การประเมินสมรรถนะของตัวบรรยายท้องถิ่นสำหรับการรู้จำใบหน้า

นางสาว มูห์ฟิซาทูราห์มาห์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556

PERFORMANCE EVALUATION OF LOCAL DESCRIPTORS FOR FACE RECOGNITION

Ms. Muhfizaturrahmah

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Electrical Engineering

Department of Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2013

Thesis Title	PERFORMANCE EVALUATION OF LOCAL DESCRIPTORS FOR FACE RECOGNITION
By	Ms. Muhfizaturrahmah
Field of Study	Electrical Engineering
Thesis Advisor	Assistant Professor Supavadee Aramvith, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

....Dean of the Faculty of Engineering

(Professor Bundhit Eua-arporn, Ph.D.)

THESIS COMMITTEE

....Chairman

(Assistant Professor Widhyakorn Asdornwised, Ph.D.)

....Thesis Advisor

(Assistant Professor Supavadee Aramvith, Ph.D.)

....Examiner

(Assistant Professor Charnchai Pluempitiwiriyawej, Ph.D.)

....External Examiner

(Sanparith Marukatat, Ph.D.)

มูห์ฟิซาทูราห์มาห์ : การประเมินสมรรถนะของตัวบรรยายท้องถิ่นสำหรับการรู้จำใบหน้า. (PERFORMANCE EVALUATION OF LOCAL DESCRIPTORS FOR FACE RECOGNITION) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร. สุภาวดี อร่ามวิทย์, 71 หน้า.

การรู้จำใบหน้าเป็นหนึ่งคุณลักษณะที่มีความโดดเด่นทางไบโอเมตริก ซึ่งประกอบด้วยสามขั้นตอนหลัก คือ การตรวจหาใบหน้า การสกัดคุณลักษณะ และการจับคู่คุณลักษณะ สำหรับการสกัดคุณลักษณะจะถูกพิจารณาความสำคัญในฐานะตัวแทนที่มีลักษณะเฉพาะสูง ที่ถูกสร้างขึ้นโดยคุณสมบัติบางประการ นักวิจัยได้มีการพัฒนาตัวบอกคุณลักษณะท้องถิ่นมากมายบนพื้นฐานของคุณลักษณะที่มีความสำคัญ ซึ่งได้ถูกนำเสนอเพื่อใช้งานสำหรับภาพทั่วไปได้อย่างดี แต่ในกรณีของภาพใบหน้าซึ่งมีคุณลักษณะเฉพาะ การตรวจสอบประสิทธิภาพของตัวบอกท้องถิ่นดัง กล่าวจึงมีความจำเป็นเพื่อให้ได้สมรรถนะของการรู้จำที่ดีที่สุดที่เป็นไปได้ ในวิทยานิพนธ์นี้นำเสนอการประเมินสมรรถนะของตัวบอกคุณลักษณะท้องถิ่นสำหรับภาพใบหน้า โดยตรวจสอบตัวบอกคุณลักษณะท้องถิ่นที่มีความทันสมัยประกอบ ด้วย Scale Invariant Feature Transform (SIFT) Speed Up Robust Features (SURF) Binary Robust Independent Elementary Features (BRIEF) Binary Robust Invariant Scalable Keypoints (BRISK) และ Fast Retina Keypoint (FREAK) โดยตรวจสอบสมรรถนะด้วยภาพที่มีการแปลงคุณลักษณะต่างๆ คือ ขนาด ความพร่ามัว การหมุน ความสว่าง และท่าทาง การวัดสมรรถนะประกอบด้วย ความสามารถในการทวนซ้ำ ความแม่นยำ อัตราการรู้จำ และเวลาที่ใช้ในการคำนวณถูกใช้ในการวัดเชิงคุณภาพ ผลการทดลองแสดงให้เห็นว่าตัวบอกคุณลักษณะท้องถิ่นแต่ละชนิดมีประสิทธิภาพในการสกัดคุณลักษณะในสถานการณ์ที่แตกต่างกัน ดังนั้นการเลือกตัวบอกคุณลักษณะท้องถิ่นจึงต้องใช้ลักษณะเฉพาะของสถานการณ์ และสภาพแวดล้อมในการพิจารณา

ภาควิชา...........วิศวกรรมไฟฟ้า............ลายมือชื่อนิสิต...
สาขาวิชา...........วิศวกรรมไฟฟ้า............ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก....................
ปีการศึกษา................2556...................

5470517421 : MAJOR ELECTRICAL ENGINEERING
KEYWORDS : PERFORMANCE EVALUATION / FACE RECOGNITION / FEATURE EXTRACTION / LOCAL DESCRIPTOR

MUHFIZATURRAHMAH: PERFORMANCE EVALUATION OF LOCAL DESCRIPTORS FOR FACE RECOGNITION
ADVISOR : ASST. PROF. SUPAVADEE ARAMVITH, Ph.D.,71pp.

Face recognition (FR) is one of prominent feature in biometric. There are three main steps in FR: face detection, feature extraction and feature matching. Among three steps, feature extraction is considered important as highly distinctive representation will be constructed using certain properties. Researchers have developed several local feature descriptors. However, such those features are proposed to perform well for generalized images. In case of face images which possess distinct feature, it is necessary investigating the effectiveness of such local descriptors to achieve the best possible recognition performance. In this thesis, performance evaluation of local feature descriptors for face images is conducted. Several state of the art local descriptors including Scale Invariant Feature Transform (SIFT), Speed Up Robust Features (SURF), Binary Robust Independent Elementary Features (BRIEF), Binary Robust Invariant Scalable Keypoints (BRISK) and the Fast Retina Keypoint (FREAK), are investigated. We investigate the performance using transformed images with the following properties: scale, blur, rotation, brightness, pose. The performance parameters including repeatability, precision, recognition rate, and computational time are used as measurements. The results indicate that each local descriptor is considered effective in extracting features in different scenarios. Thus, the consideration of choosing local descriptor should take the characteristic of scenario and environment into account.

Department : Electrical Engineering Student's Signature..................

Field of Study : Electrical Engineering Advisor's Signature..................

Academic Year : 2013

Acknowledgements

In the name of Allah, the Lord of universe, the Most Gracious and the Most Merciful. First of all, I would like to thank Mama (Mardiati) and Abah (Mat'hani) for their endless trust and support for me. My elder sister (Ezzatunnor) and her husband (Mas Teguh Kustyo B.) who always support me in every step the way, also my little niece, Salwa Salsabila, my spirit who bring me up everytime I fall. I climb up because of you.

I am so grateful to have a chance to become one of Arjarn Supavadee Aramvith's advisee. Thank you very much Arjarn, for your patience and guidance. I would like to thank Arjarn Widhyakorn Asdornwised, not only for his role as chairperson of thesis committees but for his way on teaching that encourage me, Arjarn Charnchai Pluempitiwiriyawej and Dr. Sanparith for their beneficial comments during my proposal until final defense. I owe Mr. Canggih Puspo Wibowo and Mr. Nathachai Watcharapinchai who guided me to have better knowledge on computer programming.

It is impossible for me to pass this endeavor without so many times sharing knowledge with groupmates: Kanokphan L., Adisorn P., Htoo Maung Maung, Watchara R., Sirawich S., At this moment also I would like to say my gratitude to have Nur Soleha, Hai Yen Thi Nguyen, Emilia P. Sevilla and Silvy Astari who stays fighting with me as helpful and very understanding roommates.

I am blessed to have lifetime friends: Kasuma Yuda, Nurul Oktarina and Fadillah who always believe in me and pushing me to reach even farther that I could have imagined before. Last but not least, I would also thank my fellows in Indonesian Students Association in Thailand (PERMITHA) for one whole wonderful year as staff, hand in hand, work together made our PERMITHA become more productive.

Contents

List of Tables

Table Page

List of Figures

Page

CHAPTER I

INTRODUCTION

1.1 The Significance of Research

In development of human perception, computer vision and pattern recognition field, face recognition (FR) demand undeniably has become one of the most ubiquitous, spurred by reduced prices of cameras and also by the escalated processing technology. Research of face recognition becomes inherently developed towards the increasingly demand of facial-based biometric. Among biometrics, face recognition has its own privilege that is the database enrolment does not need subject's cooperation thus this biometric is more flexible than others such as fingerprint, iris, etc.

Face Recognition inevitably has become part of modern life. Surveillance, information security, smart-gate, forensic are only some examples to name it. Lots of researches have been developed face recognition technology for more than two decades to enhance the performance of FR for both precision and computational time. However, the challenge also increases because today the format of input not only still-images but also video. Video contains more disturbance than still-image plus a video is a moving sequential frames that need more efforts in analysis. Closed Circuit Television (CCTV) camera is one common simple example. Almost every spot in the city where most of people gather such as mall, city garden, station, airport are under coverage of surveillance camera network.

Unlike still-image face recognition technology that relatively achieves very trusted result so far, video-based face recognition has been attaining less result in terms of robustness due to improper information gaining from input captured image. One instance is real-time surveillance face recognition system. The fact that real-time video-based face recognition is still far to become robust is caused by several factors. First, most of commercial low price camera has low quality or in other words the resolution of video produced by using this camera will be too poor to be able to be processed furthermore to be recognized. Second, in case partially face identification is done by human, multi-camera network with long hours video made face identification by human power is not efficient and exhausting. Human is good to recognize known faces but human are not trained to identify the large number of unknown faces. Third, distance between face and the camera is far with unconstrained environment, face movement, various pose, illumination and expression come together brings face recognition algorithm today suffer or even failure in such scenario.

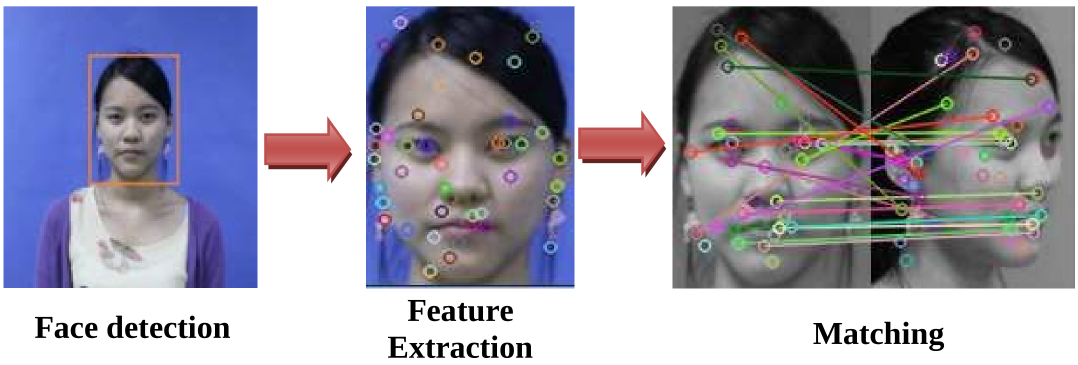


Figure1.1: General pipeline of face recognition.

In general concept, face recognition process consist of three main steps that involved face detection, feature extraction and descriptor matching as shown in Figure.1.1. An input image or video which contained single person or many persons were observed to locate the position of face(s). Face detection is how the most likely face region is separated from entire scene of image plane. It is important even crucial part since if the face region is not segmented, it will be impossible to carry out the recognition process. Too many redundant and unnecessary features will be detected which costly and wasting time. Second step is extracting meaningful traits and transform it into such domain that represents highly distinctive information that could be certain face area or measures (vector). The final step is matching. The extracted descriptor from query images are compared to the database descriptors sets and seek its best match features set to eventually declare subject's identity. Identity could be name, address, and others.

In order to achieve high recognition rate of automated face recognition system, robust and invariance feature is needed to be extracted. Local feature descriptor is one of powerful feature extractor that gained more attention nowadays especially from computer vision world. Its pattern contrast makes it saliently differs to the neighborhood and become strong distinctive descriptor. Local descriptor can be edge, corner, points, blob, etc. A set of local features can build a distinguish landmark that allow recognition even without segmentation process before. Each local descriptor offers its own advantage. We cannot generalize which one is the most robust. The selection of descriptor should depend on application and scenario. Robustness to background and occlusion and invariance to image transformation and deformation made local descriptor suitable for face recognition particularly real-time face recognition that contain noise, environmental changes and motion blur.

1.2 Literatures Review

Tracing back the beginning of local feature detection method was started by the knowledge that information on shape is focused at salient points having high curvature published by Attneave in 1954 [1]. The focus was especially on the accurately localizing the candidate points. There was understanding that intersection of straight lines and straight corners indicates a strong feature also. Corner become more powerful than both flat and edge regions since it has substantial change in every direction as shown in Figure 1.2.

There are several clusters of local feature in bibliography [2]. First, **contour curvature-based methods** and the second are **intensity-based methods.** Hessian-based approach [3] is categorized under intensity-based methods. The third cluster is called **biologically plausible methods**. Principally, this method proposed in visual recognition and artificial intelligence. In general, this cluster had been developed without a particular application purpose with the main goal was to make the model of human brain process. Innumerable models of human visual interest or saliency were found in Computer Vision and Cognitive Psychology literatures. Nevertheless, most of the works solely of theoretical interest and only a little number were implemented and assessed with real images. Color allows extra information which can be explored further to obtain more information in feature extraction which is the fourth cluster of local feature named **color based methods**. Some of biologically plausible methods as in [4] and [5] used color information. Other is **model-based methods**, this fifth cluster is aimed in improvement of the accuracy detection of Hessian-based corner detector [3], example for model-based methods were introduced in [6] and [7]. The sixth cluster, **Toward viewpoint invariant methods** is divided into three sub-cluster that are multi-scale methods [8, 9], scale invariant methods [10, 11, 12, 13, 14, 15] and affine invariant methods [16, 17]. The last two clusters are **segmentation-based methods** [18, 19, 20] and **machine learning-based methods** [21].

Further study about toward viewpoint invariant methods brought to conclusion that corner is a strong region where interest points lying. Corners can be found at various kinds of intersection, e.g. on dense textured image plane. It is sufficient or many practical applications, because the aim is to find a number of stable and repeatable features.

