

# **Registration of face image using modified BRISK feature descriptor**

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY

IN

ELECTRONIC SYSTEMS AND COMMUNICATIONS

BY

PRANAV G S

ROLL NO. -213EE1284



**Department of Electrical Engineering**

**National Institute of Technology, Rourkela-769008**

2015

# **Registration of face image using modified BRISK feature descriptor**

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY

IN

ELECTRONIC SYSTEMS AND COMMUNICATIONS

BY

PRANAV G S

ROLL NO. - 213EE1284

Under the Guidance of

PROF. DIPTI PATRA



**Department of Electrical Engineering**

**National Institute of Technology, Rourkela-769008**

2015



National Institute Of Technology, Rourkela

## CERTIFICATE

This is to certify that the thesis entitled “**Registration of face image using modified BRISK feature descriptor**” submitted by **MR. PRANAV G S** in partial fulfillment of the requirements for the award of Master of Technology Degree in **Electrical Engineering** with specialization in “**ELECTRONIC SYSTEMS AND COMMUNICATION**” at National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

**Prof. Dipti Patra**

Department of Electrical Engineering

National Institute of Technology

Rourkela

# ACKNOWLEDGEMENT

Firstly, I would like to thank Prof. Dipti Patra, Department of Electrical Engineering, National Institute of Technology, Rourkela as my guide, who encouraged and challenged me throughout my project work, thesis and to put effort on our workings. She is a perfect motivator and I am glad to have guide like her.

After that, I would like to thank, Prof. Anup Kumar Panda, head of department, electrical engineering, National Institute of Technology, Rourkela and professor of my specialization Prof. Prasanna Kumar Sahu, Prof. Susmita Das, Dr. Supratim Gupta and Prof. K. Ratna Subhashini.

I would also like thanks my fellow lab mates in Image Processing & Computer Vision Lab: Rajkumar Maharaju, Saka harshavardhan, Bedadatta bedanta, Smita Pradhan, Rajashree Nayak, Umesh kumar sahu, and Yogananda Patnaik for their help and support.

PRANAV G S

## Abstract

Automatic face recognition is a hot area of research in the field of computer vision. Even though a lot of research have been done in this field, still researchers are unable to develop an algorithm which can detect the face images under all possible real time conditions. Automatic face recognition algorithms are used in a variety of applications such as surveillance, automatic tagging, and human-robot interaction etc. The main problem faced by researchers working with the above real time problems is the uncertainty about the pose of the detected face, i.e. if the pose of the sensed image differ from the images in the trained database most of the existing algorithms will fail. So researchers suggested and proved that the detection accuracy against pose variation can be improved if we considered image registration as a preprocessing step prior to face recognition. In this work, scale and rotation invariant features have been used for image registration. The important steps in feature based image registration are preprocessing, feature detection, feature matching, transformation estimation, and resampling. In this work, feature detectors and descriptors like SIFT, SURF, FAST, DAISY and BRISK are used. Among all these descriptors the BRISK descriptor performs the best. To avoid mismatches, using some threshold values, a modified BRISK descriptor has been proposed in this work. Modified BRISK descriptor performs best in terms of maximum matching as compared to other state of arts descriptors. The next step is to calculate the transformation model which is capable of transforming the coordinates of sensed image to coordinates of reference image. Some radial basis functions are used in this step to design the proper transformation function. In resampling step, we used bilinear interpolation to compute some pixels in the output image. A new algorithm is proposed in this work to find out the possible image pairs from the train database corresponds to the input image, for doing image registration. In this work, image registration algorithms are simulated in MATLAB with different detector-descriptor combination and affine transformation matrix. For measuring the similarity between registered output image and the reference image, SSIM index and mutual information is used.

# Contents

CERTIFICATE.....	i
ACKNOWLEDGEMENT .....	ii
Abstract.....	iii
List of figures.....	vi
List of tables.....	vii
<b>CHAPTER 1</b> .....	<b>1</b>
1. Introduction.....	1
1.1 Important face recognition algorithms developed so far.....	2
1.2 Challenges faced by face recognition algorithms .....	2
1.3 Possible techniques for pose invariant face recognition .....	2
1.4 Motivation.....	4
1.6 Thesis objective .....	6
1.7 Thesis contribution.....	7
1.8 Thesis Organization .....	7
<b>CHAPTER 2</b> .....	<b>8</b>
2. Image registration .....	8
2.2 Feature detection.....	10
2.2.1 Area-based methods.....	10
2.2.2 Feature-based methods.....	10
2.3 Feature matching.....	11
2.3.1 Area based methods .....	11
2.3.2 Feature-based methods.....	12
2.4 Transform model estimation .....	13
2.4.1 Global mapping models. ....	13
2.5 Image resampling and transformation.....	14
2.5.1 Bilinear Interpolation .....	15
<b>CHAPTER 3</b> .....	<b>16</b>
3. Study of different feature detectors and descriptors .....	16
3.1 SIFT - Scale Invariant Feature Transforms.....	16

3.1.1 Local extrema detection .....	18
3.1.2 Orientation assignment .....	18
3.1.3 The local image descriptor .....	19
3.2 Speeded Up Robust Feature Detector (SURF).....	20
3.2.1 Fast Hessian Detector.....	21
3.2.2 Orientation Assignment .....	22
3.2.3 Descriptor components .....	23
3.4 FAST: Features from Accelerated Segment Test.....	26
3.5 BRISK: Binary Robust Invariant Scalable Key points .....	27
3.5.1 Scale Space Key point Detection .....	27
3.5.2 Key point Description .....	29
3.5.3 Sampling Pattern and Rotation Estimation. ....	29
3.5.4 Building the Descriptor.....	30
3.5.5 Descriptor Matching .....	31
<b>CHAPTER 4</b> .....	<b>33</b>
4. Proposed Modified BRISK descriptor .....	34
4.1 Modified BRISK Descriptor .....	34
4.2 Proposed Algorithm to find out the possible reference image pair for face registration .....	36
4.3 Output of registration algorithm with different feature extractors.....	38
<b>CHAPTER 5</b> .....	<b>46</b>
5. Conclusion and future scope.....	46

# List of figures

Figure 1 face image of a person with different poses .....	1
Figure 2 Block diagram for feature based face image registration.....	3
Figure 3 Bilinear Interpolation: .....	15
Figure 4 Algorithm for the construction of difference of Gaussian.....	17
Figure 5 Technique to find extreme point from difference of gaussian .....	18
Figure 6 Key-point descriptor for SIFT .....	19
Figure 7 Scale pyramid for computing the key point .....	20
Figure 8 Gaussian second order partial derivatives in y-direction and xy-direction .....	22
Figure 9 the SURF descriptor .....	22
Figure 10 Algorithm to find out DAISY descriptor components .....	24
Figure 11 The DAISY descriptor.....	25
Figure 12 point segment test corner detection in an image patch.....	27
Figure 13 Scale-space interest point detection in BRISK detector.....	28
Figure 14 The BRISK sampling pattern .....	30
Figure 15 Repeatability vs SSIM plot of various descriptor.....	32
Figure 16 Proposed algorithm for feature based face image registration .....	37



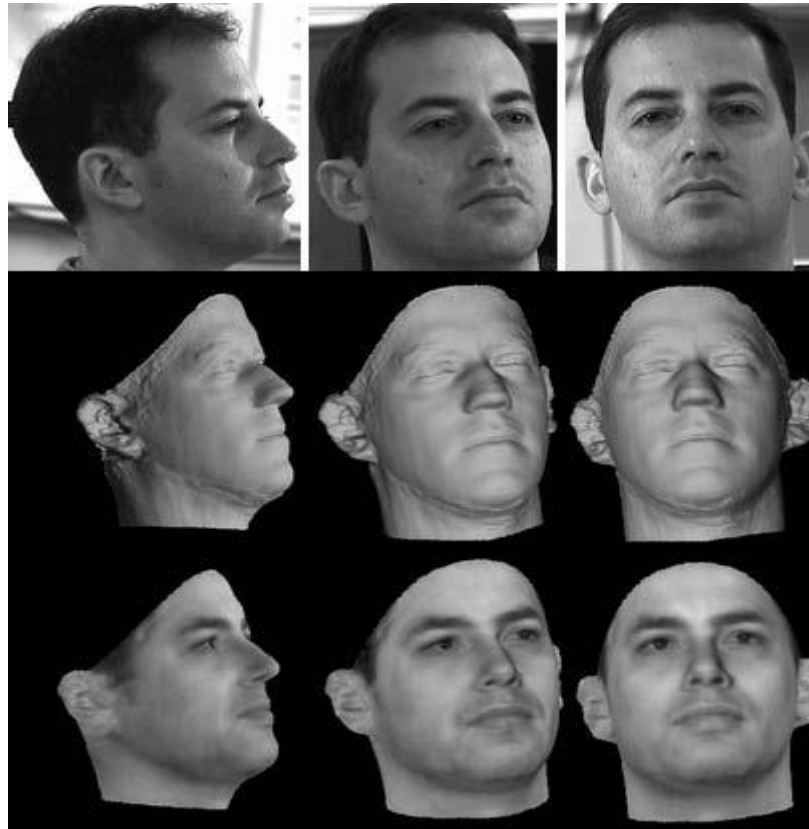
# List of tables

Table 1 Comparison table for different detector/descriptor.....	333
Table 2 Performance analysis of different feature based face registration.....	41
Table 3 comparison of registration algorithms using different feature detector-descriptor combination using head pose database.....	45

# CHAPTER 1

## 1. Introduction

In this modern world face recognition algorithms are commonly used in many places. Attendance systems, biometric locking systems and automatic tagging systems are some of the examples for this. We can find more application also. But from our experience we can say that none of the face recognition algorithms developed so far are perfect. Illumination variation, contrast variation and effect of noises are the main possible reasons which will reduce the performance of a face recognition algorithm. In many applications like surveillance, automatic tagging, and human robot interaction most of the face recognition algorithms will fail. Because in these particular applications the pose of the face images will be different from the pose of the image in the train database. The figure 1 as shown below shows different poses of a person. In the train database how many images we can store? If another pose which is not available with the train database comes it will not get detected or false detection can takes place.



