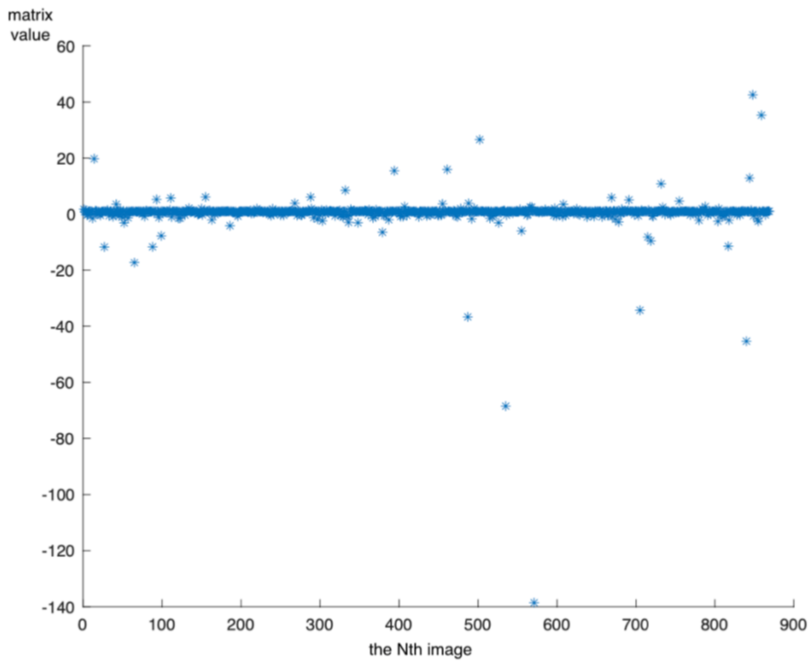


Figure 3.8: Elements in the first row and first column of matrix

The Fig.3.8 shows the discrete distribution of the element in the first row and first column in matrix of all the images. It can be seen that an explicit horizontal line there in the figure, it really helps to narrow the range of threshold values, and then the more precise values can be determined by checking image alignment results.

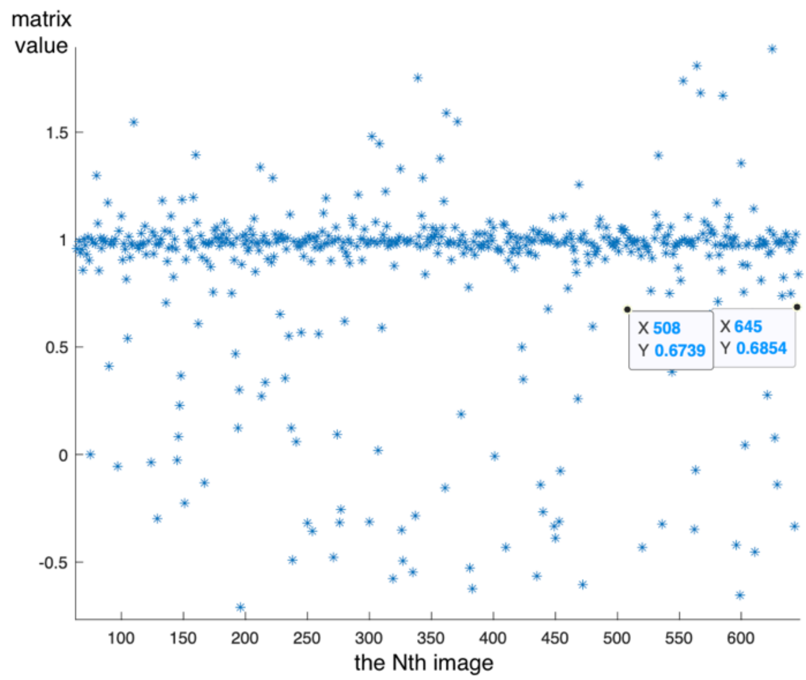


Figure 3.9: “Zoom in” version of Fig.19

More details can be seen in Fig.3.9 when zoom in the discrete distribution in Fig.3.8. In Fig.3.9, every blue point represents an image, and values in y axis show the corresponding element value in first row and first column of matrix, according to the “zoom in” version, approximate values of upper and lower threshold can be determined by checking acceptable results of image alignment.