In 1988 Harris Corner Detector [22] was invented. This method facilitates elements from different angle of view to be matched using image patches of fixed size. The necessity of extracting elements with flexible image patches led Lindeberg [12] to use Laplacian of Gaussian. Stimulated by Lindeberg work, scale invariant features was born ten years later. Unfortunately, second order derivatives are sensitive to noise. The blurring process inside the algorithm smoothes the noise out and stabilizes the second order derivative. Despite of that advantage, second order derivatives are too computationally expensive [23].

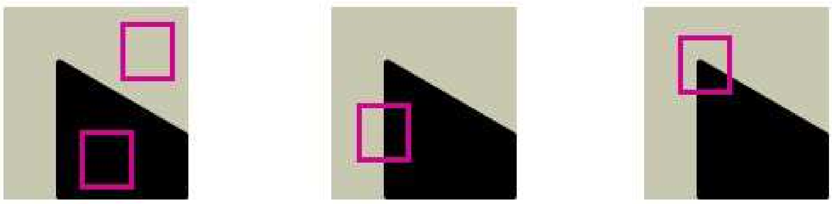


Figure 1.2: Illustration of flat, edge and corner region on image.

So far the most detectors talked extract features at a sole scale, detector internal parameters is determined factor. Late of the 1990s, the usage of local features was rising in the context of object recognition and large gallery matching, there was an increasing need to build features that able to cope with broad range scale changes and further even pose changes. Dealing with various scale changes, a direct approach consists of extracting points over a range of scales and using all the points all together to represent the image. This is referred to as a multi scale or multi resolution approach [24]. To deal with the many overlapping detections, typical of multi-scale approaches that resulting in scale-invariant property have been introduced. These methods automatically calculate the location of features and also scale of those features. A famous and powerful algorithm for object recognition utilized local minima and maxima in the scale-space pyramid built with Difference of Gaussian (DoG) filters was introduced in [25].

Before discussion goes deeper to elaborate recent local feature methods, here we explain some terminology generally used in this field.

1. Detector vs. extractor. There are two main tasks in feature extraction methods. The first is the process to localize the features on image plane that called feature detection. The second one is feature description, the detected featuress are built into specific form that are uniquely differentiated. From that we derive terms ‘detector’ and ‘descriptor’. Some people call descriptor as extractor.
2. Invariant vs. covariant [2]. A function is said invariant if its value does not change when a certain transformation is applied to its argument. A function is said to be covariant when it replaces along with the transformation. In other words, applying the transformation to the function argument will have the same effect as applying the transformation to the function output.
3. Interest Point, Region or Local Feature [2]. The ideal local feature would be a point as defined in geometry. It will possess location in space without having spatial areas. Nevertheless, in practical world, images are discrete and having pixel as the smallest spatial unit and discretizing effects playing an important role. To localize features in images, a local neighborhood of pixels has to be outlined, giving all local features some implicit spatial extent.

Discussing about features extraction brought us to the desired qualities that we would like to produce by constructing such feature descriptor. Here are the ideal local feature properties [2]:

1. Repeatability: Given two images of the same object, taken under different conditions, a set feature detected and extracted on the first image should be able to

be detected and extracted on the second image with the ability to correspond those features.

2. Distinctiveness: Intensity of the patterns underlying the detected features should reveal large of variation, thus the features able to be differentiated and matched.
3. Locality: The features should be local, so as to reduce the probability of occlusion and to allow simple model approximations of the geometric and photometric deformations between two images taken under different viewing conditions
4. Quantity: Total number of detected features should be adequately big, so that even on low small objects, reasonable number of features is still located. However, the optimal number of features depends on the application.
5. Accuracy: The features located should be precisely localized.
6. Efficiency: Time consumed for feature extraction should permit the time-critical applications.

1.3 Objectives

In this thesis we do performance evaluation of recent feature extraction methods in literatures with more focus on local descriptors as part of feature extraction process. This thesis work has several objectives as follows:

1. Quantify the behavior of local feature extraction methods with transformed images as query images.
2. Determine the quality of local feature descriptors in identification of face.
3. Give recommendation how to select detector and descriptor in order to work effectively in certain scenario of face recognition.

1.4 Scope of Work

The scope of the performance evaluation of local feature extraction methods are represented as follows:

1. Quantify the repeatability score, precision, recognition rate and also computational time for each local feature extraction method.
2. Evaluate the measurements produced and analyze the causes of such behavior.

1.5 Research Procedures

1. Reviewing literatures related to automatic face recognition, local feature extraction, facial image detection, and image matching.
2. Collecting facial image database.
3. Proposing a performance evaluation to measure behavior of feature extraction methods with various deformation and transformation on query images.
4. Simulating the protocol of evaluation using several face databases.
5. Analyzing the result
6. Writing publication for international conference.
7. Writing thesis paper.

CHAPTER II

LOCAL FEATURE EXTRACTION METHODS

There are two main processes in feature extraction. The first one called feature detection is the process which the information from image is collected and local decision is made at every point to determine whether there is feature point on that point or not. The produced feature points will be subsets of the image domain. The features can be the form of isolated points, connected regions or continuous curves. If the input data is too large and information on it expected to be adversely redundant then the better representation is necessary. The representation is made in a process called feature description. Some people named the feature as keypoint or interest point. However those terms is just used in order to make a better and easier explanation. In this work we agree to wholly use feature detection and feature description as the terms.

In this chapter, focus of the discussion covers all related feature detectors that are Hessian Detector, Harris Corner Detector, Difference of Gaussian or DoG, Features from Accelerated Segment Test (FAST), Adaptive and Generic Accelerated Segment Test (AGAST). The discussion also includes the recent local feature descriptor in literatures. The referred feature descriptors are Scale Invariant Feature Transform (SIFT), Speed Up Robust Features (SURF), Binary Robust Independent Elementary Features (BRIEF), Binary Robust Invariant Scalable Keypoints (BRISK), and Fast Retina Keypoints (FREAK).

2.1 Feature Detector

In this thesis, detector that we used mainly is corner-based detector which yields strong features because their substantial change of intensity. However there will be described a glance of the prominent feature detector since its initial development in history.

2.1.1 Hessian Detector

One of the earliest feature detectors is Hessian detector [3] that proposed in year 1978 by Beaudet. This method based on Hessian matrix H on image with I intensity function. $I_{xx}(x)$, $I_{xy}(x)$, $I_{xy}(x)$, $I_{yy}(x)$ represent the terms for second order partial derivatives of I at location $x = x, y$. The matrix H is computed using (2.1). After determinant value of H is calculated, the features will appear over the image.

$$H = \begin{bmatrix} I_{xx}(x) & I_{xy}(x) \\ I_{xy}(x) & I_{yy}(x) \end{bmatrix} \tag{2.1}$$

$$\det H = I_{xx} I_{yy} - {I_{xy}}^2 \tag{2.2}$$

This method eliminates non extrema features by using 3 x 3 windows and takes only the points having higher value than its 8 neighbors.

2.1.2 Harris Corner Detector

Harris Corner Detector is the famous method for corner detection on image plane invented by Chrish Harris and Mike Stephens back in 1988 [22]. The basic thought of this method is that at an intersection there are two edges available, which there is massive intensity change. Therefore, the value of gradients on both edge directions has a high variation, which can be utilized. Second moment of Matrix M is the base of this method. The matrix M enables to describe the alteration of intensity in the local neighborhood at certain point x.

Imagine there is an image window as shown with yellow rectangle in Figure 2.1. As we can carefully see, even the slight move produces noticeable appearance. The displacement happened is following (2.3).

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2 \tag{2.3}$$

Where $w(x, y)$ is the window at point (x, y), $E(e, u)$ is the displacement, $I(x, y)$ is the intensity at point (x, y) and $I(x + u, y + v)$ is the intensity after displacement.

The search is looking for windows with major intensity variation. Therefore, we have to maximize the equation (2.3) such way so that value for equation inside the square brackets has a large value. This can be done by using Taylor expansion. Here, $I(x+u, y+v)$ becomes $I(x,y) + uI_x + vI_y$ as described in (2.4) and derived until (2.8).

$$E(u,v) \approx \sum_{x,y} \left[I(x,y) + uI_x + vI_y - I(x,y)\right]^2$$

$$E(u,v) \approx \sum_{x,y} u^2 {I_x}^2 + 2uvI_xI_y + v^2\, {I_y}^2 \tag{2.4}$$

$$E(u,v) \approx [u\ v]\left(\sum \begin{bmatrix} {I_x}^2 & I_xI_y \\ I_xI_y & {I_y}^2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}\right) \tag{2.5}$$

$$M = \sum w(x,y) \begin{bmatrix} {I_x}^2 & I_xI_y \\ I_xI_y & {I_y}^2 \end{bmatrix} \tag{2.6}$$

$$E(u,v) \approx [u\ v]M \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.7}$$

$$R = \det M - k(traceM)^2 \tag{2.8}$$

R score is calculated for each window, when the value is above certain value then the window region is found as corner.

Figure 2.1: Window movement of Harris Corner search.

2.1.3 Difference of Gaussian (DoG)

Harris corner detector is good to locate corners but when image size is various, the scale of window also changes thus this method cannot perform effectively. Scale-space is needed in order to obtain scale-invariant property. In Difference of Gaussian (DoG), scale space is generated by resizing images from original images into half size then blurred it by using Gaussian filter with certain blurred level. The resizing and blurring process are repeated depend on the number octave decided. The number of octave depends on original image size but in original method [25], inventor suggested four octaves and five blur levels were ideal for the algorithm. This process follows (2.9). Where $L(x, y, \sigma)$ is produced from convolution of Gaussian $G(x, y, \sigma)$ with $I(x, y)$ input image. Visualization of blurring process is illustrated on Figure 2.1.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.9}$$

LoG is approximated by Difference of Gaussian (DoG) as shown in Figure 2.3 and Figure 2.4. The Laplacian of Gaussian is great for finding features in an image. But it comes with disadvantage because second-order derivatives are computationally expensive [23]. Instead of using LoG, this method approximates LoG using difference between two consecutive scales. Replacing second order derivatives into simple subtraction. Generation of DoG images is represented on Figure 2.2.

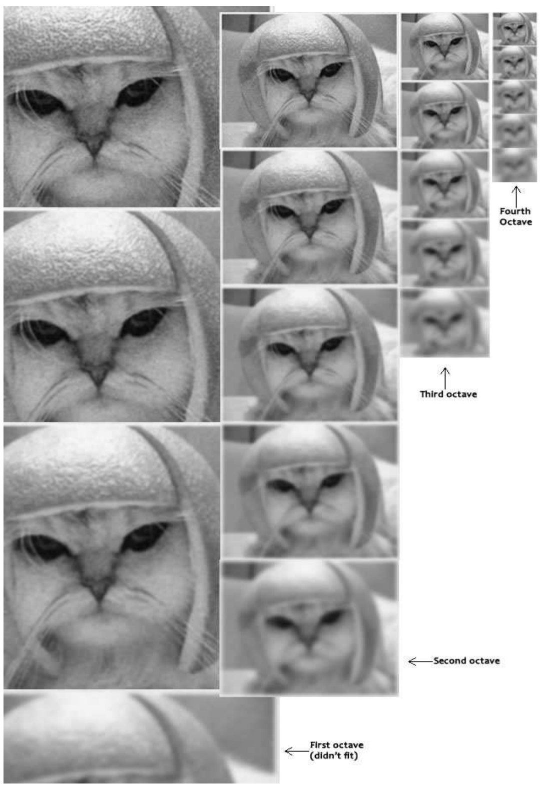


Figure 2.2: Progressive generated blur images of SIFT [26].

Finding maxima and minima on DoG images around approximate feature as shown in Figure 2.5 by using Taylor Expansion (2.10).

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x. \tag{2.10}$$

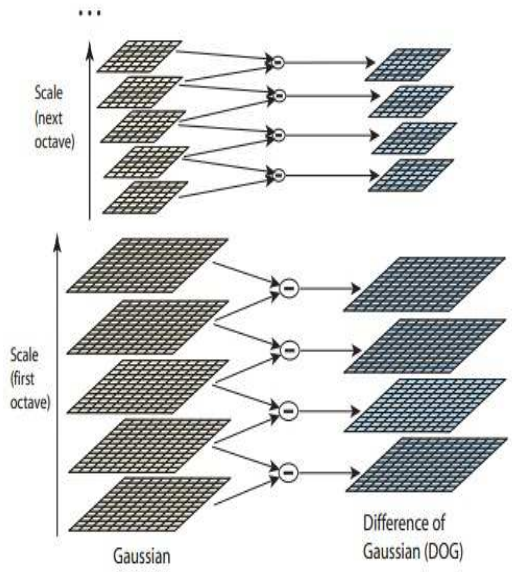


Figure 2.3: Difference of Gaussian (DoG) of an image [25].

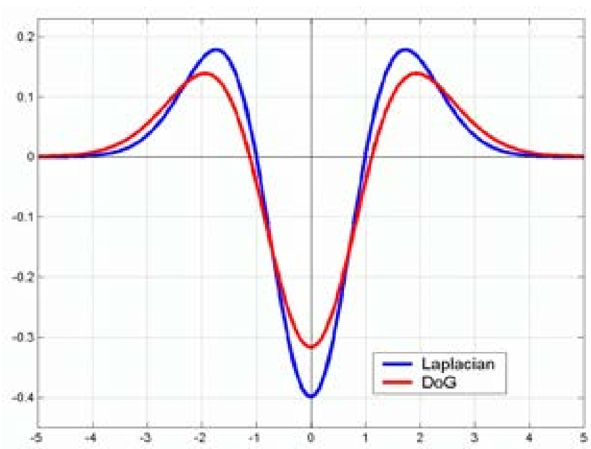


Figure 2.4: Difference-of-Gaussian values approximate Laplacian-of-Gaussian values [25].

The last step is to remove feature with low contrast and edges. Features generated in the previous step produce a lot number of features. Some of them lie along an edge, or they do not have enough contrast. In both cases, they are not useful as features. So those features were omitted. Elimination of bad features approach is similar to the one used in the Harris Corner Detector for removing edge features. For low contrast features, by simply checking their intensities.

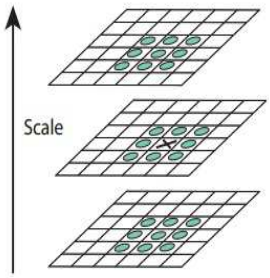


Figure 2.5: Maxima and minima of DoG images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles) [25].