*Figure 1 face image of a person with different poses*

## 1.1 Important face recognition algorithms developed so far

Statistical methods are the one of the most famous methods for face recognition. These methods includes Eigen faces [1], ICA [2] and LDA [3]. These methods are based on the assumption that a face image can be denoted as the sum of independent components determined from other trained images in the database. These components are directly determined from the pixel values. So, variations in poses of the face, translation, scaling and rotation will reduce the rate of accurate recognition on those methods. Elastic bunch Graph matching is another important method for face recognition. In this method recognition of face images takes place by computing a set of features using a data structure called a bunch graph. The main disadvantage associated with this method is it is too much sensitive to illumination variation. Template matching is another method which uses face templates from different prospects to describe a single image. Compared with other methods this method is very easy for implementing. Local binary pattern based face recognition is another important technique [4]. This method gives some robust method against different face expressions. Face recognition is possible by computing the geometrical features of the face images also. These methods are giving less accuracy and takes more time for computation.

## 1.2 Challenges faced by face recognition algorithms

Illumination changes are the one of the main problems faced by the early developed face recognition algorithms. Because face recognition is a real time operation the computational complexity also can take as a matter of concern. Face expressions and noises are examples of advanced problems. But for most of the above stated problems we have solutions. Face image frontal pose change is the one of the most challenging issue in face recognition. Most of the existing algorithms need proper front pose for correct detection. But how it is possible for automatic tagging, surveillance, or tracking.

## 1.3 Possible techniques for pose invariant face recognition

Adding image registration as a initial step to face recognition can be a good step for achieving pose invariance. Image registration is the manual or automatic technique which tries to find out the correspondence between 2 or more images of the same scene but captured at different conditions and spatially align them to the same coordinate system. Here the trained images in the

data base is called as target image and the detected image which is the output of sensor is called image. In image registration we will change the alignment of sensed image to that of the reference image. In other words we will change the coordinates of the detected image to coordinates of the target image. This can improve the accuracy of recognition. There are mainly 2 types of image registration algorithms are there Intensity based and feature based. Intensity based methods are sensitive to illumination changes and noises. But feature based methods gives best results against illumination changes and noises. Block diagram for feature base face image registration is shown in the block diagram below.

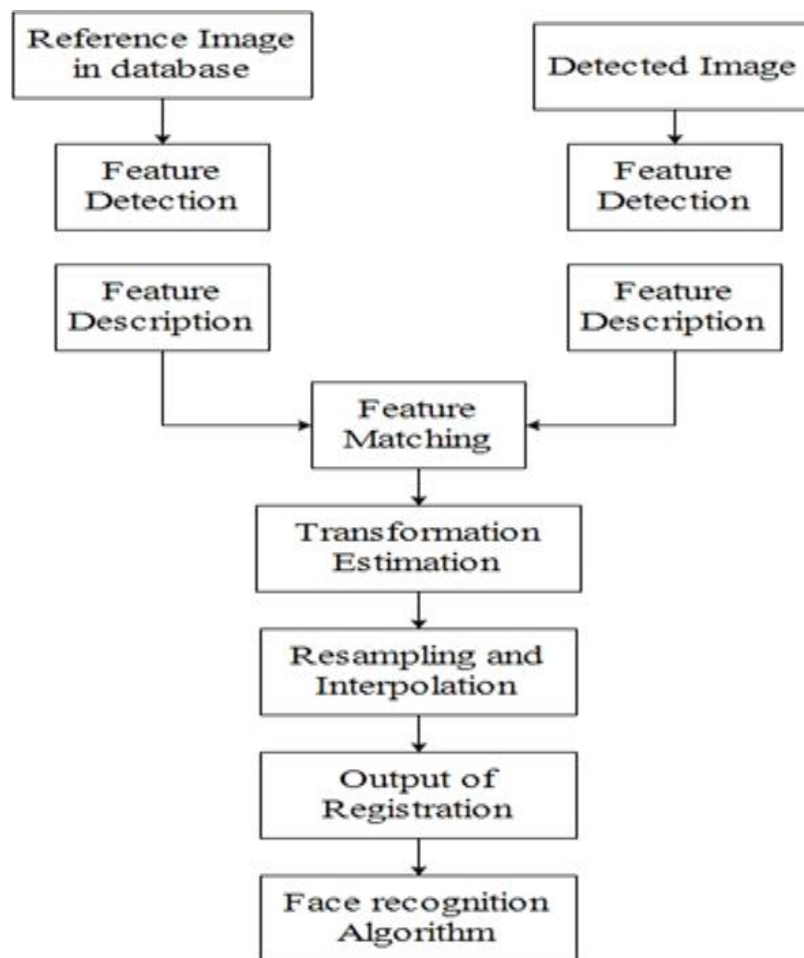


Figure 2 Block diagram for feature based face image registration

Feature detection, feature description and feature matching are the first step in feature based registration. The features using for registration must be scale, rotation, and noise invariant. Also the features must be stable and distinctive under all conditions are preferred. Here in this work feature detectors and descriptors like Scale Invariant Feature Transforms [7], Speeded Up Robust Feature [8], Features from Accelerated Segment Test [9], DAISY [10], and Binary Robust Invariant Scalable Key points [11] are used.

For SIFT, SURF and DAISY descriptor matching can be done by calculating its nearest neighbor in the database of interest-points from target images. The nearest neighborhood is nothing but the one with minimum geometrical distance. In the case of BRISK and its modified version the minimum hamming distance corresponds to nearest neighborhood. Transform estimation has been performed using some radial basis function [12]. In this work we are using this function because it handle even locally varying geometric transformations also. For the last step image resampling, image interpolation techniques are required. We have gone through many sophisticated interpolation techniques like quadratic splines and cubic B-splines, etc. But in this work we used bilinear interpolation only because it gives the best tradeoff between computational complexity and accuracy.

#### 1.4 Motivation

The demand for face-recognition algorithms with good accuracy and invariance with all possible geometrical deformations are increasing day by day. The reasons are demand for computer based security systems increasing tremendously. But still we don't have any algorithm which will give best result in all conditions. Illumination changes, low contrast and noises are the common problems. Now we have good solution for all of these, but pose invariance still it is not having proper solution. Some commonly used techniques like PCA, LDA and ICA are basically statistical analysis. These methods are assumes that a face image can be represented as the sum of independent components which can be directly determined from the pixels. Therefore any changes with the face images including face positioning, scaling, and/or rotation can decrease the probability of correct detection. Later on some new methods came with local binary pattern and its new modifications. Compared with the earlier methods LBP methods are for dealing with pose variation and misalignment. The reason is LBP descriptor is rigid descriptor with a good matching strategy. Similarly many of the face recognition algorithms are not particularly aims at pose

invariance. If we use image registration as a preprocessing step prior to face recognition it can spatially rearrange the alignment of detected image with reference image. So the face recognition algorithms will become robust against pose variation. By using scale and rotation invariant features correct matching can be done even if the detected image has undergoes a locally varying deformation, or illumination changes. From the output of the registered image it will be easy for performing face recognition. The face image registration concept will be useful for modeling of face image from the face image with different poses obtained from CCTV. This will be a useful thing for police men and other crime detection agencies. Similarly this will be useful for social media also for automatic tagging. i.e. if a person's photos with different pose is there in database another pose also can detect using this face registration. Similarly the tremendous application of face registration in different field motivates me to do this work

## 1.5 Literature Survey

- ❖ **Lisa Gottesfeld Brown** [5] has given a detailed survey on image registration methods. This paper is mainly concentrated on Intensity based image registration. Correlation and Sequential Methods, Fourier Methods, Point Mapping and Elastic Model-Based Matching are the main image registration methods explained in this paper. He explained the characteristics of the above methods also.
- ❖ **Barbara Zitova** [6] published a comprehensive survey on image registration techniques. He explained both intensity based and feature based techniques in this paper. Feature based techniques are the main topic discussion in this paper. He explains many methods using for transformation estimation and interpolation.
- ❖ **David G. Lowe** [7] introduced a new feature detector and descriptor called scale invariant feature transform. This is a both scale and rotation invariant descriptor. This method allow us to compute distinctive key-points of an image and a descriptor vector so that we can find out the correspondence between images.
- ❖ **Herbert Bay, Tinne Tuytelaars, and Luc Van Gool** [8] modified the SIFT descriptor and proposed a new feature detector cum descriptor known as speeded up robust features (SURF). Here the approximation has been done using hessian matrix and haar wavelets. As its name indicates this descriptor is faster than SIFT, but less accurate comparatively.