For example, the first row and first column value in matrix of the No.508 image is 0.6739 shown in Fig.3.9, and the alignment result is shown in Fig.3.10, which looks not so great, then next step is to find image whose matrix value is larger than 0.6739, it turns out that the No. 645 image has 0.6854 matrix value, and the result is shown in Fig.3.10, which seems a good enough result, so a lower threshold value can be determined for the element in the first row and first column of matrix. It is the same way to determine upper threshold value.

No.508 image

No.645 image

Figure 3.10 The alignment result of two images

After testing and checking performance of results, the upper and lower threshold values are determined as 0.68 and 1.3, which means the value in first row and first column of geometric transformation matrix should satisfy a condition:

$$0.68 < M[1][1] < 1.3 \qquad Eqn. 9$$

The test only ran based on 1000 images, more precise threshold values can be obtained if more images can be tested.

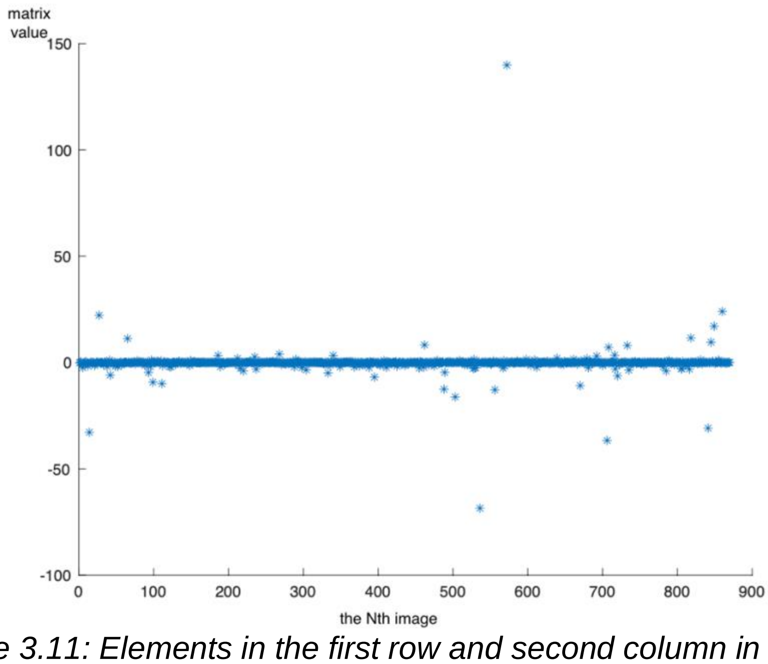


Figure 3.11: Elements in the first row and second column in matrix

The Fig.3.11 shows the discrete distribution of the element in the first row and second column in matrix of all the images. The same steps discussed before are repeated to find upper and lower threshold values in first row and second column of matrix, the result is:

$$-0.53 < M[1][2] < 0.57 \qquad Eqn.10$$

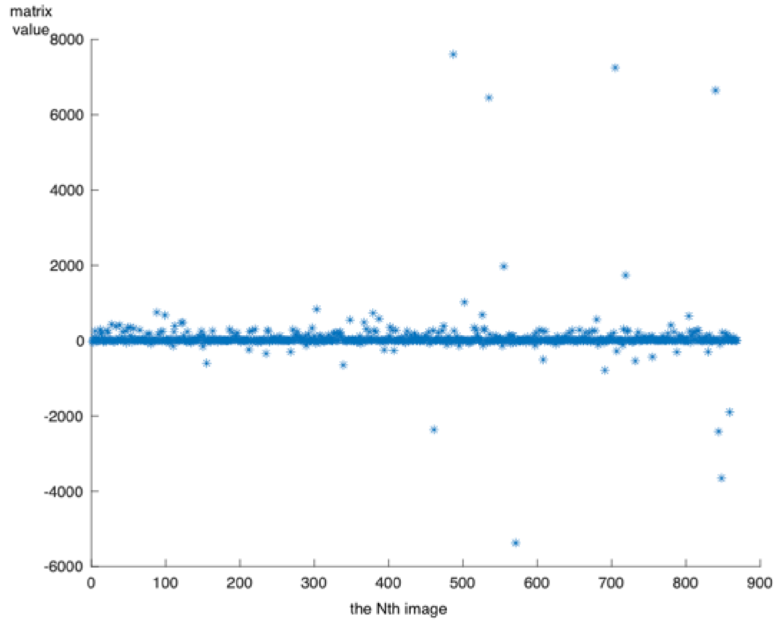


Figure 3.12: Elements in the first row and third column in matrix

The Fig.3.12 shows the discrete distribution of the element in the first row and third column in matrix of all the images. The same steps discussed before are repeated to find upper and lower threshold values in first row and second column of matrix, the result is:

$$-54 < M[1][3] < 93 \qquad Eqn.11$$

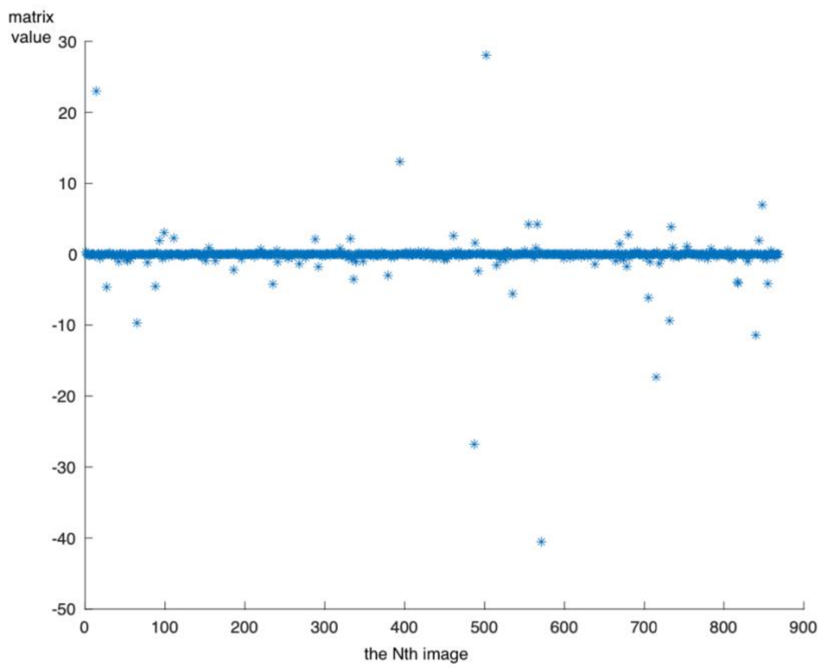


Figure 3.13: Elements in the second row and first column in matrix

The Fig.3.13 shows the discrete distribution of the element in the second row and first column in matrix of all the images. The same steps discussed before are repeated to find upper and lower threshold values in first row and second column of matrix, the result is:

$$-0.121 < M[2][1] < 0.19 \qquad Eqn. 12$$

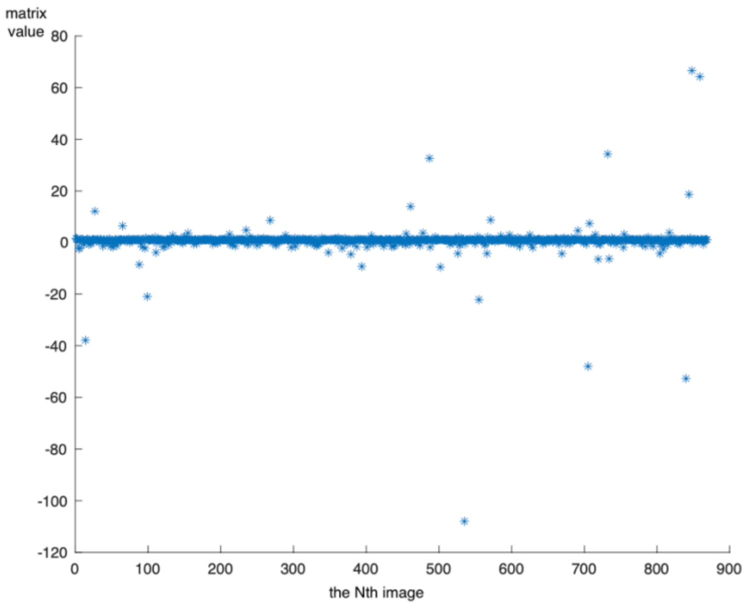


Figure 3.14: Elements in the second row and second column in matrix

The Fig.3.14 shows the discrete distribution of the element in the second row and second column in matrix of all the images. The same steps discussed before are repeated to find upper and lower threshold values in first row and second column of matrix, the result is:

$$0.7 < M[2][2] < 1.63 \qquad Eqn. 13$$

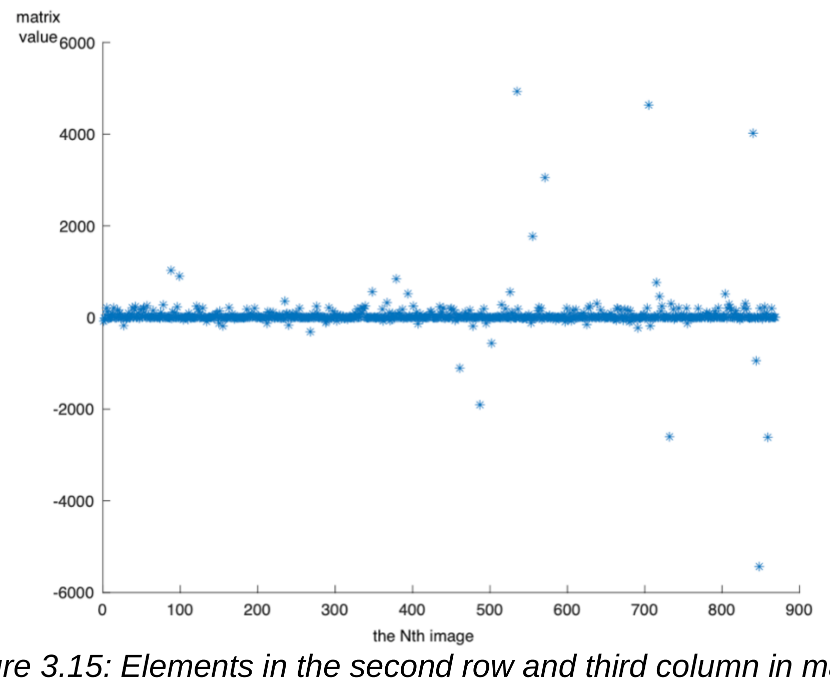


Figure 3.15: Elements in the second row and third column in matrix

The Fig.3.15 shows the discrete distribution of the element in the second row and third column in matrix of all the images. The same steps discussed before are repeated to find upper and lower threshold values in first row and second column of matrix, the result is:

$$-75 < M[2][3] < 40 \qquad Eqn. 14$$

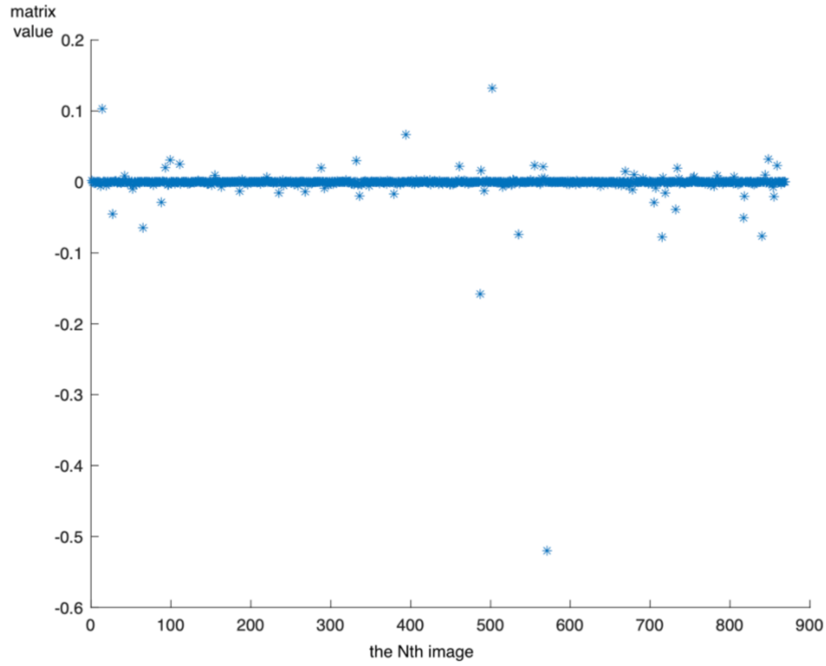


Figure 3.16: Elements in the third row and first column in matrix

The Fig.3.16 shows the discrete distribution of the element in the third row and first column in matrix of all the images. The same steps discussed before are repeated to find upper and lower threshold values in first row and second column of matrix, the result is:

$$-0.00085 < M[3][1] < 0.00092 \qquad Eqn. 15$$

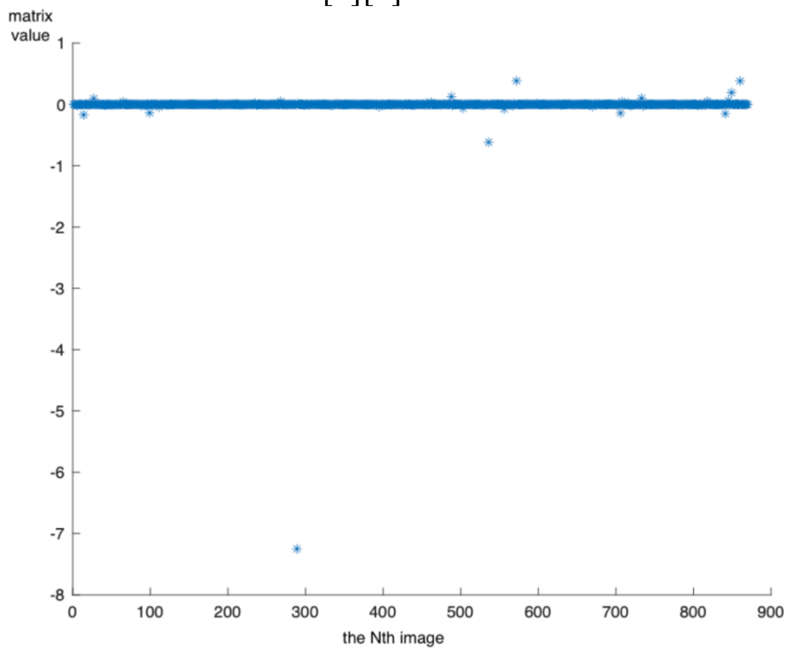


Figure 3.17: Elements in the third row and second column in matrix

The Fig.3.17 shows the discrete distribution of the element in the third row and second column in matrix of all the images. The same steps discussed before are repeated to find upper and lower threshold values in first row and second column of matrix, the result is:

$$-0.0009 < M[3][2] < 0.0013 \qquad Eqn. 16$$

After setting eight upper and lower threshold values, the precision of the result is improved a lot, the more detailed data can be seen in Table 2.