2.1.4 Features from Accelerated Segment Test (FAST)

Difference of Gaussian is reliably scale-invariant detector but unfortunately the cost comes with complex computation yields long computational time in detection process. When it comes to real-time application, such a long computational time is unwanted. FAST as machine learning-based method is proposed to achieve high speed computation by E. Rosten and T. Drummond in 2010 [27].

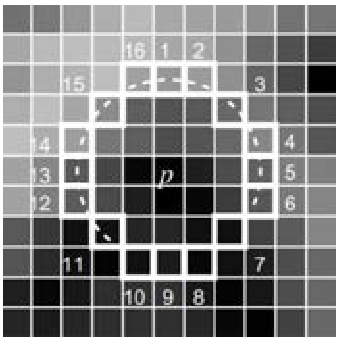


Figure 2.6: FAST candidate corner surrounded by its 16 pixel circular neighbor pixels [27].

Core and its circular neighbor pixels are utilized like in SUSAN corner detector [28]. Given a candidate of corner on the center of an image patch, not entire region is evaluated, only discretized circle depicting segment is calculated. A set of n adjacent pixels around the core might be in two conditions. It can be all brighter or all darker than the intensity of core pixel I_p plus the threshold t. There are three conditions until one can determine whether p is a corner or not as in (2.11).

$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \leq I_p - t \quad \text{(darker)} \\ s\,, & I_p - t < I_{p\to x} < I_p + t \quad \text{(similar)} \\ b, & I_p + t \leq I_{p\to x} \quad \text{(brighter)} \end{cases} \tag{2.11}$$

In deciding which pixel to be compared first, this method employed ID3 decision tree gaining large number of stable features. However using this tree to perform the search did not guarantee that entire configuration is found. Only a small rotation of camera yields disturbance in distribution of pixel configurations extremely.

2.1.5. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test (AGAST)

The success of FAST in locating the features with very fast computation stimulated Mair et al to optimize the original method. Their method is called AGAST is optimization of FAST [29]. The detector also based on SUSAN corner detector that assessing circular pixels around the core pixel. The optimization is built by constructing binary decision tree that not just generic but adaptable to new environment thus there will not be necessity to perform the search from the scratch each time the environment amended. The look of the tree is described in Figure 2.7. Combination of two trees allow the corner detector to adjust into environment automatically and conducts very expeditious decision tree for the image patch and only has one pixel delay.

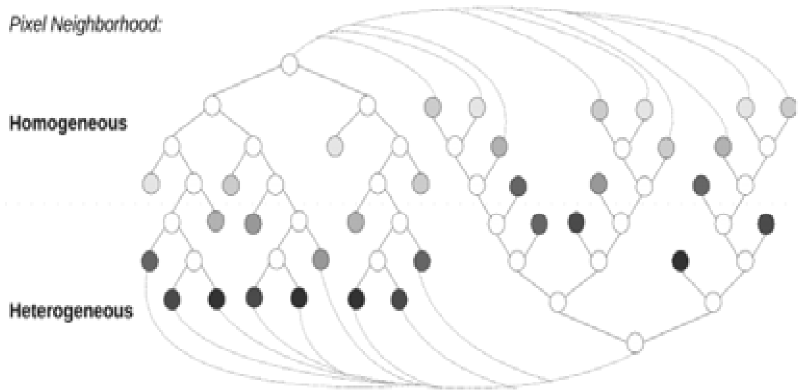


Figure 2.7: Decision tree scheme to determine pixel configurations of AGAST [29].

In AGAST, the conditions for corner determination use more detail specifications. There are six conditions as shown in (2.12) that includes five physical appearances darker which are darker, not darker, similar, not brighter and brighter.

$$S = \begin{cases} d, I_{n\rightarrow x} < I_n - t & \text{(darker)} \\ d_{not}, I_{n\rightarrow x} \nless I_n - t \wedge S'_{n\rightarrow x} = u & \text{(not darker)} \\ s, I_{n\rightarrow x} \nless I_n - t \wedge S'_{n\rightarrow x} = b_{not} & \text{(similar)} \\ s, I_{n\rightarrow x} \ngtr I_n + t \wedge S'_{n\rightarrow x} = d_{not} & \text{(similar)} \\ b, I_{n\rightarrow x} \ngtr I_n + t \wedge S'_{n\rightarrow x} = u & \text{(not brighter)} \\ b_{not}, \quad I_{n\rightarrow x} > I_n + t & \text{(brighter)} \end{cases} \quad (2.12)$$

S denotes the state of each pixel location while $S'_{n\rightarrow x}$ express the prior state. *I* denotes the intensity brightness in pixel and *u* express the unknown state.

2.2 Feature Descriptor

2.2.1 Scale Invariant Feature Transform (SIFT)

SIFT is scale and rotation invariant feature descriptor using Difference of Gaussian (DoG) to detect features, SIFT is one of the best among local features in literature today regarding repeatability rate produced by using this method. In order to achieve rotation-invariant property, orientation assignment is computed. Before building the descriptor, orientation assignment is computed provides rotation-invariance.

The gradient and magnitude around candidate features are calculated then distribution of the gradients are placed in histogram bins. Consider each bin has range

10 degrees as represented in Figure 2.8. The most prominent gradient orientation(s) are identified. If there is only one peak, it is assigned to the feature. If there are multiple peaks above the 80% mark, they are all converted into a new feature (with their respective orientations). In the figure, the highest bins are the most prominent orientation then the orientation in that feature is two way.

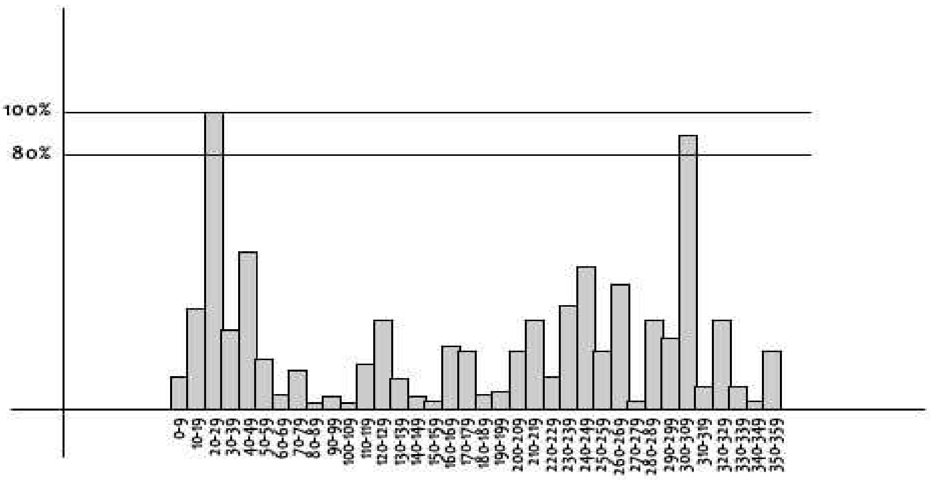


Figure 2.8: SIFT orientation assignment using histogram [26].

The last step is generating features. In order to construct such feature that distinctive but also lenient, a 16×16 window around the feature is divided into sixteen 4×4 windows. Orientations and gradient magnitudes within each 4×4 window, gradient magnitudes and are calculated. The orientations were placed into an 8 bins histogram. Gradient orientation in the range 0-44 degrees will be stored to the first bin. Range 45-89 add to the next bin. The amount added to the bin depends on the magnitude of the gradient. Eventually we have features with 4x4x8 = 128 dimensions as appeared in Figure 2.9, then normalize it.

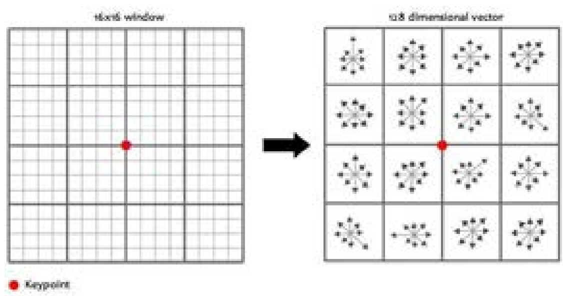


Figure 2.9: SIFT 128 dimensions highly distinctive descriptor [26].

2.2.2 Speed Up Robust Features (SURF)

SURF is a scale and rotation invariant feature descriptor. It could be categorized under the family tree of the mostly used SIFT feature. These SIFT like features are commonly used in various applications such as object recognition, image retrieval and image stitching since the last decade. SURF uses integral image to speed up the computation.

Viola and Jones have proposed to use integral images for face detection in [30], this method allows for rapid computation of Haarwavelets or any box-type convolution filter as shown on Figure 2.10. Integral image is employed to quickly approximate Hessian matrix.

$L_{xx}(x,y,\sigma)$ is the Laplacian of Gaussian of the image. It is the convolution of the Gaussian second order derivative with the image. Lindeberg has [12] shown that Gaussian function is optimal for scale-space analysis. This method approximates determinant D_{xx} to get L_{xx}.

$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \tag{2.13}$$

$$det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (\omega D_{xy})^2 \tag{2.14}$$

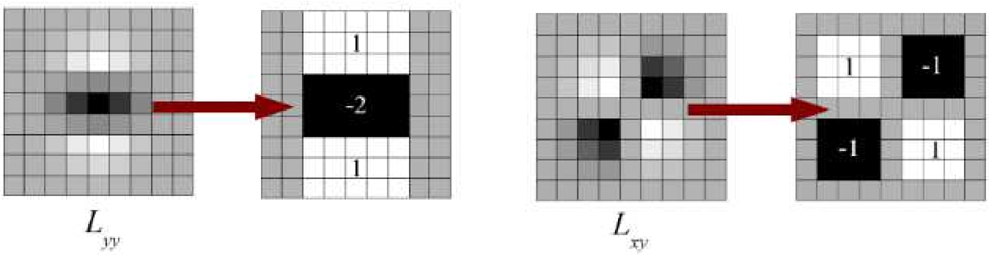


Figure 2.10: Approximated second order derivatives with box filters [31].

Scale analysis is done with constant image size so no need to generate image pyramid. A major orientation is calculated when a point is considered a keypoint. The second step is to construct the scale invariant descriptor on each keypoints detected. Integral Image (summed area tables) is an intermediate representation for the image and contains the sum of gray scale pixel values of image as represented in (2.15) and illustrated on Figure 2.11.

$$s = A - B - C + D \tag{2.15}$$

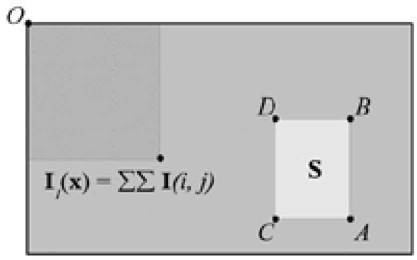


Figure 2.11: Integral image.

In order to achieve rotation invariant, this method straighten a rectangle to the major orientation. The size of the rectangle is proportional to the scale where the interest point is detected. The rectangle is then cropped into a 4 by 4 grid. Different information such as gradient or absolute value of gradient are then subtracted from each of these sub square and composed into the interest point descriptor.

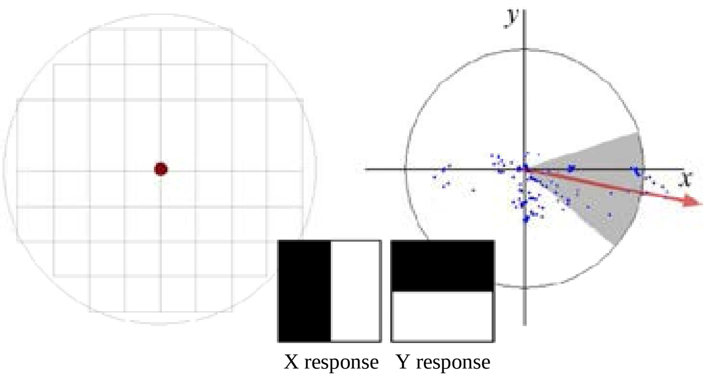


Figure 2.12: SURF Orientation Assignment

The SURF feature is a speed up version of SIFT, which uses an approximated DoG and the integral image trick. The integral image method is very similar to the method used in the famous Viola and Jones' adaboost face detector. An integral image is simply an image which its each pixel value is the sum of all the original pixel values left and above it. The advantage of integral image is that after an image is computed into an integral image, it can compute block subtraction between any two blocks with just six calculations. With this advantage, finding SURF features could be several order faster than the traditional SIFT features.

2.2.2 Binary Robust Independent Elementary Features (BRIEF)

A simple comparison of intensity in image patches can be utilized as efficient descriptor. Given test τ on image patch q as shown in Figure 2.13 (region inside yellow circle) with M x N

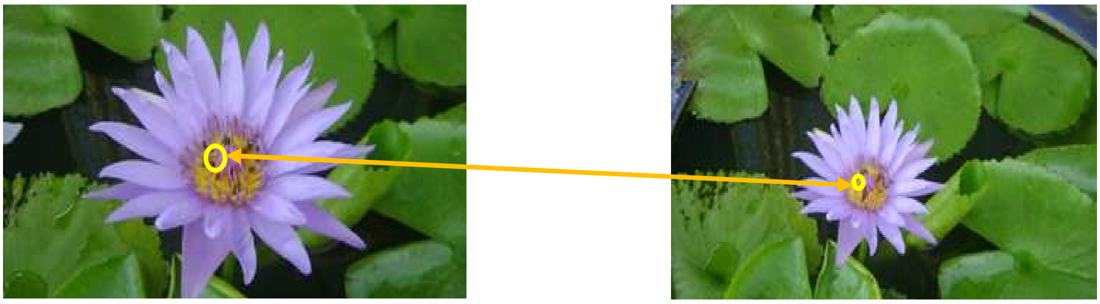


Figure 2.13: Intensity of patches in images can be employed as features.

$$\tau(q;\ x, y) = \begin{cases} 1 & if\ q(x) < q(y) \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

$q(x)$ denotes the intensity of pixel in a smoothed version of q at $x = (u, v)^T$. The way to select a set of $n_d(x, y)$-location pairs uniquely defines a set of binary tests. BRIEF descriptor has n_d-dimensional bitstring. In their work they used n_d = 128, 256, and 512. The size of n_d sets the computational time, storage and repeatability.

$$f_{n_d(q)} := \sum_{1 \le i \le n_d} 2^{i-1} \tau(q; x_i, y_j) \tag{2.17}$$

Taking information at any pixel directly yields noise sensitivity. In order to avoid the noises, smoothing the patches using some kernel should be applied prior. The smoothing allows the descriptor stability and repeatability enhanced. They found a 9 × 9 pixels Gaussian kernel will be sufficient.