- ❖ **Edward Rosten and Tom Drummond** [9] has developed a new detector algorithm called FAST: Features from Accelerated Segment Test. This detector is used where speed is important than accuracy.
- ❖ **Engin Tola, Vincent Lepetit, and Pascal Fua** [10] developed an efficient image descriptor, DAISY, which is very efficient to compute densely. Compared with previously explained algorithms this also compute descriptor at every pixels. But this is faster compared with SIFT and efficient compared with SURF.
- ❖ **Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart** [11] proposed a new method for interest point detection, description and matching called BRISK. Here the detector is comparably fast because it is based on scale-space FAST detector. This is binary descriptor so descriptor is nothing but a bit stream.
- ❖ **M. Ehlers, D.N. Fogel** [12] proposed a method for estimating high precision transformation function using multi quadratic functions.
- ❖ **Daniel Russakoff, Carlo Tomasi, Torsten Rohlfing, Calvin R. Maurer** [13] used concept of Mutual information for measuring the similarity of 2 images. The main drawback with MI method is MI considers each pixels independently so relation between the pixel and its neighboring pixel is not considered.
- ❖ **Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, Eero P. Simoncelli** [14] proposed that SSIM is the best method for image similarity measure as it do luminance similarity, Contrast similarity and Structure similarity.

## 1.6 Thesis objective

The objectives of the thesis can be described as follows.

- To study about the possibility of using different feature based face image registration techniques that can be used as a preprocessing step for face recognition, so that the face recognition can become pose invariant.
- To study and compare the performance of different feature detectors-and descriptor that can be used for face registration
- To develop a better feature descriptor and compare it's performance with existing ones.

- Compare the output of the registration algorithms using different detector-descriptor combinations with SSIM index.

### 1.7 Thesis contribution

- Proposed a new image registration based method as a preprocessing step for face recognition so that it become pose invariant.
- Proposed a modified version of BRISK descriptor which is more robust than the original descriptor.
- A new algorithm has been developed to find out the possible image pairs from the train database corresponds to the input image, for doing image registration.

### 1.8 Thesis Organization

In this thesis the chapter 1 covers introduction, motivation and literature survey. A brief explanation about the face recognition algorithms developed so far and challenges faced for each method is explained here. How to achieve pose invariance through image registration is the problem addressed here. Chapter 2 explains about image registration algorithms, especially feature based registration algorithms, and transform estimation and resampling. Chapter 3 is the study of different feature detection-description algorithm for face image registration.



# CHAPTER 2

## 2. Image registration

Image registration is the technique which tries to find the correspondence between two or more images of the same scene capture at different times, from various view angle, and/or by various sensors and spatially align them to the coordinates of the target image.

Before exploring the various methods developed by researchers, here we are introducing some basic terms which are often used in image registration literature.

**Reference image:** the image which fixed as the target. The expected output should be in the coordinates of this image.

**Sensed image:** the image which is spatially mapped to be aligned with the reference image.

**Transformation:** the mapping function used to map the sensed image to the reference image.

Because of the different types of images (taken at different time, angle, equipment) that need to be registered and the various degradation types that the image has undergone, it is not possible to define an generalized method which can be used in all registration assignment, because we have to consider not only the geometric deformation but also the presence of noise in the images which will be unique for each and every cases.

Nevertheless, we can summarize the main steps of image registration as

1. Feature detection

(extracting key-points and describing them using the neighborhood)

2. Feature matching

(finding feature correspondence between 2 images)

3. Transformation estimation

(find out the transformation function)

#### 4. Resampling

(use the transformation function to change the coordinates of the detected image to that of the targeted image)

The implementation of each step in image registration has its basic problems and we have to pick up the right feature for our particular task. The features must be distinguishable objects under all imaging conditions. The detected and the reference image must have sufficient number of common feature elements even if the sensed image under gone some geometric deformations.

The detection techniques must have good detection accuracy and less sensitive to the possible image degradation and noises. In an ideal case, the detection technique should have the ability to detect the same features every projections in the scene irrespective of the deformation that the image has undergone. Incorrect detection of features, image degradation and presence of noise, can arise as problem in the feature matching step. Sometimes features corresponds to the same key-points in 2 different images can be different due to the different imaging situations like poor lighting and/or sensor spectral sensitivity.

The feature descriptor should be designed by taking these considerations into account. That means the descriptor should be invariant to the assumed degradation, rotation, scaling. It must be able to different features for different key-points. The feature matching algorithm must be good enough to match the same features in both images without error even though many similar features are present in the sensed image

The kind of transform function must select in accordance with the previous knowledge about the image acquisition method and the possible image degradations. If prior information are unavailable, the mapping function must be flexible enough to manage all possible deformations which we can expect. The quality of the feature detection algorithm, the robustness of feature descriptor, the reliability of the feature matching algorithm, and the tolerable approximation error need to be considered too.

Finally, the choice for the suitable technique for resampling depends on the compromise between the accuracy expected for the interpolation method as well as the computational easiness (time required also). The nearest-neighbor techniques or bilinear interpolation techniques are will give

best results in most of the applications, but some applications require more advanced methods like spline interpolation and etc.

## 2.2 Feature detection

Selection of proper key points from an image is an important step, because the features must be easily identifiable. If the features are invariant and robust the matching will become more accurate, and the number of detected features must be high enough to ensure proper matching and reduce number of incorrect matches. These type of features should not be determined directly on the intensity values of the image or gray level values because light intensity or noise in the sensor can vary among sensed and reference images.

### 2.2.1 Area-based methods

For area based methods feature detection step is not required. So first step in image registration can be omitted. Area-based methods put more importance on the feature description step than the detection step.

### 2.2.2 Feature-based methods

The feature based approaches are generally based on the extraction of unique interest points, features in the images. Significant regions, lines or points are considered as features here. They should be distinct, unique and uniformly distributed over the image and easily distinguishable in both the images. They are supposed to be unaffected by time and will be stable at the exact positions throughout the total experiment.

The repeatability of the feature elements in both the source and target images is confirmed by the invariance to the possible variations and accuracy of the feature detector. In other way, the total number of feature elements which are common in both the detected and reference image must be high enough, irrespective of the geometrical changes, presence of additive noise, and of other possible deformations in the sensed image. Opposite to the area-based methods, the feature-based techniques are not directly depends on the pixel values. The features contains more information

than the intensity value of each pixels. This advantage makes feature-based methods appropriate for some places where light variations are expected or multisensory analysis is required.

#### 2.2.2.1 Point features

Point features are the features which are used most widely for image registration methods because they will give a highly unique and distinguishable description about the key-points. The descriptor generally find out the magnitude and orientation of the key-points using the neighborhood regions. For some descriptors transforms are used for approximating the key-point description process. In binary descriptors the descriptor is nothing but a binary string obtained from simple intensity comparison of pixels.

### 2.3 Feature matching

After extracting the key-points and its corresponding descriptor vector, for both detected and targeted images, feature correspondence has to be established. Feature matching is the technique responsible for this. Matching can be established either by image intensity values of the pixels nearby, or by the distance between feature descriptor vectors depending on the specific registration method. The two major classifications under feature matching are area-based and feature-based methods.

#### 2.3.1 Area based methods

Area-based methods, sometimes called correlation-like methods or template matching method. These methods are not considering the structure and salient features of the objects there. Here templates of predefined size or even total images are used for the matching. The area based methods have lot of limitations. The first problem is the rectangular window can support for translations only, not to locally occurring deformations. In other words if the images undergone some local complex transformations, rectangular windows won't be able to cover all parts of the scene in the target and detected images. Another limitation associated with area-based techniques are commonly known as 'remarkableness' of the template content. i.e. if the mask used for template matching contains smooth areas without any significant details, it will results in incorrect matching with other soft areas in the reference image. The features using for image registration

must be comfortably detected in different location of the image. Conventional area-based methods like cross-correlation is purely depends on image intensity and not on the structure of the image. So obviously, these methods will be sensitive to the intensity variations introduced by the addition of noise and light changes

### 2.3.2 Feature-based methods

In feature based methods after the feature extraction step in both reference and detected image pairwise correspondence has to be computed. The most commonly used method in this step are methods using spatial relations and the methods using invariant descriptors.

#### 2.3.2.1 Methods using spatial relations

When the detected features are unable represented as a descriptor vector (the knowledge about the neighborhood is not properly available), then methods under the above classes are used.

#### 2.3.2.2 Methods using invariant descriptors

Here in this method feature descriptor corresponds to the interest point is used for establishing the matching or correspondence between reference and sensed image. Feature descriptors which can handle the expected image deformation are preferably used. The descriptor which is used here should satisfy various criteria. The most important property is the invariance (even if the sensed image undergoes rotation, scaling and/or any locally varying geometric deformations the descriptor vector of the corresponding key-points from the targeted and detected image must not show any difference), uniqueness (two distinct features shouldn't have same descriptions), stability (the descriptor vector of a key-point and slightly deformed version of the same key-point in should be close to each other), and independence (the elements of the feature description vector should be mathematically independent ). However, generally it is not required to satisfy all these criteria together and it is required to find out a suitable trade-off. Feature elements from the detected and target images with the maximum closer invariant descriptions are said to matching pairs. The choice of the type of the detector to be used depends on the feature properties and the expected geometric changes occurred with the images. When trying to find out the best matching feature pairs between sensed and reference images, the minimum distance rule is often used

## 2.4 Transform model estimation

Construction of the mapping function is the next important step in feature based registration after feature correspondence between the reference and the sensed image has been found out. This transformation function is the backbone of the registration process as it is used for changing the coordinates of the sensed image to coordinates of targeted image. Choosing the appropriate transform function and estimating its coefficients are the challenging and difficult tasks in this step. The possible geometric distortions of the detected image, the method used for capturing the image and the quality required for the registration are the important aspects to consider while designing the transformation function. Global mapping models and Local mapping models are the two important classifications of the transformation functions.