The size of all tested images is 266 * 343 pixels. After testing feature-based method on more than 1000 images from database, the result can be seen:

No. of images					Percent (%)	
TOTAL	TP	FP	FN	TN	Precision	Accuracy
1063	526	21	207	309	96.16	78.55

Table 2.

It can be seen form Table 2 is that the number of FN (the images should not be discarded in theory but classified incorrectly) is much higher than the result of cross-correlation algorithm, and the accuracy rate is also much lower.
In this test, there are several threshold values should be determined, so the error of result depends largely on the selection of threshold values.

3.3 The comparison of two methods

1. Some images can be aligned by using ORB algorithm, but cross-correlation does not work.

In Fig.3.18, this pair of images can be perfectly aligned by using feature-based method, but when I used cross-correlation method to align the same pair of images, the test image was assumed to be discarded because the offset value is larger than threshold (15 pixels).

two images before alignment After alignment

Figure 3.18: The result of after implementing ORB

2. Feature-based method ORB can deal with rotation and scale, but cross-correlation does not have these characteristics.

Original images After using ORB After using cross-correlation

Figure 3.19: The results of ORB and cross-correlation

It can be seen from Fig.3.19 that when two images at a certain angle, the cross-correlation algorithm is not useful for image alignment, but ORB algorithm can deal with image rotation because of descriptors of oBRIEF and gives us an expected result.

Original images After using ORB After using CC

Figure 3.20: *The results of ORB and cross-correlation*

It can be seen from Fig.3.20 that when test image needs scale transformation, the cross-correlation algorithm cannot achieve this, but for ORB, because a mathematical way (Harris corner detector) is used, so the feature points can be quantized to scale, which makes ORB algorithm can handle this kind of situation.

3. The accuracy of cross-correlation method is higher than feature-based method.

Original images After using ORB After using CC

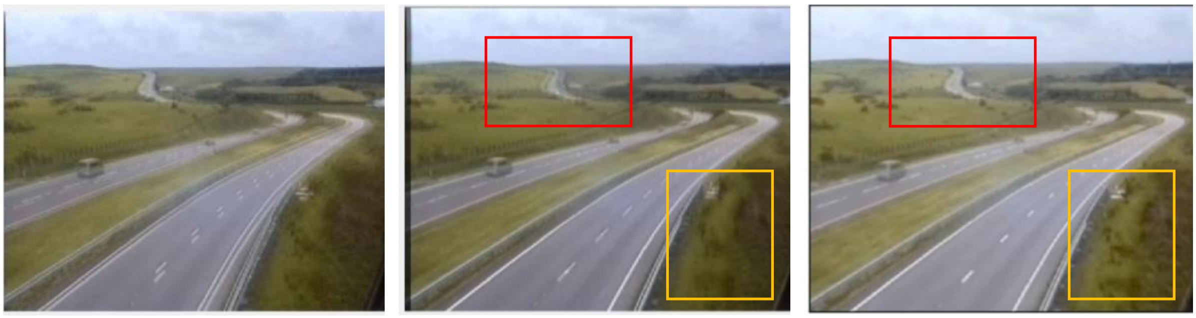
Figure 3.21: *The results of ORB and cross-correlation*

In Fig.3.21, it can be seen that the result of cross-correlation algorithm is much better than ORB algorithm, because ORB is a kind of feature-based method, so the mathematical transformation matrix is calculated based on feature points, the images with matching points can be seen from Fig.3.22 that all the meaningful feature points centralize at the upper left part of images, so this part of image can be aligned perfectly, but there are no feature points at the lower right and centre part of images, so this part of image can be seen in Fig.3.21, it is even worse than original images. For the result of cross-correlation algorithm, it is perfectly overlapped.

Because of the location of matching points cannot be controlled, so the accuracy of ORB algorithm is much lower than cross-correlation algorithm.

Figure 3.22: The correspondences between two images

4. Although cross-correlation algorithm has higher accuracy, but ORB has higher precision.



Original images Use cross-correlation Use ORB

Figure 3.23: The results of ORB and cross-correlation

It can be seen from red rectangle area in two results, the image using cross-correlation still has a “shadow” of the road, and in the yellow triangle area, the greenness can be seen clearer after using ORB algorithm. Based on analysis of the result of same images, it can be known that the ORB algorithm has higher precision than cross-correlation algorithm.