Generating descriptor n_d brought us the question how to select the location for the test. There are five ways to select the pattern of image patch with assumption that the core of the patch is right at the center of the patch.

1) $(X,Y) \sim i.i.d.$ Uniform $\left(-\frac{S}{2},\frac{S}{2}\right)$: The (x,y) locations are distributed equally all over the patch and tests can be loated near to patch boundary.
2) $(X,Y) \sim i.i.d.$ Gaussian $\left(0,\frac{1}{25}s^2\right)$: The tests are sampled from an isotropic Gaussian distribution.
3) $X \sim i.i.d.$ Gaussian $\left(0,\frac{1}{25}s^2\right)$ $Y \sim i.i.d.$ Gaussian$\left(0,\frac{1}{100}s^2\right)$: The sampling requires two steps. It forces the tests to be more sectional. Test locations outside the patch are clamped to the patch edge.
4) $(x_i y_i)$ are random sample from discrete locations of a coarse polar grid introducing a spatial quantization.
5) $\forall i\ : x_i = (0,0)^T$and $y_i$is takes all possible values on a coarse polar grid containing n_d points.

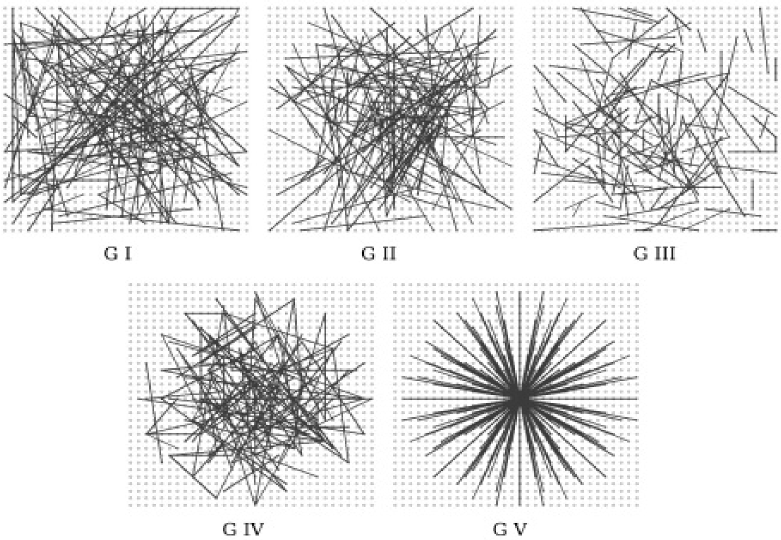


Figure 2.14: Four random sampling approaches to select test locations except the GV [32].

There is no orientation assignment in this method thus this method can only tolerate little change of rotation.

2.2.3 Binary Robust Invariant Scalable Keypoints (BRISK)

A new robust feature extraction method called Binary Robust Invariant Scalable Keypoints [33] was introduces in 2011. Original design of BRISK descriptor method made it flexible to be coupled to any feature detector and vice versa. This method has focus on efficiency of computation and was triggered by the knowledge that in detecting regions of interest in the image based on accelerated segment test called AGAST (Adaptive and Generic Accelerated Segment Test). This method looks for the features not only on scale-space images but also in plane between. See the downsampling of original image to its half-size in Figure 2.15. In BRISK searching for features not only on images a, b, and c but also in image plane between a – b and b-c.

The keypoints of BRISK is its utilization of sampling pattern around the feature point. Once the features are located, the gradient of point sampling pair is generated by (2.18).

$$g(p_i, p_j) = (p_j - p_i).\frac{I(p_j, \sigma_j) - I(p_i, \sigma_j)}{\left\| p_j - p_i \right\|} \tag{2.18}$$

where $g(p_i, p_j)$ is local gradient, $I(p_j, \sigma_j)$ and $I(p_i, \sigma_j)$ are smoothed intensity values, and (p_i, p_j) is a sampling-point pairs. Considering the set A of all sampling-point pairs:

$$\mathrm{A} = \left\{ (p_i, p_j) \in \Re^2 x \Re^2 \middle| i < N \wedge j < i \wedge i, j \in N \right\} \tag{2.19}$$

$$\begin{aligned} S &= \left\{ (p_i, p_j) \in A \middle| \ \left\| p_j - p_i \right\| < \partial_{\max} \right\} \subseteq A \\ L &= \left\{ (p_i, p_j) \in A \middle| \ \left\| p_j - p_i \right\| > \partial_{\min} \right\} \subseteq A \end{aligned} \tag{2.20}$$

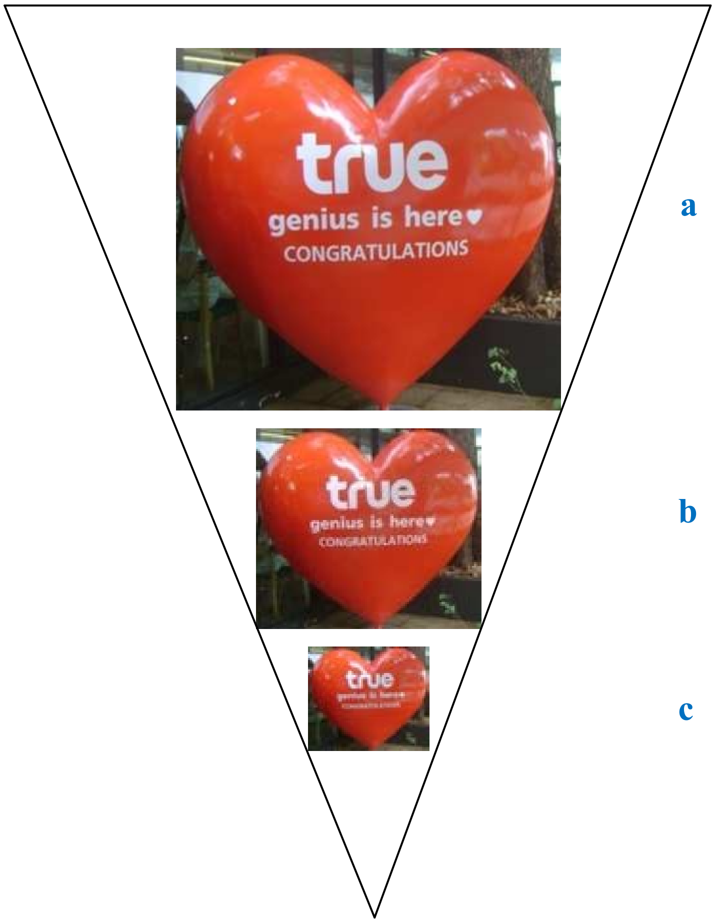


Figure 2.15: Generating image pyramid by downsampling the original image.

The threshold distances are set to $\partial_{\max} = 9.75t$ and $\partial_{\min} = 13.67t$ which t is scale of *k*. Iteration of pattern direction of the keypoint *k* to be:

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(p_i, p_j) \in L} g(p_i, p_j) \tag{2.21}$$

The final step is building the descriptor by constructing binary strings using (2.22).

$$b = \begin{cases} 1, & I(p_j^{\alpha}, \sigma_j) > I(p_i^{\alpha}, \sigma_i) \\ 0, & otherwise \end{cases} \tag{2.22}$$

$$\forall (p_i^{\alpha}, p_j^{\alpha}) \in S$$

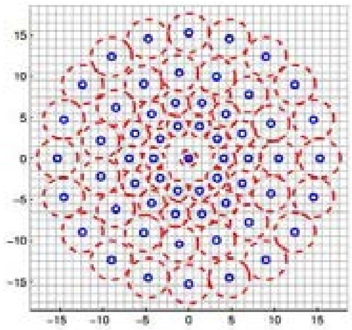


Figure 2.16: The BRISK sampling pattern visualization [32].

2.2.4 Fast Retina Keypoint (FREAK)

The last two descriptors in this chapter had claimed that simple intensity comparison enables is sufficient to be feature descriptor. FREAK did the pattern of feature even farther by using human retina-like pattern. Neuroscience development provides the knowledge in understanding the visual system and how brain perceives the information that are transmitted from an image. This method proposed to imitate the same strategy to design the feature descriptor. The analogy of human retina to computer vision is represented in Figure 2.13.

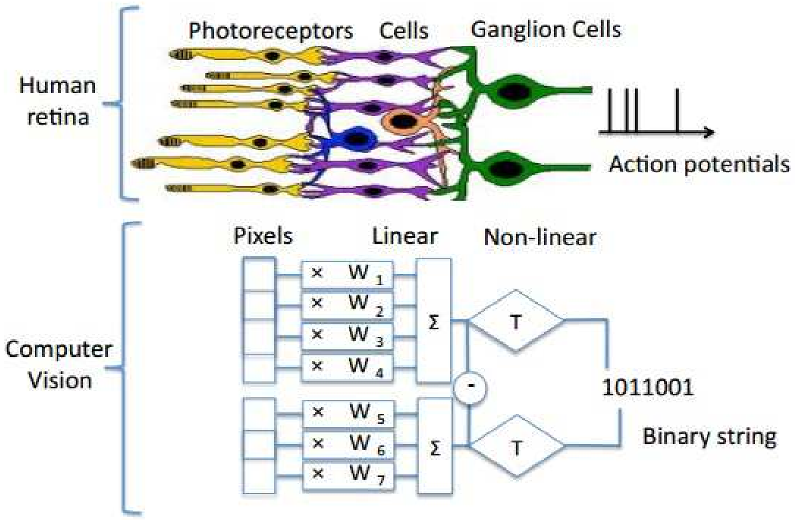


Figure 2.17: Analogy of human retina to computer vision [34].

The F descriptor is constructed by thresholding the difference between pairs of receptive fields with their corresponding Gaussian kernel. F descriptor is a binary string formed by a sequence of one-bit Difference of Gaussians (DoG). F is computed using (2.23) where P_a denotes the pair of receptive fields and n is descriptor size desired.

$$F = \sum_{0 \le a < n} 2^a T(P_a) \tag{2.23}$$

$$T = \begin{cases} 1 & if \ \left(I(P_a^{r_1}) - I(P_a^{r_2})\right) > 0, \\ 0 & otherwise, \end{cases} \tag{2.24}$$

$I(P_a^{r_1})$ denotes intensity of the first receptive field of the pair Pa smoothed by Gaussian. With such amount of receptive fields led to a huge size descriptor. This huge size of descriptor contains of the possibility of unnecessary pairs. In order to avoid such disadvantage, the selection algorithm similar to ORB [35] is chosen.

Human vision is not static. Eyes always move with personal movements named saccades. Cells topology in retina is the reason for that saccades. Fovea captures high-resolution information produced by the high density photoreceptors. Consequently, it provides an important part in recognizing and matching objects. On the other side, perifoveal sensates less specific information. These two characteristics of retina cells are optimally be used to construct such sampling pattern.

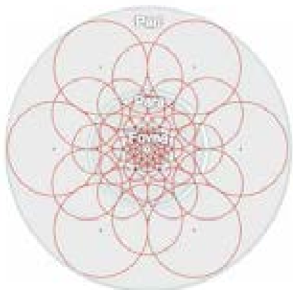


Figure 2.18: Sampling pattern of FREAK similar to the retinal ganglion cells pattern [34].

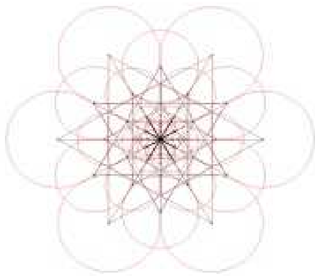


Figure 2.19: Pair selection in order to compute the orientation [34].

2.3 Descriptor Matching

Talking about matching between two descriptor sets, once local feature descriptors are matched, distances between descriptors are calculated using certain distance formula. Euclidean Distance and Hamming distance are two among popular ones.

2.3.1 Euclidean Distance

Given two feature points lying on $a = (x_1, y_1)$ dan $b = (x_2, y_2)$, the Euclidean distance is calculated using (2.25) below.

$$d = \sqrt{(x_2 - x_1)^2 - (y_{2-}y_1)^2} \tag{2.25}$$

In general, Euclidean distance is represented in (2.15) as follows:

$$d = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{2.26}$$

2.3.2 Hamming Distance

The difference between two binary strings can be calculated using Hamming Distance. The Hamming distance can only be calculated if the two strings have equal length. There are three simple steps to calculate Hamming distance that are:

1. Compare the first bit in both strings as described with red curve arrow on table below. If they have the same value then give “0” otherwise give “1”
2. Compare each bit pair successively
3. Sum all the “0” and “0” in together to obtain the Hamming distance.

Simple example:

String one: 001000101110

String two: 010011010011

String one	0	0	1	0	0	0	1	0	1	1	1	0
String two	0	1	0	0	1	1	0	1	0	0	1	0
Comparison	0	1	1	0	1	1	1	1	1	1	0	0

$$\text{Hamming distance} = 0+1+1+0+1+1+1+1+1+1+0+0$$

$$= 8$$

Hamming distance between two mentioned strings is 8.

CHAPTER III

PERFORMANCE EVALUATION FRAMEWORK

One object can have various appearances on image scenes as appear in Figure 3.1. Depends on how the picture is taken, the equipment, the environment and so on. Various point of view, scale size, lighting and orientation might happen. In this thesis, we measure the behavior of local feature descriptor due to transformed testing images.

Figure 3.1: Various appearance of image with one main object.

3.1. Parameters

The work in this research followed the protocol proposed by Mikolajczyk et al [36]. In their work, transformed images as the testing dataset are compared to original images as training dataset. Various transformations includes scale size, blur degree, changed in viewpoints, lighting diversities, JPEG compression were employed to amend the ideal testing images into approximately real world images.