### 2.4.1 Global mapping models.

One of the most commonly used global models uses two variable polynomials of low degrees. This model uses similarity transform which consists of scaling, rotation and translation operations.

$$\begin{aligned} u &= s(x \cos(\varphi) - y \sin(\varphi)) + t_x \\ v &= s(x \sin(\varphi) + y \cos(\varphi)) + t_y \end{aligned} \dots\dots\dots (1)$$

This model is commonly known as shape saving model as it keeps the shape (angle and curves) as it is.

More generalized and linear version of the above model is called affine transformation.

$$\begin{aligned} u &= a_0 + a_1x + a_2y \\ v &= b_0 + b_1x + b_2y \end{aligned} \dots\dots\dots (2)$$

But this transform models can't handle locally occurring distortions and transformations. So this transform is not suitable for most of the practical applications like medical imaging and airborne imaging.

### 2.4.2 Local mapping models

This method basically aims at solving the limitations of global mapping models at local geometric

changes. The weighted least square method and weighted mean methods are the 2 important local mapping models.

### 2.4.3 Mapping by radial basis functions

Radial basis function based mapping functions are capable of handling locally occurring geometric variation. Basically they are the extended version of global mapping methods. The general form of the mapping function is the sum of linear combination of translational radially symmetric function and a low-degree polynomial.

$$u = a_0 + a_1x + a_2y + \sum_{i=1}^N c_i g(x, x_i) \dots\dots\dots (3)$$

and similarly v also.

These functions were originally designed for interpolation operation for irregular surfaces. The name ‘radial’ indicates an important characteristics of this function that the value at each point in it depends only on the distance between the point and the key-points, not on its specific location.

## 2.5 Image resampling and transformation

The transformation functions developed in the last stage are used to map the detected image to the targeted image and so as to perform image registration. The mapping operation can be implemented in 2 ways, Forward or backward methods. In forward method each pixel from the detected image can be directly mapped using the calculated transformation functions. Compared with backward method this is a difficult process to establish. In this method quality of output image also will be low due to the discretization and rounding. This will make holes, gaps and/or overlap regions in the registered image. So the backward approach is the commonly used method. The output image in the registration operation can be determined by using the coordinates of the reference image and the inverse of the calculated transformation function. The image interpolation operation performed in the detected image on the regular basis. In this way possibility for occurrence of holes, gaps and/or overlaps can be avoided in output image.

The interpolation operation can be implemented by performing convolution operation between the image and the interpolation kernel. 2D sinc function is known as the optimal interpolant kernel.

But this kernel is hard to implement in practical because ideal sinc function is not causal. Thus, lot of simpler interpolants has been developed for reducing the computational cost. To improve the computational easiness separable interpolants are preferably considered. The separability means it allows to split an  $m \times m$  2D convolution operation into  $(m+1)$  1D convolution operation which is comparably faster.

The most commonly used interpolants for image registration includes the nearest neighbor function, the bilinear and bicubic function etc. Bilinear interpolation is the most commonly used interpolant among all of the above, even though it is offering less performance compared with another sophisticated methods with respect to robustness, accuracy, quality and visual appearance of the output image. The reason behind this is nothing but this method gives the best compromise between accuracy and computational easiness.

### 2.5.1 Bilinear Interpolation

Here the pixel values in the new coordinate system can be calculated by arithmetic mean of the 4 neighboring pixels around the key-point. This interpolation kernel will make the image smother and introduce some new values which are totally newer to the output image. Comparing with position accuracy this method is better than nearest neighborhood method.

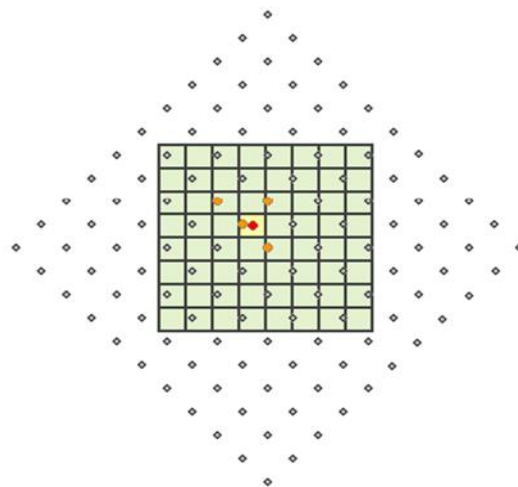


Figure 3 Bilinear Interpolation:



# CHAPTER 3

## 3. Study of different feature detectors and descriptors

### 3.1 SIFT - Scale Invariant Feature Transforms

SIFT [7] is an efficient scale and rotation invariant detector cum descriptor which is commonly used for feature matching and registration. This descriptor is comparatively less sensitive about scaling transformations in the image domain and robust to locally occurring deformation and light intensity variation. SIFT consists of four different steps: scale-space key point detection, key point localization, orientation assignment and feature description.

The initial stage of key point detection is to find out the position and scales of the key-points that can be repeatably assigned under various viewing conditions of the same target. Detection of scale invariant points of the image can be achieved by checking for stable key-features throughout all possible each and every scales, using a special function of scale called as scale space. It has been experimentally proved under some reasonable assumptions that Gaussian function is the only possibility for a scale space kernel. Therefore, the scale space function of an image  $L(x, y, \sigma)$ , can be defined as the convolution between the variable-scale Gaussian kernel,  $G(x, y, \sigma)$ , and the input image,  $I(x, y)$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \dots \dots \dots (5)$$

where ‘\*’ is a mathematical operator which denotes convolution.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \dots \dots \dots (6)$$

Now the next step is to find out the scale space extrema point. To achieve this goal we used the result of convolution between DoG function and input image.  $D(x, y, \sigma)$  can be obtained from the difference between the two adjacent scales differed by a constant scaling factor  $k$ . By this method we can accurately detect the stable candidate point locations and their scale in the scale space.

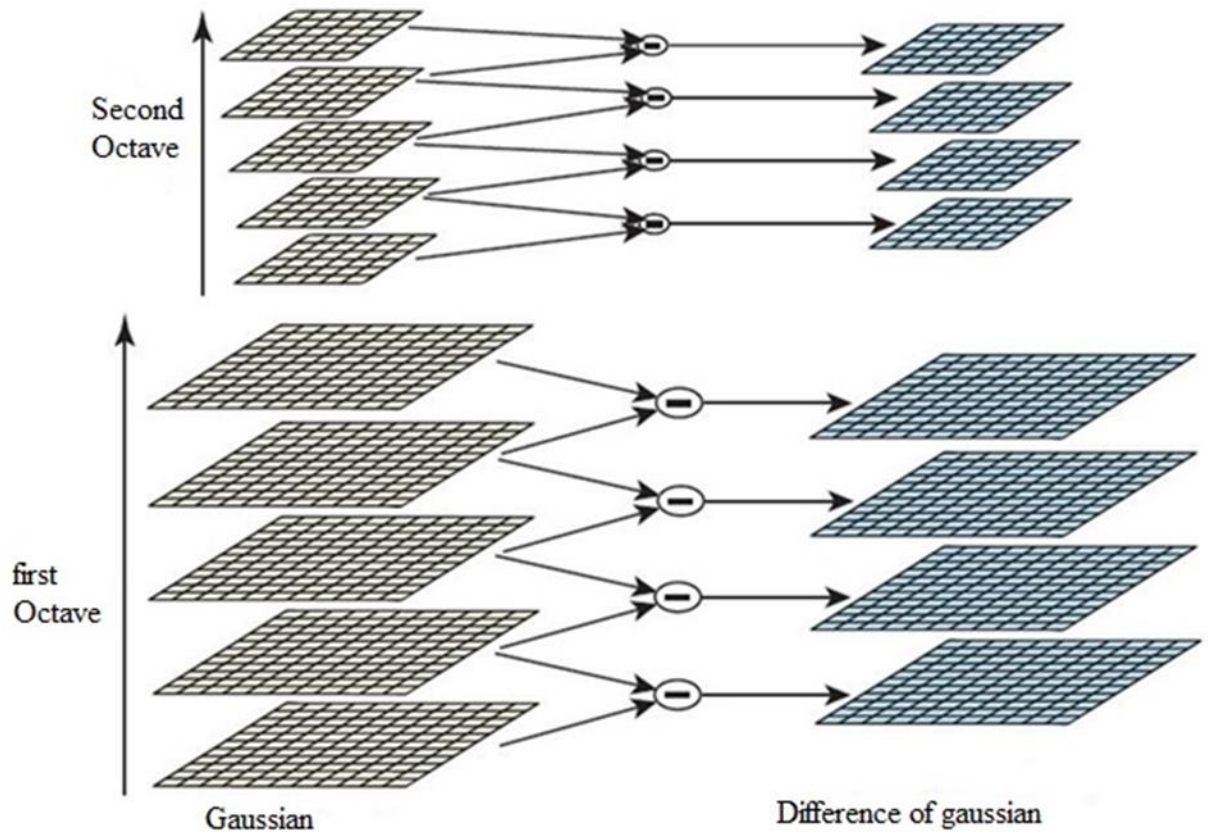


Figure 4 Algorithm for the construction of difference of Gaussian

$$\begin{aligned}
 D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\
 &= L(x, y, k\sigma) - L(x, y, \sigma) \dots \dots \dots (7)
 \end{aligned}$$