Chapter 4: Conclusion

4.1 Time plan reflection

Looking back at the time plan, the black arrows indicating when the task started and it ended, it can be seen from Fig.35 that most of tasks can be completed on schedule, but it is regretful that the start time of writing draft thesis is later than expected, one of the reason is that I was still working with my coding part and the algorithm still needs to be improved, actually, it should be done earlier so I can have more time to finish my thesis.

When wrote proposal of my project, I didn't have much background research of the project, so "find possible method" is a task to find possibly feasible algorithm to achieve the aims of project, and for now it turns out it is feature-based algorithm. At first, I thought I can find more than one method to do this project, when use theoretical knowledge to implement practical code, there is always some problems shown in the test results, then on the basis of the specific circumstance, different methods should be used to fix them, so it takes more time on background research and tests.

Actually, considering improve the performance of ORB algorithm, use deep learning is a great choice to improve accuracy and speed if there is more time left for this project. Based on the situation that a large number of images should be processed, a model can be trained to achieve the aims.

But for now, I think the objectives of this project are completed.

Figure 4.1: Time plan reflection

4.2 Conclusion

This thesis shows background knowledge and steps of implementing cross-correlation algorithm and ORB feature-based method, and all the performance of results are showed and discussed. For each method, the pros and cons are also discussed a little based on test results. All the objectives of this project are completed successfully.

All the result of tests demonstrated ORB feature-based algorithm is practical by using Python and OpenCV library, and the vulnerability of this algorithm such as accuracy can be improved by other advanced algorithm such as deep learning.

References

[1] D.E.Morris, D.S.Boyd, J.A.Crowe, et al. "*Exploring the Potential for Automatic Extraction of Vegetation Phenological Metrics from Traffic Webcams*", Remote Sens. 2013, 5(5), 2200-2218.
[2] Kyong Hwan Jin et al., "Deep Convolutional Neural Network for Inverse Problems in Imaging", IEEE Transactions on Image Processing, vol. 26, no. 9, Sept 2017.
[3] L.Brown, "A survey of image registration techniques", ACM Computing Surveys, vol. 24, no.4, pp. 326-376, 1992.
[4] R. Szeliski, "Computer Vision: Algorithms and Applications", 2010.
[5] L. M. G. Fonseca, L. M. Namikawa and E. F. Castejon, "Digital Image Processing in Remote Sensing", 2009 Tutorials of the XXII Brazilian Symposium on Computer Graphics and Image Processing, Rio de Janeiro, 2009, pp. 59-71.
[6] B. Chitradevi, P. Srimathi, "An Overview on Image Processing Techniques", International Journal of Innovative Research in Computer and Communication Engineering, Vol.2, Issue 11, November, 2014.
[7] Damir Demirovic et al., "Performance of some Image Processing Algorithms in Tensorflow", 25th International Conference on Systems Signals and Image Processing (IWSSIP), 2018.
[8] Jog, S., Dixit, M.: Supervised classification of satellite images. In: 2016 Conference on Advances in Signal Processing (CASP), pp. 93–98, June 2016.
[9] A. Procházka and O. Vyšata, "History and biomedical applications of digital signal and image processing", 2014 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), Paris, 2014, pp. 1-5.
[10] U.M.Babri, M.Tankvir and K.Khurshid, "Feature Based Correspondence: A Comparative Study on Image Matching Algorithms", International Journal of Advanced Computer Science and Applications, Vol. 7, No.3, 2016.
[11] David G.Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", IJCV, 60 (2), 2004.
[12] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF", 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2564-2571.
[13] D. G. Lowe, "Distinctive Image Features from Scale-invariant Keypoints", International Journal of Computer Vision, 60(2):91-110, 2004.
[14] N. Snavely, S. M. Seitz, and R. Szeliski, "Skeletal Sets for Efficient Structure from Motion", In Proc. Computer Vision and Pattern Recognition, 2008.
[15] S. Se, D. Lowe, and J. Little, "Mobile Robot Localization and Mapping with Uncertainty Using Scale-invariant Visual Landmarks", International Journal of Robotic Research, 21:735-758, August 2002.
[16] E. Rosten, R. Porter and T. Drummond, "Faster and Better: A Machine Learning Approach to Corner Detection", in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, pp. 105-119, Jan. 2010.
[17] L. Trujillo, G. Olague, "Synthesis of Interest Point Detectors through Genetic Programming", Proc. Eighth Ann. Conf. Genetic and Evolutionary Computation, pp. 887-894, 2006.
[18] W. Kienzle, F. A. Wichmann, B. Schölkopf, M. O. Franz, "Learning an Interest Operator from Human Eye Movements", Proc. 18th IEEE Conf. Computer Vision and Pattern Recognition Workshop, 2005.
[19] E. Rosten, T. Drummond, "Fusing Points and Lines for High Performance Tracking", Proc. 10th IEEE Int'l Conf. Computer Vision, vol. 2, pp. 1508-1515, 2005.