In this thesis, the evaluation measurements are based on a number of correct and incorrect matches given by an image pair. Computational time also included as measurement of effectiveness of the methods. The repeatability, precision and recognition rate are chosen as parameters. The definitions of the terms are as follows:

1) **Repeatability**

   Given a pair of image descriptor, repeatability is the number of correspondences occurred between two image descriptor divided by total number of features in a query image. Repeatability is the most important parameter for feature extraction. Once the repeatability value is known, we can calculate the precision and recognition rate simultaneously. After being transformed, there will appear some of unnecessarily additional features on testing image, this feature will not be counted because only patches that exist in both training and testing images are included. Repeatability is computed using (3.1) as shown below

$$Repeatability = \frac{number\ of\ matches}{number\ of\ features\ detected} \cdot 100\% \tag{3.1}$$

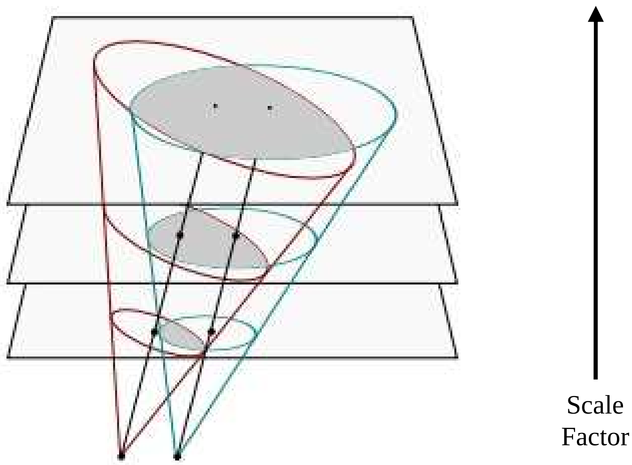


Figure 3.2: Ovelapping ellipses on two features [37].

Each time the features from two images are being compared, there are two types of ellipses in few sizes and point of view. Exhaustive search is applied to locate overlapping ellipses. In Mikolajczyk's original protocol, 40% overlap error was permitted. The ellipses sizes effect the results. The bigger the ellipses are the narrower is the overlap error. Therefore, all the ellipses should previously be normalized into 30 pixels radius before overlap error is computed. The influence of rescaling the ellipses for overlap measurement is described in Figure 3.2. The influence of increasing relative size in overlapping gray area is obviously visible.

2) Precision

Precision is the comparison between the total number of correct feature matches (pass reprojection error threshold) and the total number of correspondences occurs by a given image pair. Precision is simply represented in (3.2) as follows:

$$Precision = \frac{number\ of\ correct\ matches}{number\ of\ matches} \cdot 100\% \tag{3.2}$$

3) Recognition Rate

Recognition rate is the success identification recorded. Given a query image, the success identification happens when the features in a query image passes homography threshold and found its match in training image gallery. Recognition rate is shown in (3.3) as follows:

$$Recognition\ Rate = \frac{number\ of\ correct\ identifications}{number\ of\ identifications} \tag{3.3}$$

4) Computational Time

Time elapses for feature detection, feature description and descriptors matching are recorded and presented in milliseconds.

3.2 Datasets

In order to provide satisfying properties of various transformations and of testing image, we use several databases which properties are explained in detail below.

3.2.1 Carnegie Mellon University PIE Database

CMU-PIE is database under Robotics Institute, Carnegie Mellon University. The dataset contains of 68 subjects with total 41,368 images having 13 different poses for each subject, 4 different facial expressions and 43 different illumination conditions. Figure 3.1 illustrates the diversity in database. The file is color image in JPEG format with original size 640 x 486 pixels. Example images of CMU-PIE Face Database is shown on Figure 3.3.

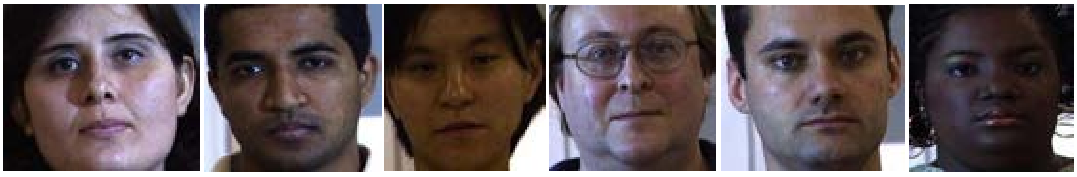


Figure 3.3: Sample of images in CMU-PIE Database.

3.2.2 The Sheffield Database

The Sheffield (prior known as UMIST Database) Face Database [38] has 20 subjects with total 564 images possessing various gender, race, and look. A wide range of poses is captured for each subject. The images are saved in PGM format, 256-bit grey-scale, having size of 220 x 220 pixels. Example of the various poses in database is shown on Figure 3.4.

Figure 3.4: Sample of poses in Sheffield Database.

3.2.3 Surveillance Camera Face Database

SCface database [39] is a face image database made by Prof. Mislav Grgic and his team from Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. The dataset contains of 130 Subjects having total 4160 images. The images captured in uncontrolled room. Five surveillance cameras in various qualities and brands are installed. One high resolution camera is employed also. Images taken from various types of camera resemble the real-world condition. The images are taken in both visible and infrared). The images are color image with various pixels size. Example images of CMU-PIE Face Database is shown on Figure 3.5.

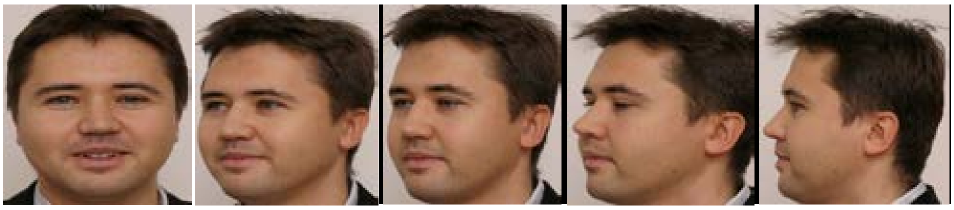


Figure 3.5: Sample of poses in SCface Database.

3.3 Implementation Properties and Simulation Process

Implementation was built by using Visual Studio C++ with dependency to OpenCV library. We used notebook with core i5 processor, RAM 4 GB and Windows 7 operating system.

There are four image transformations that we process inside the simulation that are scale, rotation, blur and brightness. From original image, we resize the scale from 0.1-0.9 with increment value is 0.1 subsequently. Rotation is from -30^0 to $+30^0$ with increment value is 5 degrees for each step. Blurring the original image, we employ Gaussian Blur with kernel size 1-9. Brightness change is provided by changing the brightness constants value from -100 to 100. Pose variations were provided by the raw database thus we process that without any transformation.

There are three main processes in the simulation. First is training and second is testing, the last one is matching between training descriptors and testing descriptors.

3.3.1 Training

On training, we select 20 original up-frontal images with 240 x 320 pixels size. Each training image will pass training process in order to get descriptors extracted for mentioned methods in Chapter II. The steps on training process are described on Figure 3.6.

3.3.2 Testing

Testing part has number of images vary depends on the number of argument in transformations. Mostly we will use at least 100 testing images for every test. We have two kinds of testing. Testing with transformed query images and testing without transformation. For testing the scale, orientation and blur, we generated degraded images from up-frontal original image. For pose changes test, we did not apply any transformation, since the raw data of face images were already has various wide range of pose. The process of testing is represented in flow-chart on Figure 3.6.

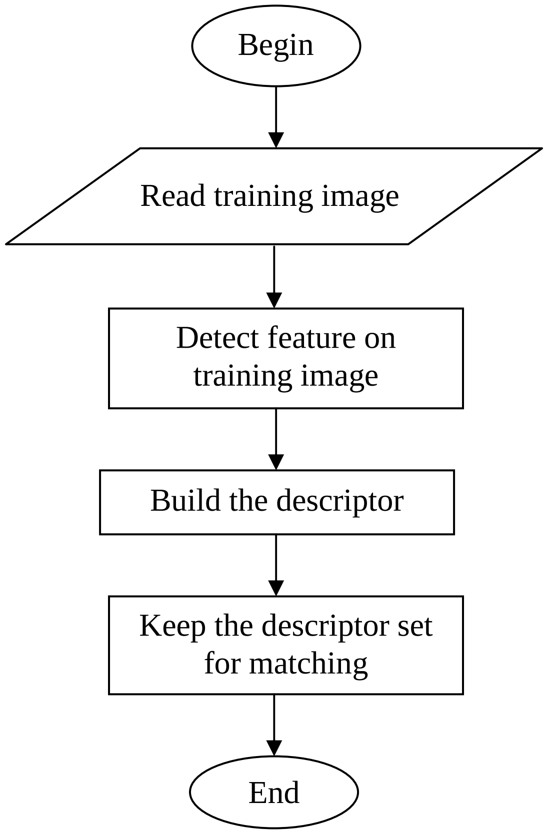


Figure 3.6: Flow-chart of training process.

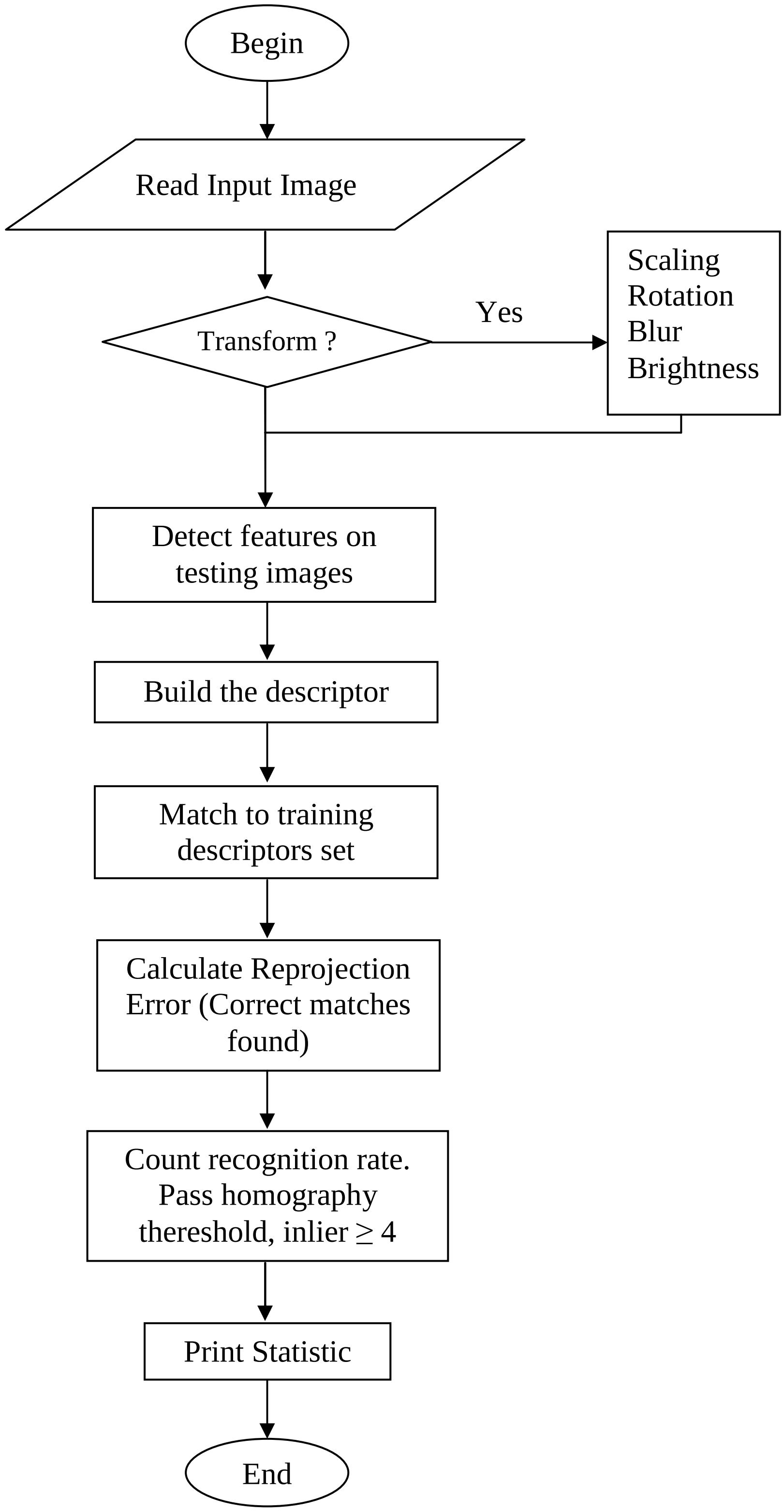


Figure 3.7: Flow-chart of testing process.

3.3.3 Matching

A transformed testing image is compared to original images of 20 subjects in training dataset. Homography between given images compared are calculated before. Extracted features of each testing images are projected and compared to the features of training images. Matching was done by using k-nearest neighbor (kNN) [38] matching to classify the closest training examples to determine the existence of correspondence. In order to eliminate the outlier we use curve fitting called Random Sample Consensus (RANSAC) [40].

There is one main difference objective of this work than in the original work of The Mikolajczyk et al [36]. Their objective was mainly to determine the highest performance method due to view-points change testing images. In this thesis, we would like to find the behavior of each local descriptor for face identification benefit. This means, the object in images are human face(s) so that matching search occurs between two objects in same class.

CHAPTER IV

EXPERIMENTAL RESULT AND DISCUSSIONS

1.1. Image Transformation

Four types of image transformation, scale, in-plane rotation, blur, brightness change were applied in this research. The behavior of feature extractions method due to changes of each transformation is represented consecutively. We present the qualitative result in graphs and tables. The rest we elaborate the reasons behind such characteristic on each feature extraction method one by one.

1.1.1. Scale

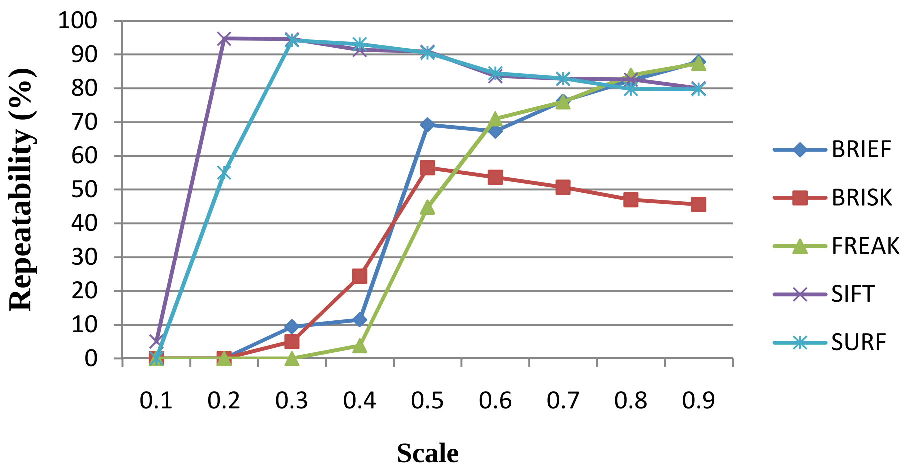


Figure 4.1: Repeatability comparison of feature extraction methods due to changing of scaling size.