A robust algorithm for the implementation of  $D(x, y, \sigma)$  is shown in Figure 4. The original image is convolved with Gaussian kernel with increasing scale to produce images differed by a constant scaling element  $k$  in scale space, illustrated in the figure above. Each octaves in the scale space is obtained by scaling the previous octave with a factor 2. Typical value of the number of images required in each octaves are fixed as  $s + 3$ . The difference between neighboring image scales in the same octave are calculated to find out the DoG. After a total octave has been formed, we up sample the Gaussian image with a scaling factor  $2\sigma$  by taking every  $2^{\text{nd}}$  point in each and every row and column. The precision of sampling operation depend to  $\sigma$  is unchanged compared with the initial stage of the previous octave, while computational easiness is comparably increases.

### 3.1.1 Local extrema detection

For detecting the maximum or minimum point of  $D(x, y, \sigma)$ , we have to compare each key-point to its all the neighbors in the present frame (eight) and all the neighbors in the scale top and bottom (nine) (see Figure 5). The point will consider it as the interest point if and only if it is larger or lesser than all of these neighbors. The expense for this search (computational time) is considerably less because of the fact that many of the candidate points will be expelled in the first few search.

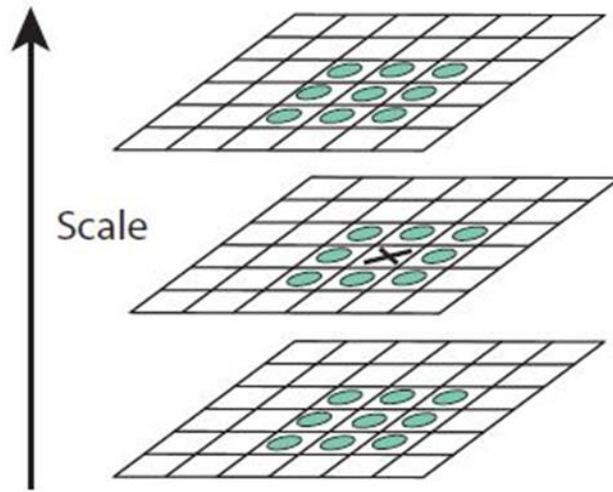


Figure 5 Technique to find extreme point from difference of gaussian

### 3.1.2 Orientation assignment

By including the exact orientation and gradient magnitude to each and every interest point based on the neighborhood of the interest point, the interest point descriptor become rotation invariant. The scale of the detected candidate point is utilized in the selection of the Gaussian kernel to smooth the image,  $L$ , with the nearest scale, so that all operations can be done in a scale-independent manner. For a particular image sample,  $L(x, y)$ , at its scale, the gradient magnitude, is denoted as  $m(x, y)$ , and orientation is denoted as  $\theta(x, y)$ , can be computed using the difference between adjacent pixels.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \dots (8)$$

The orientation histogram of this descriptor is made from the orientations of the key-point inside a local area around the interest point. The orientation histogram consists of total 36 bins around the key-point.

### 3.1.3 The local image descriptor

The next step is to calculate a key-point descriptor by considering the local neighborhood of the interest point that is highly distinguishable and unique yet is as invariant to scale changes and rotation. This descriptor is invariant to variations, such as change in lighting or other degradation.

#### Descriptor representation

The candidate point descriptor is formed by first calculating both the gradient magnitude and orientation of the neighborhood area around the key point location, as illustrated on the figure. All these vectors are then concatenated into orientation histograms by including the contents of all 4x4 sub regions. Here the length of each arrow represents the total sum of the gradient magnitudes closer to that direction in the vicinity of the region. Here we use an orientation histogram having 8 bins around the interest point. Therefore we will get a feature descriptor vector of dimension  $4 \times 4 \times 8 = 128$ .

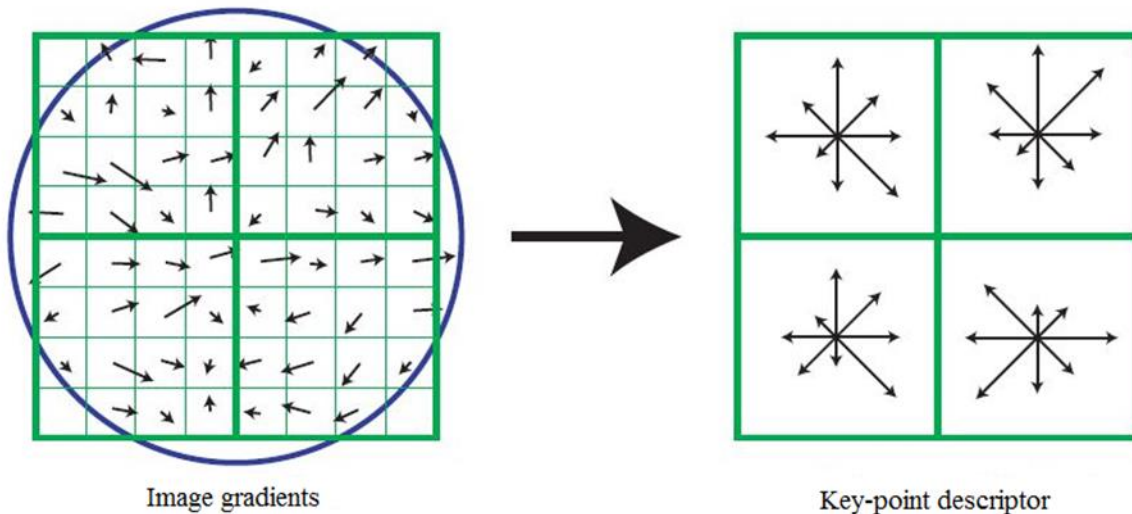


Figure 6 Key-point descriptor for SIFT

### 3.2 Speeded Up Robust Feature Detector (SURF)

SURF is basically derived from SIFT. Like the name indicates for feature extraction, SURF is faster than SIFT which is the main requirement of the today's real time applications. SURF detector used the help of approximated Hessian Matrix [15] to modify the SIFT detector [7]. SURF descriptor uses response of the haar-wavelets within the neighborhood of the interest point for feature description. . Both the detector and descriptor consumes less time because the detector is an approximated version of SIFT and the descriptor is of lower dimension compared with SIFT. So that SURF is better than previously used schemes (SIFT) with respect computational time. Here the increase in speed is achieved at the expense of losing accuracy.

SURF forms an image pyramid without performing 2:1 down sampling for upper levels in the pyramid, so that the resolution of all images will be the same. With the help of box filter approximation of second-order partial derivatives of Gaussian function SURF descriptor filters the stack. This is an advantage with the integral images and it allows to calculate the rectangular box filters in considerably very less time. For interest point matching operation, the nearest neighbor can be defined as the key points whose descriptors are at minimum distance

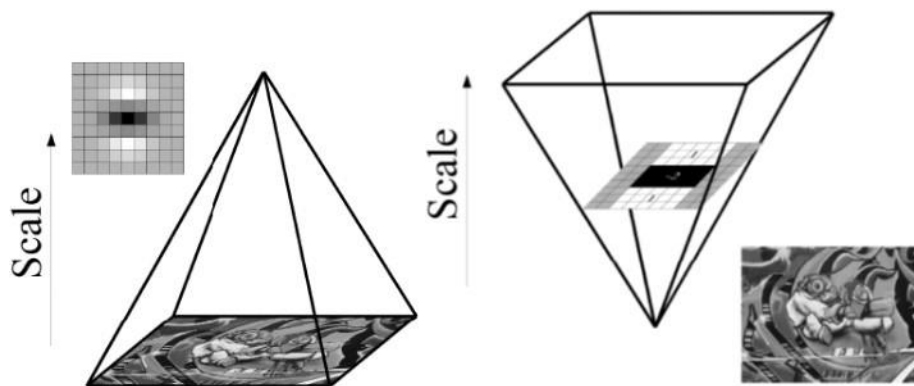


Figure 7 Scale pyramid for computing the key point

### 3.2.1 Fast Hessian Detector

Surf detector uses the help of Hessian metrics to approximate the 2<sup>nd</sup> order approximation of the Gaussian function. This gives good robustness and also ability to find good matches with pair image. Consider the point  $X = (x, y)$  is the given image  $I$ , then the Hessian metrics  $H(x, \sigma)$  for the point  $X$  with the Scale  $\sigma$ , is described as the equation below

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \dots\dots\dots (9)$$

Where  $L_{xx}(x, \sigma)$  is the convolution of the image  $I(x, y)$  and Gaussian second order derivative  $\frac{\partial^2}{\partial x^2} g(\sigma)$  and similarly  $L_{xy}(x, \sigma)$ ,  $L_{yx}(x, \sigma)$  and  $L_{yy}(x, \sigma)$ .