[20] E. Rosten, G. Reitmayr, T. Drummond, "Real-Time Video Annotations for Augmented Reality", Proc. Int'l Symp. Visual Computing, 2005.
[21] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "BRIEF: Binary robust independent elementary features", In European Conference on Computer Vision, 2010.
[22] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice", In Proc. ICCV, 2007.
[23] M. Calonder, V. Lepetit, K. Konolige, J. Bowman, P. Mihelich and P. Fua, "Compact Signatures for High-Speed Interest Point Description and Matching", Proc. IEEE Int'l Conf. Computer Vision, Sept. 2009.
[24] G. Shakhnarovich, "Learning Task-Specific Similarity", 2005.
[25] P. L. Rosin. Measuring corner properties. Computer Vision and Image Understanding, 73(2):291-307, 1999. 2
[26] K. Patoommakesorn, F. Vignat and F. Villeneuve, "The 3D Edge Reconstruction from 2D Image by Using Correlation Based Algorithm," 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), Tokyo, Japan, 2019, pp. 372-376.
[27] Fawcett, T, "An introduction to ROC analysis. Pattern Recognition Letters", 27(8):861–874, 2006.
[28] Harris, C. and Stephens, M. J,"A combined corner and edge detector", In Alvey Vision Conference, pp. 147–152, 1988.
[29] L. Yuan and X. Xu, "Adaptive Image Edge Detection Algorithm Based on Canny Operator," 2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS), Harbin, 2015, pp. 28-31.
[30] X. Dong, M. Li and J. Miao, "Edge Detection Operator for Underwater Target Image", 2018 IEEE 3rd International Conference on Image Vision and Computing (ICIVC), pp. 91-95, Jun 27, 2018.
[31] L. Wan-She and Z. Jin, "Application of wavelet transform in edge detection", Proc. Int. Conf. Image Signal Process. (CISP), pp. 2173-2176, 2011.
[32] S. Guiming and S. Jidong,"Multi-Scale Harris Corner Detection Algorithm Based on Canny Edge-Detection, "2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET), Beijing, 2018, pp. 305-309.
[33] Schowengerdt, R, "A Remote Sensing Models and Methods for Image Processing", Academic Press, 1997.
[34] B. Alberton, R.D.S. Torres, L.F. Cancian, B.D. Borges, J. Almeida, G.C. Mariano, J. dos Santos, L.P.C. Morellato,"Introducing digital cameras to monitor plant phenology in the tropics: applications for conservation", Perspect. Ecol. Conserv., 15, 2017.
[35] “xcorr2”, MathWorks. [online]. Available: https://ww2.mathworks.cn/help/signal/ref/xcorr2.html#d120e184597. [Accessed: 29 – April - 2020]