Various image scales with value 0.1-0.9 were applied to transform original image into degraded images that will be used as testing images. CMU-PIE dataset is used in this test. The result shows in Figure 4.1 is expected for the SIFT to perform very satisfying among other local descriptors. Four octaves scale space images reproduced in early stage of feature detection grants it scale-invariance to image resizing. SURF almost achieves the same performance as SIFT. However in SURF there is no image pyramid produced prior SURF descriptor construction. The intriguing result comes from BRISK unexpectedly. Unique pattern and thorough image sampling pairs did not indicate significant result.

BRIEF and FREAK has almost the same quality. All local descriptors are gaining more repeatability and precision as the scale climbing up. SIFT and SURF are acceptable to overpower another methods since these methods have 128 and 64 dimensional feature vector therefore has its inner-method power to build such distinctive features. The computational time for transformed query image test is described in Table 1.

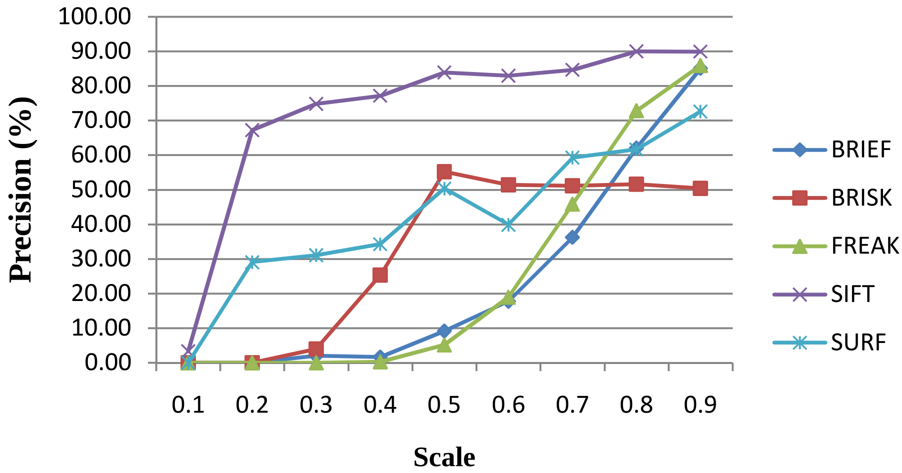


Figure 4.2: Total number of correct matches of feature extraction methods due to changing of scaling size.

Precision result as plotted in Figure 4.2 mostly follows result appeared on Repeatability graph on Figure 4.1. However, SURF indicates a different behavior. The precision drops to be near as BRISK which we can look back for reprojection error criterion. If the re-projection error is less or equal to 2 then correspondences occurred will be classified as correct matches which related to calculation of precision as stated in (3.2).

1.1.2. In-plane Rotation

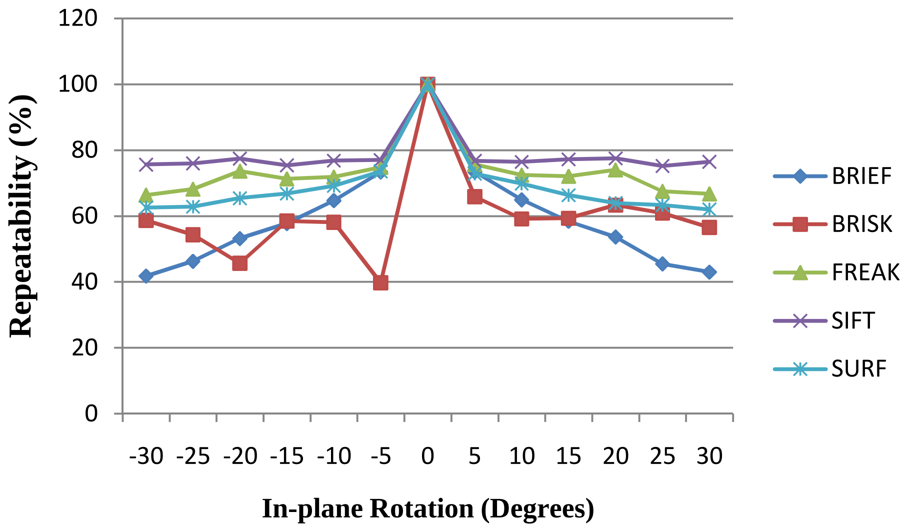


Figure 4.3: Repeatability comparison of feature extraction methods due to changing of orientation.

Result for in-plane rotation changes as shown in Figure 4.3 is very much as predicted. We used CMU-PIE dataset for this test. The closer to the zero degree rotation, the higher repeatability yielded. Here we only use rotation range between -30 to 30 degree in consideration that human face movement is most likely happen between that range.

In some scenario of face recognition, scale-invariance is far more important than accurately orientation calculation. Therefore we do not use both scale and rotation invariant features for that case. BRIEF provides local descriptor that depends on a relatively little number of intensity difference tests to represent an image patch as a binary string. Not only is the descriptor construction and descriptor matching for this descriptor much faster than other state of the art ones, it also inclines to result higher recognition rates, as far as the invariance to wide-range in-plane rotations is not a necessity. It is an important result from a practical point of view because it means that real-time matching performance can be achieved even on devices with very limited computational power. The importance is also can be seen from more theoretical point of view because it ensures the validity of the latest trend, migration from the use of Euclidean to the Hamming distance for matching process.

SURF also performed well. It is a scale-invariant feature detector based on the Hessian-matrix. However, rather than using a different measure for selecting the location and the scale as in SIFT, the determinant of the Hessian is used for both. The Hessian matrix is roughly approximated, using a set of box-type filters, and there is no smoothing applied when going from one scale to the next and no down-sampling as well.

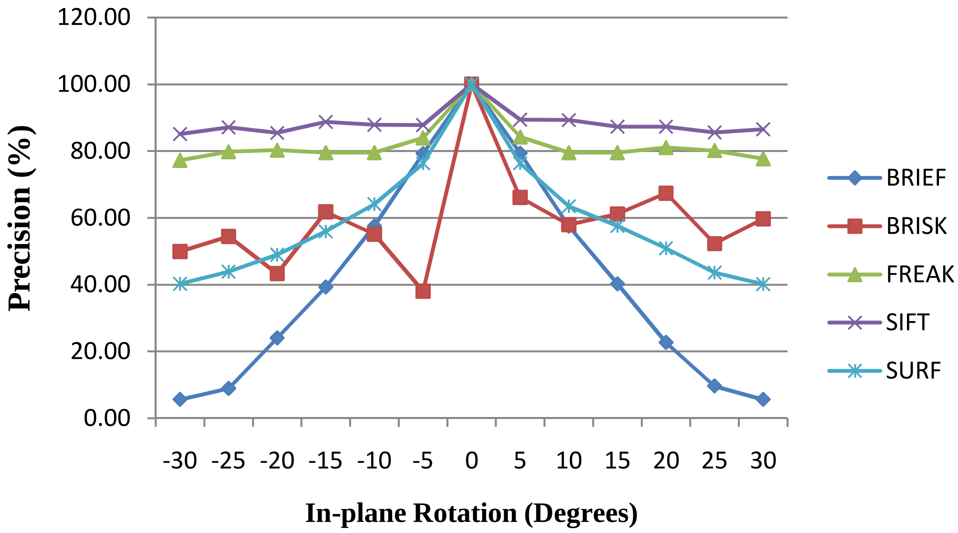


Figure 4.4: Total number of correct matches of feature extraction methods due to changing of orientation.

Similar result is produced for precision measurement. The closer in-pane rotation to zero (almost no tilting on face image) the better precision rate is achieved.

1.1.3 Blur

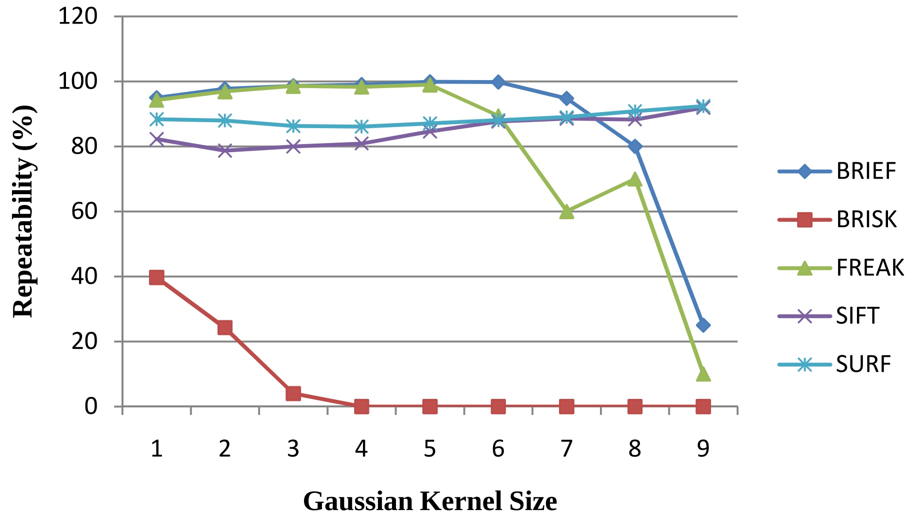


Figure 4.5: Repeatability comparison of feature extraction methods due to changing of kernel size.

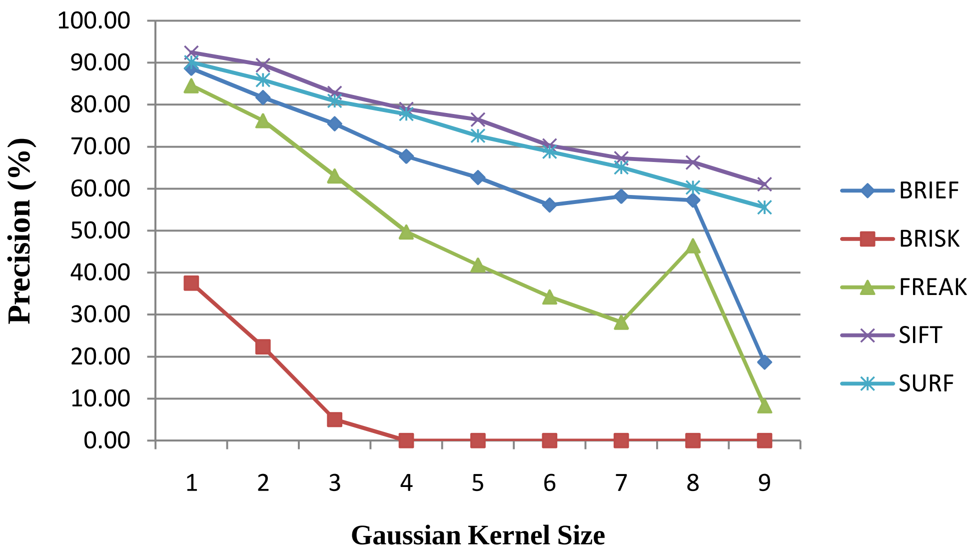


Figure 4.6: Total number of correct matches of feature extraction methods due to changing of kernel size.

Next, we test using blurred images. Taking Gaussian kernel with size 1-9, the decreasing both of repeatability rate and precision rate follows the rising of kernel. The more blur kernel size is applied the more blurred testing images we generated yields less both correspondences and percent of correct matches is obtained. We used the CMU-PIE dataset for this test.

Table 4.1 represented that FAST detector that used with both BRIEF and FREAK descriptor produced the largest number of features. The FAST detector was inspired by SUSAN detector [28]. SUSAN computes the fraction of pixels within a neighborhood which have similar intensity to the center pixel. This idea is taken further by FAST, which compares pixels only on a circle of particular radius around the point. The test criterion operates by considering a circle of 16 pixels around the corner candidate.

Inspecting Table 4.1, we will discover that SURF has longer computational time in comparison to SIFT. It is weird since SURF descriptor dimension is a half size of SIFT 128 dimensional features. This fact can be explained only with the increasing number of features. Since SURF detected lots of number more features than SIFT. It is acceptable that SURF needed more time to describe the features. Hence we know that the description time will be increase linearly to the rising number of features generated.

1.1.4 Brightness Change

One more time we used CMU-PIE dataset to test various brightness to know repeatability of features. Brightness adjustment of an image plane is one of the easiest image processing operations. The only thing needed is adding the desired change in brightness to each of the red, green and blue (RGB) colour components. Generally the degree of brightness will be in range of -255 to +255 (24 bit palette). The more negative the degree, the darker image is produced and contrary the more positive the degree, the brighter image is resulted. No exceptional outcomes from this test. The result goes with the nature that the closer brightness value into normal (zero) the better repeatability and precision are gained. However such asymmetric result appears if we take a look carefully on Figure 4.7 and Figure 4.8. Reduction in left side of the curve is greater than what occurs in right side. Therefore, we can conclude that for all local descriptor are more sensitive in darkness.