Scale spaces are generally constructed as image pyramids. Here the images are convolved many times with the Gaussian kernel and then sub-sampled in order to make a top level of the pyramid. In SIFT algorithm we used second order derivative of Gaussian function, But we use The  $9 \times 9$  box filters which are the approximations for second order derivatives of Gaussian function.  $D_{xx}$ ,  $D_{xy}$ ,  $D_{yy}$  denotes the approximations of 2<sup>nd</sup> order derivative of Gaussian function.

The weights put into the rectangular areas are kept simple to improve computational easiness and simplicity. Here in the next step we need to normalize the relative weight with the Hessian's determinant.

$$\frac{|L_{xy}(1.2)|_F |D_{xx}(9)|_F}{|L_{xx}(1.2)|_F |D_{xy}(9)|_F} = 0.912\dots \approx 0.9, \quad \text{where } |x|_F \text{ is the Frobenius norm. This gives}$$

$\det(H_{approx}) = D_{xx} D_{yy} - (0.9 D_{xy})^2$ . Because of the use of box filters and integral images, it is not required to use the same filter to the output of a previously filtered layer, but we can use filters of different sizes with good speed directly on the base image.

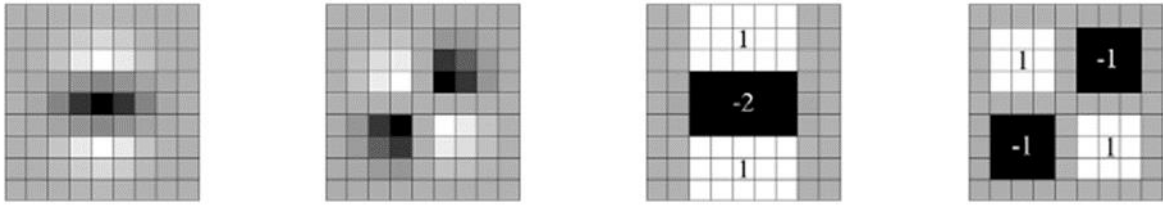


Figure 8 Gaussian second order partial derivatives in y-direction and xy-direction

### 3.2.2 Orientation Assignment

In order to make the descriptor rotation invariant it is required to compute the orientation for the key points. To achieve this goal haar wavelet response in both x and y direction has to calculate for all neighboring pixels inside a circle of radius  $6s$  around the key point. Once the wavelet response is calculated then the next step is to find out the dominant orientation of the candidate point. The dominant orientation is computed by calculating the sum of all responses (haar wavelet response in both x and y direction) inside a rotating window covering an angle of  $\pi/3$ . The responses from the both horizontal and vertical direction inside the window should be added. The sum of these responses then gives a new vector. The lengthiest such vector gives its orientation to the key point.

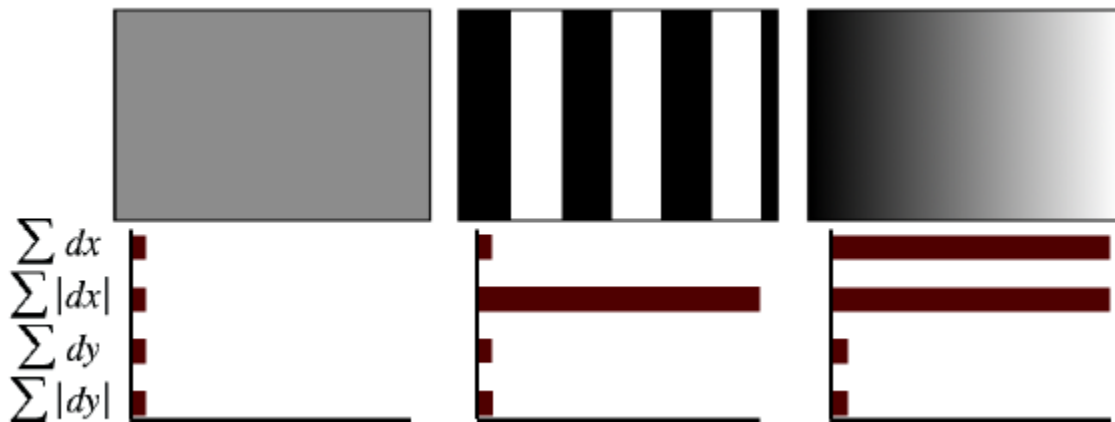


Figure 9 the SURF descriptor

### 3.2.3 Descriptor components

Constructing a square region of size  $20s$  is the next step in extracting the feature descriptor vector. Next we will split this region to  $4 \times 4$  square sub region. For each sub region we need to calculate the response of haar wavelets along both horizontal  $d_x$  and vertical  $d_y$  direction. Then, the wavelet responses  $dx$  and  $dy$  should be added up over each and every sub area and form a first set of elements of the feature descriptor vector. In order to incorporate the information about the gradient of the intensity variations, it is required to calculate the sum of the modulus of the responses,  $|d_x|$  and  $|d_y|$ . Like this, each sub-area has a descriptor vector  $v$  with dimension 4,  $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$ . Similarly all the  $4 \times 4$  areas also have four dimensional vector finally results in SURF descriptor with dimension 64D.

### 3.3 The DAISY Descriptor

DAISY is a dense image feature descriptor. To compute this descriptor the first step is to divide the 360 degree around the interest point into  $H$  parts, each part corresponds an orientation direction.  $G_i$ , where  $1 \leq i \leq H$ , one for each quantized angles, where  $G_o(u, v)$  defined as the normalized gradient of the image at the point  $(u, v)$  for an orientation angle  $o$  if its value is more than zero, if  $o$  is negative then  $G_o(u, v)$  will become equal to zero. This saves the polarity of the gray level intensity variations. Mathematically, orientation maps can be written as  $G_o = \left( \frac{\partial I}{\partial o} \right)^+$ , where  $I$  is the input image,  $o$  is the orientation direction, and  $(.)^+$  is a mathematical operation defined as  $(a)^+ = \max(a, 0)$ .

Each and every orientation map are then repeatedly perform convolution operation with Gaussian kernels of various  $\Sigma$  values to get convolved orientation maps for various sized areas as  $G_o^\Sigma = G_\Sigma * \left( \frac{\partial I}{\partial o} \right)^+$  with  $G_\Sigma$  a Gaussian kernel. A number of  $\Sigma$ s values are used to define the size of the area.

In order to reduce the computational time per descriptor we need to reduce the computational expenses of the convolution operation, and make the convolutions simple. By using separable



functions like Gaussian kernel this target can achieve very easily. In addition to that, it is very easy to calculate the orientation maps for various sizes at small expenses because convolutions operation with a big Gaussian kernel can be performed by repeated convolution operation with a number of smaller kernels. More clearly, if  $G_o^{\Sigma_1}$  is given we can easily calculate  $G_o^{\Sigma_2}$  with  $\Sigma_2 > \Sigma_1$  as

$$G_o^{\Sigma_2} = G_{\Sigma_2} * \left(\frac{\partial I}{\partial o}\right)^+ = G_{\Sigma} * G_{\Sigma_1} * \left(\frac{\partial I}{\partial o}\right)^+ = G_{\Sigma} * G_o^{\Sigma_1}, \dots \dots \dots (10)$$

With  $\Sigma = \sqrt{\Sigma_2^2 - \Sigma_1^2}$  The method to calculate the convolved orientation map of an image can be explained using a figure as shown in Fig. 10. As shown in Fig. 11, at each gray level points, DAISY descriptor comprises a vector formed by values obtained from the convolved orientation maps stayed on concentric circles centered at the candidate point, and where the level of Gaussian smoothing is directly proportional to the radii of the circles.