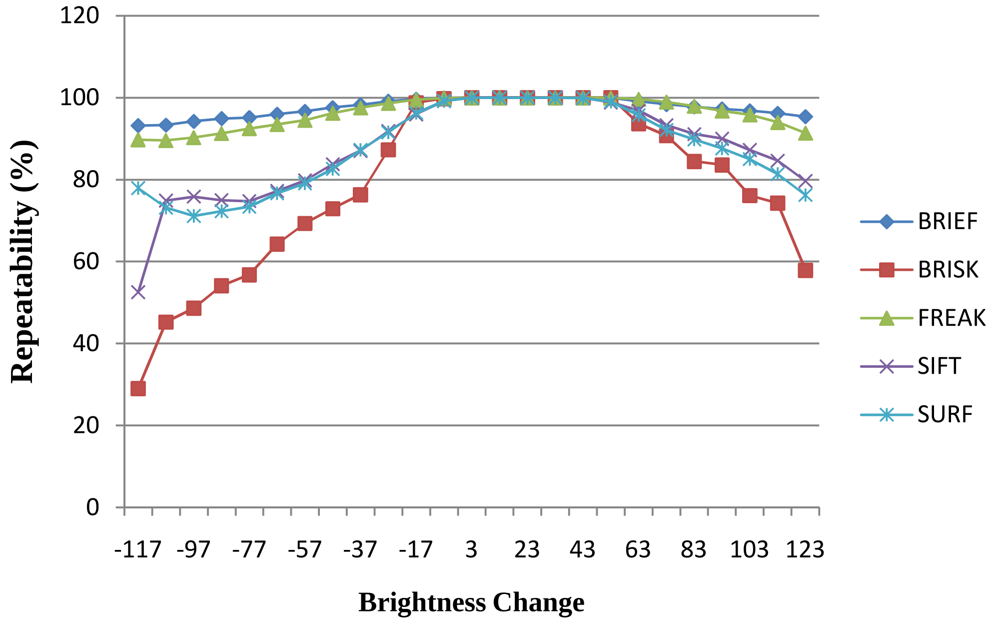


Figure 4.7: Repeatability comparison of feature extraction methods due to changing of brightness.

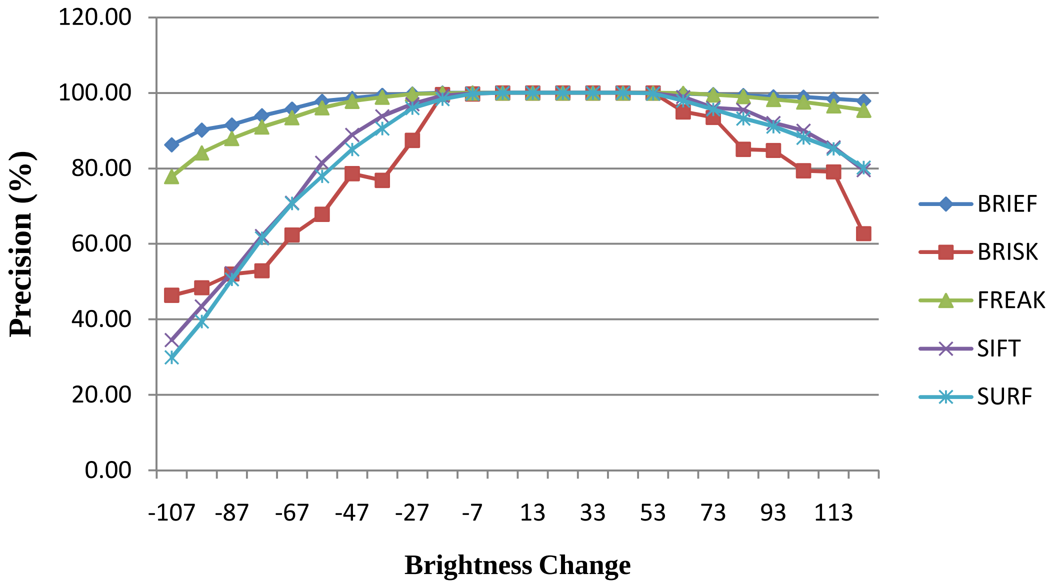


Figure 4.8: Total number of correct matches of feature extraction methods due to changing of brightness.

All methods work sufficiently satisfying in this case. The major reason is that all descriptors extracted have been normalized. This normalization makes descriptor invariant to brightness changes.

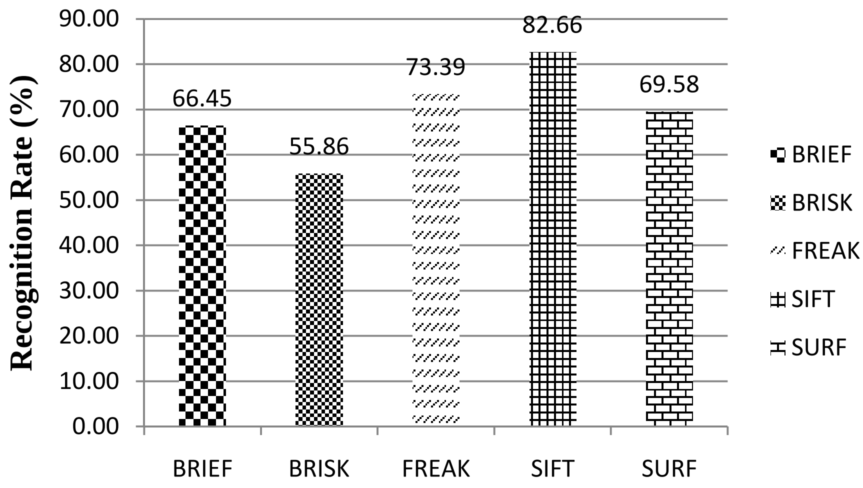


Figure 4.9: Recognition rate comparison of feature extraction methods with transformed query images.

Recognition result for all transformed (scale, in-plane rotation, blur, brightness change) testing images is calculated. The result shows that SIFT gained the best performance followed by FREAK, SURF, BRIEF and BRISK. However, SIFT also possess the second longest computational time as shown in Table 1. Thus SIFT is not suitable for real-time face recognition application which sets the speed efficiency as the main goal.

Table 4.1: Computational time of feature extraction methods in milliseconds (ms) with transformed query images.

Methods	Detection	Description	Matching	Total
BRIEF	49.31	1047.71	415.98	1513.00
BRISK	188.46	19.84	6.85	215.15
FREAK	49.31	344.43	839.36	1233.1
SIFT	2178.73	3302.68	78.99	5560.41
SURF	2009.27	6459.40	481.66	8950.33

Table 4.2: Number of features and number of correspondences feature extraction methods with transformed query images.

Methods	#features	#Correspondences
BRIEF	1570	1239
BRISK	46	25
FREAK	1570	1249
SIFT	387	320
SURF	841	679

4.2 Pose Change

Pose change test used difference dataset than the other four tests. We used both Sheffield database and SCface Database. Technical problem made us to randomly select 100 images to conduct this test. The result is unexpectedly very interesting. BRISK that claimed [33] to have a reasonable performance, failed for various pose face image recognition while SIFT that famously reliable on accuracy performed just just fairly well. SURF also shows downfall in performance for this test. The first thing we need to keep in mind is that during the test, no image pre-processing is applied thus the testing image comes as raw as original image which contains of pixels intensity information only to be used (image is converted into gray-scale inside the algorithm of feature description, no color cue is taken into account).

The extraction times and quantities of features and descriptors are compared in this section. All results are computed on a set of 100 images. Table 4.3 shows the averaged results. The largest number of features are extracted by FAST detector that we used together with BRIEF and FREAK descriptor. The variation in the number of features is expected, since the various detectors respond to different types of image structures. This can be controlled to a small extent by parameter settings but the order of numbers remains the same. The most efficient detector is FAST which is faster than SURF and also faster than DoG.

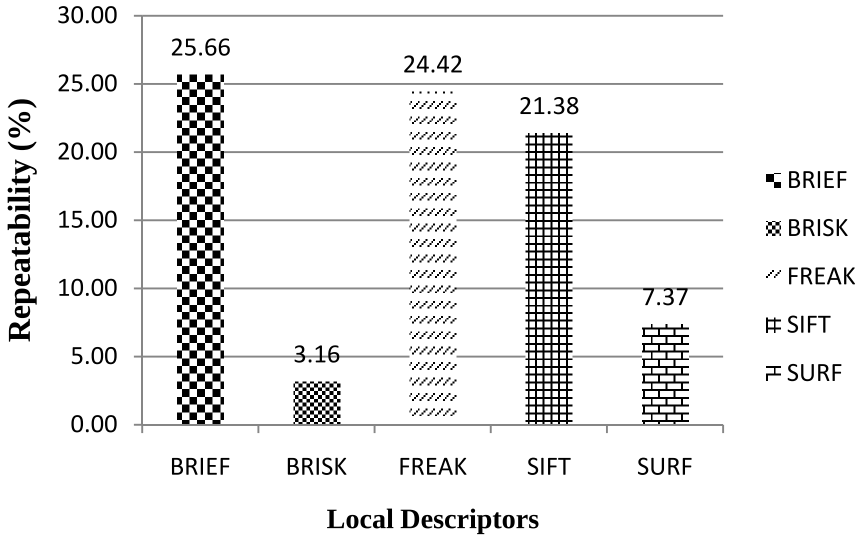


Figure 4.10: Repeatability comparison of feature extraction methods due to changing of pose.

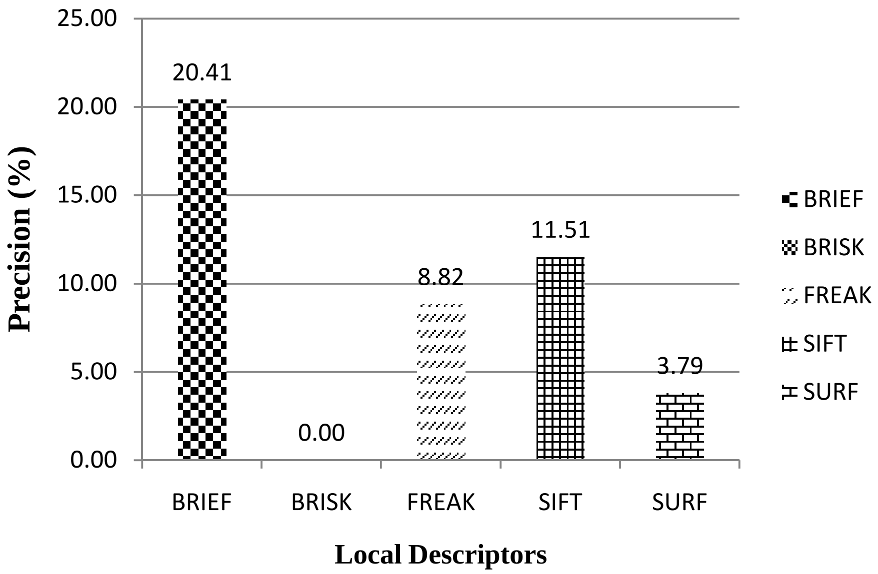


Figure 4.11: Total number of correct matches of feature extraction methods due to changing of pose.

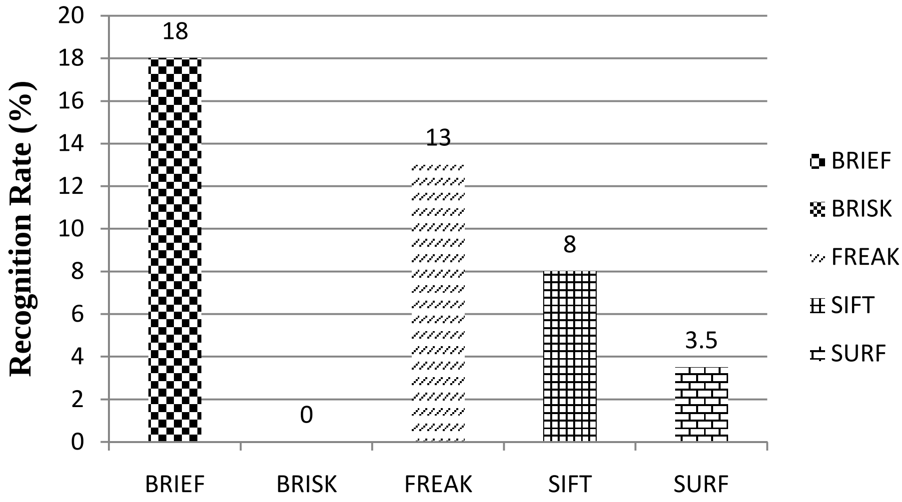


Figure 4.12: Recognition rate of identification with various pose in query images.

The zero result gained by BRISK brought us so many questions. The feature detection was done using AGAST which is the optimization of FAST in its decision tree also the search for feature on both image plane and between image plane in image pyramids are expected to produce large number of features. The fact is opposite, instead of finding sufficient number of features, BRISK has the smallest number of features detected. It might can only be explained by the BRISK implementation in openCV library that we utilized for this test. In figure 4.13 shows that even though BRISK can detect a features in testing image but it is incorrectly corresponded to descriptor in training image. Therefore, no correct matches and furthermore success idenfification is achieved.

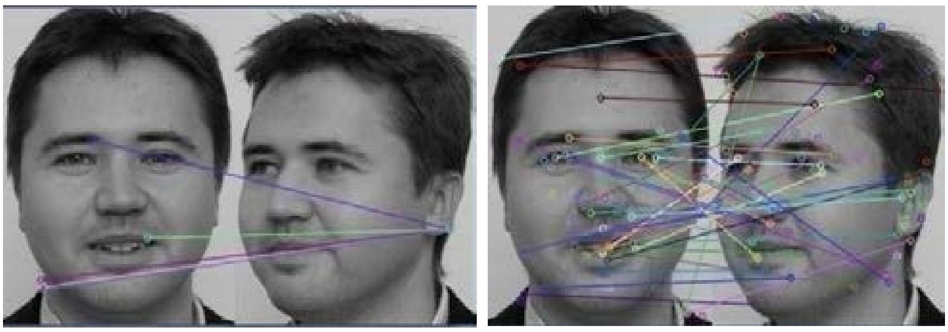
Table 4.3: Computational time of feature extraction methods in milliseconds (ms) with various pose in query images.

Methods	Detection	Description	Matching	Total
BRIEF	0.50	5.37	0.64	6.52
BRISK	1.06	5.88	0.19	7.13
FREAK	0.50	1.90	1.15	3.55
SIFT	13.87	20.43	0.38	34.68
SURF	11.70	27.66	1.09	40.45

Table 4.4: Number of features and number of correspondences feature extraction methods with various pose in query images.

Methods	#Features	#Correspondences
BRIEF	7073	1815
BRISK	48	2
FREAK	7073	1727
SIFT	359	77
SURF	2072	153

Similar experiment is done for the calculation of descriptors. It is accepted to acknowledge that the comparison is unfair because the number of features is different for each methods The results are summarized in table 4.3. The fastest descriptor is BRIEF (32 bytes), followed by BRISK (64bytes).



(a) (b)

Figure 4.13: Failure of BRISK (a) on in comparison to SIFT (b) for changed pose query images.

4.3 Overall Evaluation

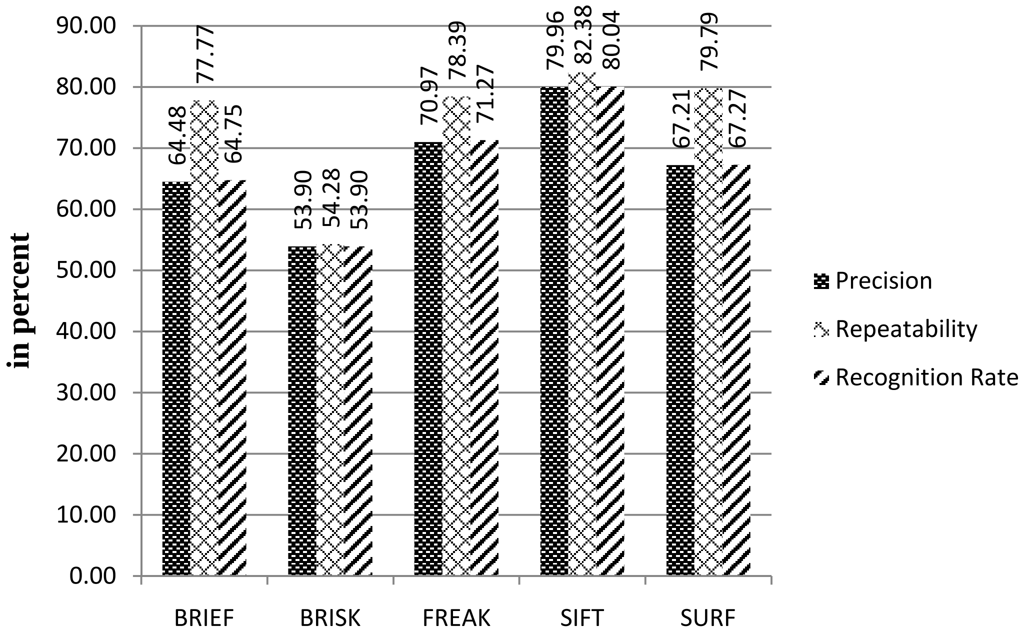


Figure 4.14: Overall comparison of feature extraction methods.

All the result are collected and the value is averaged. Overall performance evaluation result is represented in Figure 4.13. As shown in the figure, overall result does not have extreme rates among descriptors. SIFT has the highest score three parameters: repeatability, precision and recognition rate. One to notice, between repeatability and precision for both of BRISK and FREAK, there are gaps. The gaps is caused by large number of features detected by FAST detector but not all the features are good enough and able to pass reprojection error threshold. In certain application which provides sufficient amount of pixels in an image, automatically we do not need looking for too large number of features because if so, there will be redundant feature which basically unnecessary and possibly give adverse feedback by prolonging computational time which undesired.

Investigating the result for both quality and speed the trade-off between those two properties, it empowers us to make certain decision in designing face recognition system regarding its requirements. The requirements depend on environment, clients preference, cost, etc.

As time goes by, the demand 'speed' is increasingly growing. Real-time face recognition system is one common example. However the application that need accuracy the most also keep on developed such as in the field of law enforcement, bio-forensic, smart cards.

Table 4.5: Overall computational time of feature extraction methods in milliseconds (ms).

Methods	Detection	Description	Matching	Total
BRIEF	47.60	1011.14	401.41	1460.14
BRISK	181.85	19.14	6.61	207.60
FREAK	47.60	332.41	809.95	1189.96
SIFT	2102.77	3187.51	76.23	5366.52
SURF	1939.18	6233.73	464.80	8637.70

Table 4.6: Overall number of features and number of correspondences of feature extraction methods.

Methods	#Keypoints	#Correspondences
BRIEF	1516	1179
BRISK	44	24
FREAK	1516	1188
SIFT	374	308
SURF	812	648

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The aim of this thesis is to present objective evaluation of feature extraction methods using degraded facial image datasets as query images. Local feature descriptor do not possess any type of intelligence. The methods simply based on the intensity changes of pixels in digital images. SIFT showed the best performance overall. On the opposite, BRISK descriptor that relies on binary string as feature descriptor performs low. SURF yields two times number keypoints of SIFT, has repeatability not worse than SIFT but both has long computational time due to its dimensional size of feature. SIFT and SURF is good at handling image with blurring or rotation while BRIEF and FREAK good in handling images with randomly changed pose.

5.2 Recommendations

In order to select suitable feature extraction on particular recognition scenario first, we have to organize the feature detectors based on the type of image structures they extract whether it is corners, blobs or regions. Relying on the content of the image, some of these structures are more common than other structures, so the number of feature points found with a given detector may differ for dissimilar image categories. If the knowledge about the image content is little beforehand, it is mostly recommended to combine different detectors that work complementary.

FAST detector is good for if the large number of features is desired. Hence, it is wise to couple FAST with any descriptor when the testing image has low resolution. Low resolution images consist of less information. The more information we can collect the better. In the opposite, when the query is high resolution image we need the most selective feature detector thus less strong features will not be detected. Pose variation fairly being handle by high dimensional vector features such as SIFT and SURF. It is still difficult to achieve pose-invariance especially if the speed is considered as one main goal in face recognition system.

Finally, there is a number properties of the features to consider. Depending on the application scenario, some of these properties are more demanding than others. When dealing with challenging environment face recognition e.g. surveillance face recognition system, robustness to small appearance variations is important to deal with the in-class variability. For online applications or applications where a large amount of data needs to be processed, efficiency is the most important criterion.

REFERENCES

[1] Attneave, F., Some informational aspects of visual perception, Psychological Review, 61 (1954) 183-193.

[2] Tuytelaars, T. and Mikolajczyk, K., Local invariant feature detector : a survey, Foundations and Trends® in Computer Graphics and Vision 3, 3 (2008) 177-280.

[3] Beaudet, P. R., Rotationally invariant image operators, International Joint Conference on Pattern Recognition, (1978) 579–583.

[4] Itti, L., Koch, C., and Niebur, E., A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 20, (1998) 1254–1259.

[5] Wurtz, R. P. and Lourens, T. Corner detection in color images through a Multiscale combination of end-stopped cortical cells, Image and Vision Computing 18, 6 –7, (2000) 531–541.

[6] Deriche R. and Giraudon, G. A computational approach for corner and vertex detection, International Journal of Computer Vision 10, 2 (1993) 101–124.

[7] Zuniga, O. A., and Haralick, R., Corner detection using the facet model, in Proceedings of the Conference on Computer Vision and Recognition, (1983) 30–37.

[8] Crowley, J. L. and Sanderson, A. C., Multiple resolution representation and probabilistic matching of 2D gray–scale shape, IEEE Transactions on Pattern Analysis and Machine Intelligence 9, 1 (1987) 113–121.

[9] Dufournaud, Y., Schmid, C., and Horaud, R. Matching images with different Resolutions. in Proceedings of the Conference on Computer Vision and PatternRecognition, (2000) 612–618.

[10] Cour, T. and Shi, J. Recognizing objects by piecing together the segmentation puzzle, in Proceedings of the Conference on Computer Vision and Pattern Recognition, (2007).

[11] Lindeberg, T., Detecting salient blob-like image structures and their scales with A scale-space primal sketch – a method for focus-of-attention, International Journal of Computer Vision 11, 3, (1993) 283–31.

[12] Lindeberg, T., Feature detection with automatic scale selection, International Journal of Computer Vision 30, 2 (1998) 77-116.

[13] Bretzner L. and Lindeberg, T., Feature tracking with automatic selection of Spatial scales, Computer Vision and Image Understanding, vol. 71, no. 3, pp. 385–392, 1998.

[14] Mikolajczyk, K., and Schmid, C., Scale and affine invariant interest point detectors, International Journal of Computer Vision 1, 60, (2004) 63–86.

[15] Lowe, D. G. Object recognition from local scale-invariant features. in Proceedings of the International Conference on Computer Vision, (1999) 1150–1157.

[16] Alvarez, L. and Morales, F., Affine morphological multiscale analysis of corners and multiple junctions, International Journal of Computer Vision 2, 25, (1997) 95–107.

[17] Deriche, R. and Giraudon, G. A computational approach for corner and vertex detection, International Journal of Computer Vision 10, 2 (1993) 101–124.

[18] Liu, S.-T. and Tsai, W.-H., Moment preserving corner detection, Pattern Recognition 23, 5 (1990) 441–460.

[19] Malik, J., Belongie, J. Shi, and Leung, T. Textons, contours and regions: Cue integration in image segmentation. in Proceedings of the International Conference on Computer Vision, (1999) 918–925.

[20] Perdoch, M., Matas, J., and Obdrzalek, S., Stable affine frames on isophotes,

in Proceedings of the International Conference on Computer Vision, (2007).

[21] Viola, P. and Jones, M., Rapid object detection using a boosted cascade of simple features, in Proceedings of the Conference on Computer Vision and Pattern Recognition, 1, (2001) 511–518.

[22] Harris, C. and Stephens, M., A combined corner and edge detector, in Alvey Vision Conference, (1998) 147–151.

[23] Ozyurt, D., and Barton, P., Application of Targeted Automatic Differentiation to Large Scale Dynamic Optimization,

[24] Crowley, J. L. and Sanderson, A. C., Multiple resolution representation and probabilistic matching of 2D grayscale shape, IEEE Transactions on Pattern Analysis and Machine Intelligence 9, 1 (1987) 113–121.

[25] Lowe, D., Distinctive Image Features from Scale-invariant Keypoints, Cascade Filtering Approach, International Journal of Computer Vision 60 (2004) 91 – 110.

[26] Sinha, U. SIFT: Scale Invariant Feature Transform, Computer Vision Tutorial [online]. 2010. Available from : http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform

[27] Rosten, E., Porter, R. and Drummond, T., FASTER and better: A machine learning approach to corner detection, IEEE Transancions on Pattern Analysis and Machine Intelligence, (2010).

[28] Smith, S. M. and Brady, J. M., SUSAN — A new approach to low level image processing, International Journal of Computer Vision 23, 34 (1997) 45–78.

[29] Mair, E., Hager, G. D.,Burschka, D., Suppa, M., and Hirzinger, G.. Adaptive and generic corner detection based on the accelerated segment test. In Proceedings of the European Conference on Computer

Vision, (2010).

[30] Viola, P. and Jones, M., Robust Real-Time Face Detection, International Journal of Computer Vision 57, 2 (2004) 137–154.

[31] Bay, H., Ess, A., Tuytelaars, T . and Gool. L. V ., SURF: Speeded up robust features. Computer Vision and Image Understanding 110, 3 (2008):346–359.

[32] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary Robust Independent Elementary Features, European Conference on Computer Vision, (2010) 778-792.

[33] S. Leutenegger, M. Chli, R.Y. Siegwart, BRISK : Binary Robust Invariant Scalable Keypoints, IEEE International Conference on Computer Vision, (2011) 2548-2555.

[34] A. Alahi, R. Ortiz, and P. Vandergheynst, FREAK: Fast Retina Keypoint, IEEE Conference on Computer Vision and Pattern Recognition, (2012) 510-517.

[35] E.Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, International Conference on Computer Vision (2011).

[36] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27,10 (2005).

[37 Kangas, V., Comparison of Local Feature Detectors and Descriptor for Visual Object Categorization. Master Thesis at Lappeenranta University of Technology (2011)

[38] D.B. Graham and N.M. Allinson. Characterizing Virtual Eigensignatures for General Purpose Face Recognition. Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences 163. (1998) 446-456.

[39] Mislav Grgic, Kresimir Delac, Sonja Grgic. SCface - surveillance cameras

face database. Multimedia Tools and Applications Journal 51, 3 (2011) 863-879.

[40] Fischler, M. A., and Bolles R. C., Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, 24, (1981) : 381-395.

[41] Khvedchenia, I., Feature descriptor comparison report, Computer Vision Talks [online]. 2012. Available from http://computer-vision-talks.com/2011/08/feature-descriptor-comparison-report/

[42] Quinlan, J. R., Induction of decision trees, Machine Learning 1, (1986) 81–106.

[43] Zhao, W., Chellappa, R. , Rosenfeld, A., Phillips, P. Face recognition: a literature survey, ACM Computing Surveys, (2003) 399–458.

[44] Chellappa, R., Sinha, P., Phillips, P. J., Face Recognition By Computer and Humans, Computer 4, (2010) 46-55.

[45] Muhfizaturrahmah and Aramvith, S., Face Recognition for Surveillance Video Application: A Short Review, ASEAN Academic Society International Conference (2012) 51-55.

[46] Muhfizaturrahmah, Lertniphonphan, K., Aramvith, Watcharapinchai, N., Face Recognition for Low Resolution Facial Image:Comparison of Local Descriptors, 16th International Workshop on Advance Image Technology (2013) 767-771.

[47] Muhfizaturrahmah and Aramvith S., Binary Robust Independent Elementary Features for Rapid Face Recognition, AUN/SEED-Net Regional Conference on Electrical and Electronics Engineering, (2013) 156-159.

Biography

Muhfizaturrahmah was born on September 1st, 1987 in Barabai, a town in Southern Borneo Island, Indonesia. She obtained bachelor degree from Department of Engineering Physics, Universitas Gadjah Mada, Yogyakarta, Indonesia on 2010. She started her study in Department of Electrical Engineering Chulalongkorn University on June 2011 with research interests: image and video processing, face recognition and computer vision for surveillance system.

List of Publications:

1. “Face Recognition for Surveillance Video Application: A Short Review,” accepted in ASEAN Academic Society International Conference (AASIC), 7-8 December 2012, Hatyai, Thailand.
2. “Face Recognition for Low Facial Image Resolution: Comparison of Local Descriptors”, accepted in International Workshop on Advance Image Technology (IWAIT), 7-9 January 2013, Nagoya, Japan.
3. “Binary Robust Independent Elementary Features for Rapid Face Recognition”, accepted in AUN/SEED-Net Regional Conference on Electrical and Electronics Engineering, 4-5 February 2013, Bangkok, Thailand.