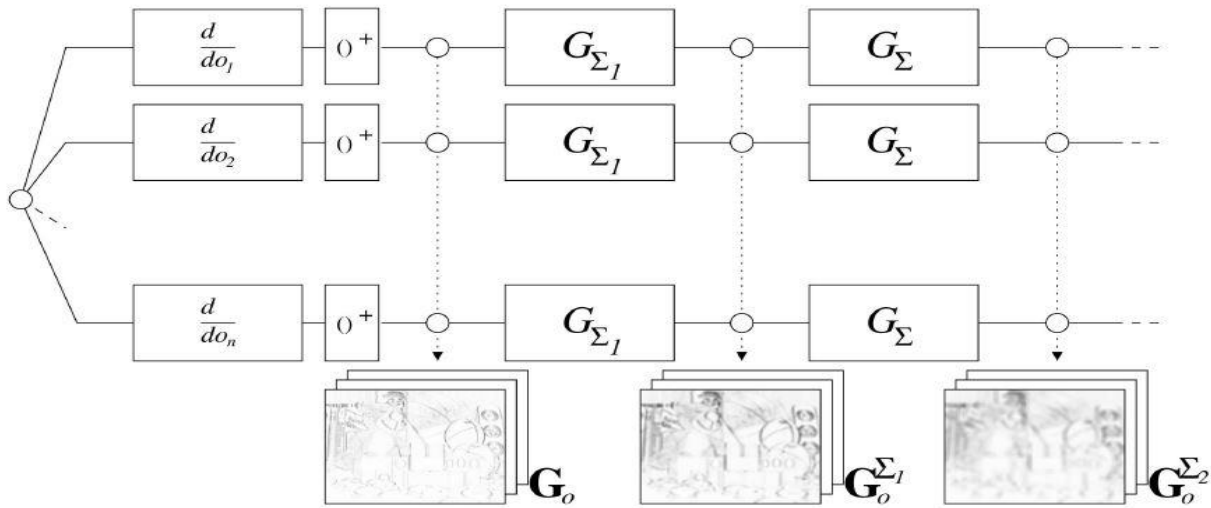


Figure 10 Algorithm to find out DAISY descriptor components

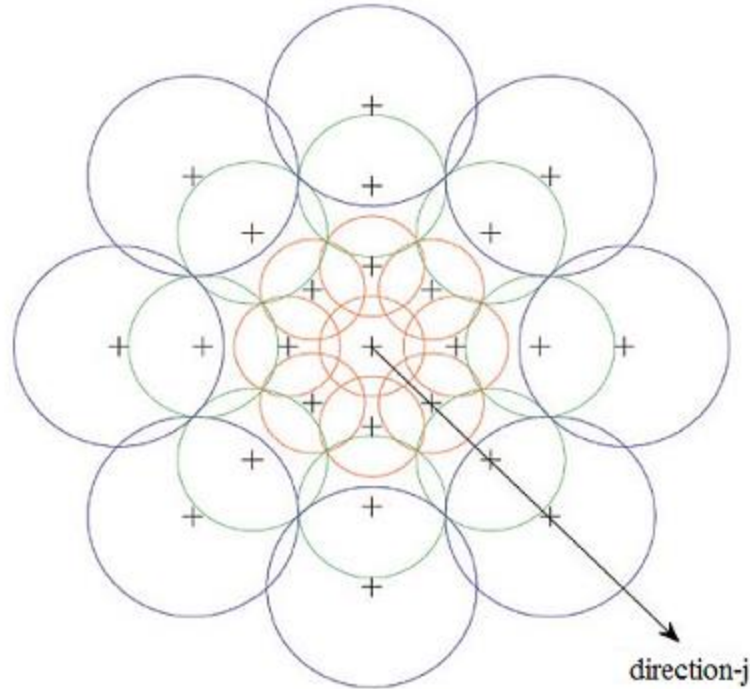


Figure 11 The DAISY descriptor

Let  $h_{\Sigma}(u,v)$  denotes the vector consists of the values at a pixel  $(u, v)$  in the orientation maps after doing convolution operation with a Gaussian kernel function of standard deviation  $\Sigma$ .

$$h(u, v) = [G_1^{\Sigma}(u, v), \dots, G_H^{\Sigma}(u, v)]^T \dots \dots \dots (11)$$

Here  $G_1^{\Sigma}$ ,  $G_2^{\Sigma}$  and  $G_H^{\Sigma}$  represents the  $\Sigma$ -convolved orientation maps in various angles. In the next step normalization of these vectors to unit vector have to perform, and represent the normalized unit vectors as  $h_{\Sigma}(u, v)$ . The normalization operation is done in each and every histogram separately to be able to denote the pixels near occlusions as exact as possible

If Q denotes the total number of various circular layers, then the complete DAISY descriptor D  $(u_0, v_0)$  for particular point  $(u_0, v_0)$  can be represented as the combination of h vectors

$$D(u_0, v_0) =$$

$$[\tilde{h}_{\Sigma_1}^T(u_0, v_0),$$

$$\tilde{h}_{\Sigma_1}^T(I_1(u_0, v_0, R_1)), \dots, \tilde{h}_{\Sigma_T}^T(I_T(u_0, v_0, R_1)),$$

$$\begin{aligned} & \tilde{h}_{\Sigma_2}^T(I_1(u_0, v_0, R_2)), \dots, \tilde{h}_{\Sigma_1}^T(I_T(u_0, v_0, R_2)), \\ & \dots\dots\dots \\ & \tilde{h}_{\Sigma_Q}^T(I_1(u_0, v_0, R_Q)), \dots, \tilde{h}_{\Sigma_1}^T(I_T(u_0, v_0, R_Q))]^T, \end{aligned}$$

where  $I_j(u, v, R)$  is the point with distance  $R$  from the point  $(u, v)$  in the angle given by  $j$  when the angles are quantized into the  $T$  values of Table 1.

### 3.4 FAST: Features from Accelerated Segment Test

As the name suggest FAST is a one of the fastest feature detector algorithm so far. But here the good speed has been achieved at expense of accuracy. Here the key-point detection is performed by a test on a circle of sixteen points centered at the interest point ‘p’. The basic detector confirms ‘p’ as an interest point if there found a set of ‘n’ continuous points without any gap in the circle whose intensity values are more than the intensity value of the key-point plus a threshold  $t$ , or less than the intensity value of the key-point minus a threshold  $t$ , as shown in Figure 12. The value of ‘n’ is fixed as twelve as it allows a checking with very good speed and also this will be very much useful in eliminating false interest-points: the test considers four points at 1, 5, 9 and 13 (the four compass directions) only. If  $p$  is an interest point then at least three continuous pixels around the circle of 16 pixels must have intensity value more than the intensity level of candidate point plus a threshold value  $t$  or less than the intensity level of candidate point minus a threshold value  $t$ . If this condition is not satisfied, then  $p$  cannot be an interest point. The full segment test algorithm can then be used at the rest of the points by checking all points in the circle.

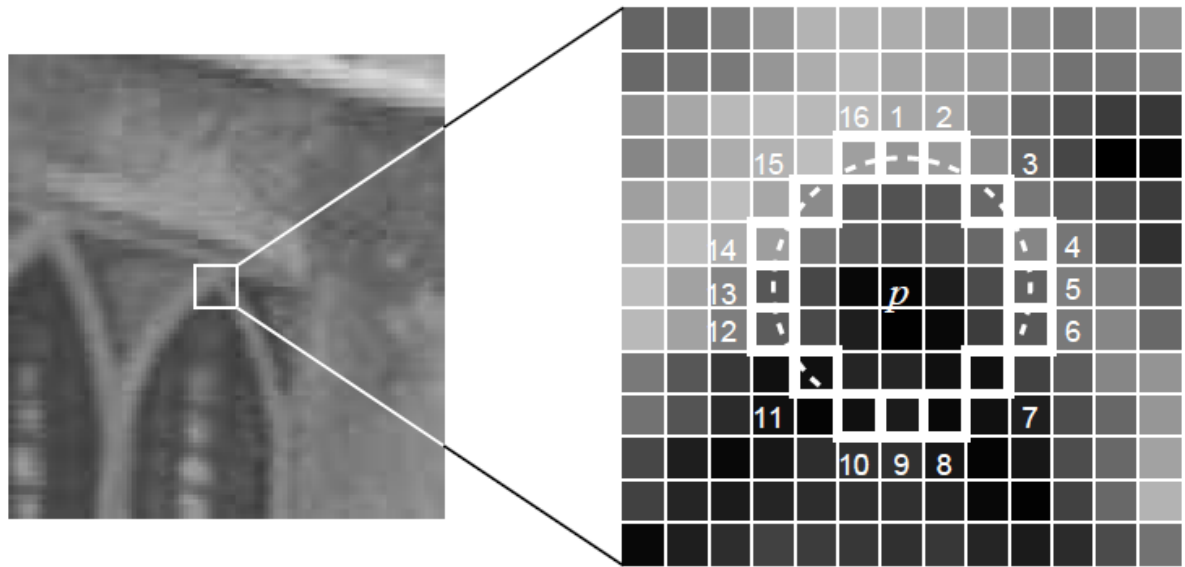


Figure 12 point segment test corner detection in an image patch

### 3.5 BRISK: Binary Robust Invariant Scalable Key points

#### 3.5.1 Scale Space Key point Detection

Achieving scale invariance is a most important thing for a high quality key point. For this purpose BRISK Descriptor go a step forward by checking for the maxima or minima point in both image plane and the scale-space plane using the FAST score 's' as a measure for robustness. To find the true scale point most of the detectors discretize the scale axis at coarser intervals, But the BRISK detector estimates the actual scale of all key point in the continuous scale-space plane.

In this detector algorithm, the scale-space stack layers comprise of  $n$  octaves  $c_i$  and  $n$  intra-octaves  $d_i$ , for  $i = \{0, 1 \dots n - 1\}$  and generally  $n = 4$ . The octaves are created from the repeated half-sampling the base image (represented as  $c_0$ ). Each intra-octave  $d_i$  is placed in-between 2 adjacent octave layers (as shown in Figure 13). The first intra-octave  $d_0$  is formed by down sampling the base image  $c_0$  by a scaling factor of 1.5, and the remaining intra-octave layers are formed by repeated down sampling by a factor 2. Therefore, if 't' represents scale then  $t(c_i) = 2^i$  and  $t(d_i) = 2^i \cdot 1.5$ .

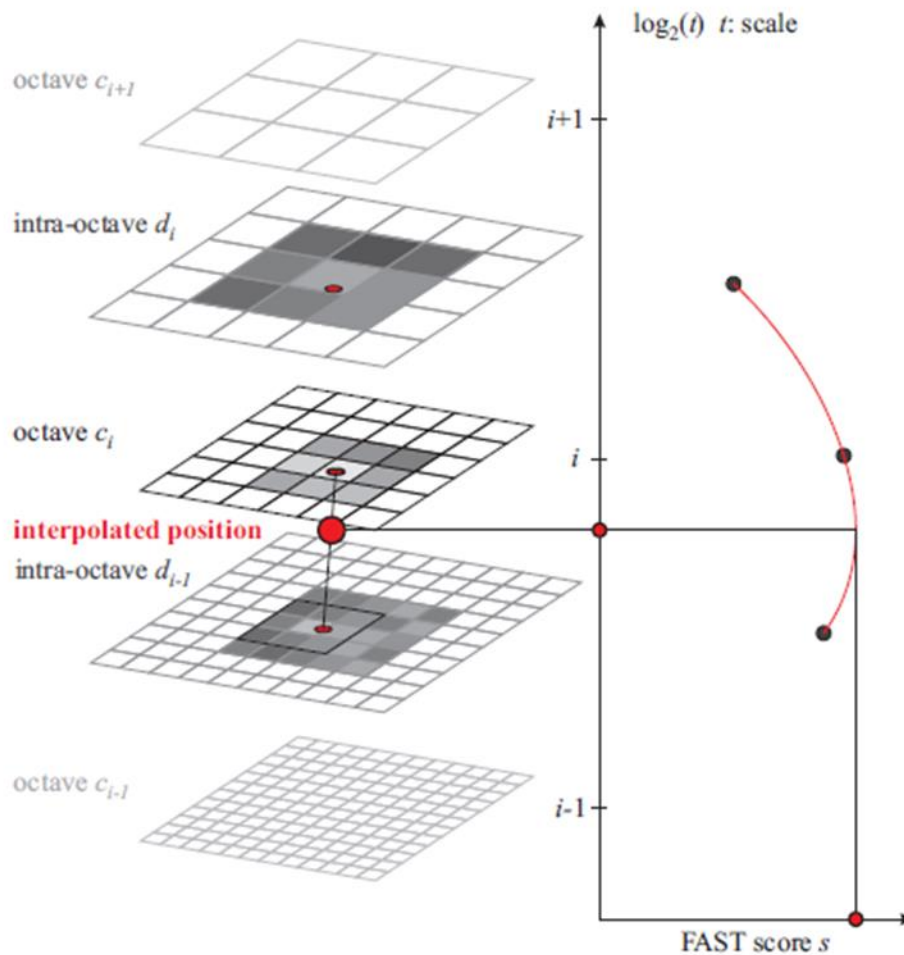


Figure 13 Scale-space interest point detection in BRISK detector

In BRISK detector, 9-16 masks are mostly used, which demands at least minimum 9 continuous points in the 16-point circle centered at the candidate point which are of intensity values more than the intensity value of the candidate point plus threshold  $t$  or less than the intensity value of the candidate point minus threshold  $t$  for the FAST condition to be satisfied.

In the initial stage, the FAST 9-16 detector is acted on each and every octave and intra-octave individually with the same threshold 'T' to find out the key-point of interest. Then, the candidate point being referred here required to satisfy the maximum condition as for its 8 neighboring FAST scores 's' in the current layer. The score 's' is nothing but the maximum value of threshold still accepting an image candidate point as a corner. Also, the scores in the top and beneath layer will must be lower too.

Taking image saliency as a continuous variable over the image as well as along the scale space, we perform a sub-pixel and ceaseless scale refinement for every identified maximum. To find out the actual scale of the interest point which is detected a 1D parabola is fitted at the scale space plane axis. As a last step, re-interpolation of the image coordinates between the patches in the layers next to the determined scale are performed.

### 3.5.2 Key point Description

For a number of key points the BRISK descriptor is made as a binary string by combining the outputs of simple intensity level comparison tests. In BRISK, we find out the orientation direction of each and every key point for design an orientation-normalized descriptors so that the descriptor becomes rotation invariant which is the most important attraction for a descriptor.

### 3.5.3 Sampling Pattern and Rotation Estimation.

The method used for sample the neighboring elements of the candidate point is the key concept behind BRISK descriptor. The pattern, described in Figure 3, describes N distinct places equally separated on circles centered at the key point. For avoiding the effects of aliasing when sampling the intensity levels of an image at a point  $p_i$  in the image, we used Gaussian smoothing filter with standard deviation  $\sigma_i$  directly proportional to the distance between the 2 points on the respective circle. If we have N sample points then we will be having N. (N-1)/2 number of sampling point pairs. Consider a sampling point pairs  $(p_i, p_j)$  and  $I(p_i, \sigma_i)$  and  $I(p_j, \sigma_j)$ , be the Gaussian smoothed version of these two sampling points. Local gradient  $g(p_i, p_j)$  can be calculated using these sampling point pairs and it's Gaussian smoothed version using this expression.

$$g(p_i, p_j) = p_j - p_i \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \dots\dots\dots (12)$$

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(p_i, p_j) \in L} g(p_i, p_j) \dots\dots\dots (14)$$

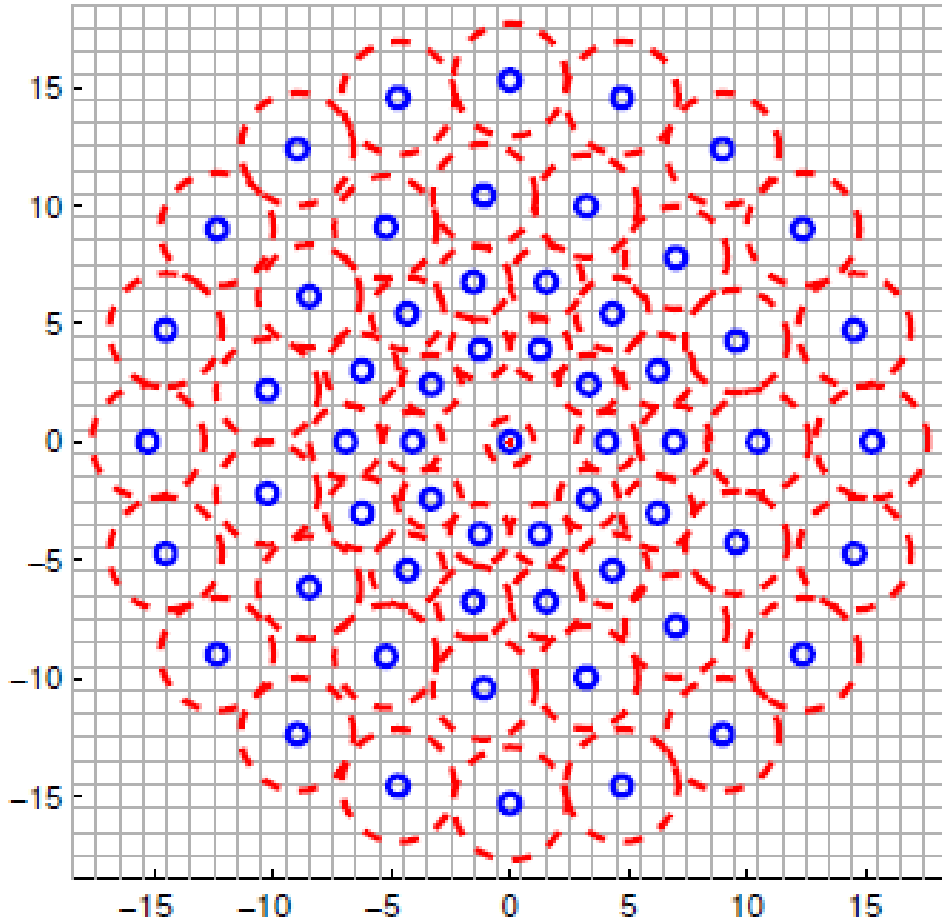


Figure 14 The BRISK sampling pattern

### 3.5.4 Building the Descriptor

To design a rotation- and scale-invariant feature descriptor, BRISK descriptor uses a sampling pattern rotated by  $\alpha = \arctan2(g_y, g_x)$  around the interest point  $k$ . The bit stream -vector descriptor  $d_k$  is created by calculating all the short distance intensity differences of sample point pairs  $(p_i^\alpha, p_j^\alpha) \in S$  (i.e. in the rotated pattern), here the component of the binary string bit ‘b’ is defined as:

$$b = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases} \dots\dots\dots (15)$$

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

### 3.5.5 Descriptor Matching

Matching between two BRISK descriptor vectors can be established by a simple calculation of their Hamming distance. The similarity between 2 descriptors can be measured by counting the total number of bits which are same for both of them. This operation can be implemented easily by doing the bitwise XOR operation followed by counting the number of zeros.

### 3.6 Performance Comparison of descriptor

In the feature based face image registration process the first step is feature detection. The feature detection is the most important step in registration. The ideal qualities required for the feature detector are scale invariance, rotation and noise immunity. The feature detector must be capable of extracting the same feature in both reference and detector image even if the detected image undergoes some deformations. The basic idea behind the matching technique is the computation of distance between the feature vectors. The descriptor vector which gives minimum neighborhood distance is the matched pair. But sometimes due to the limitations with the descriptor vector we may get more than one point in reference image which are at the same distance from a point in detected image. This type of error will result in feature mismatch. Because these are all real-time operations, the computational time is also one of the matters of concern in feature extraction.

Here in this work we studied about different feature detectors and descriptors and their performances also compared. Computational speed, repeatability, recall and precision are the parameters used here for comparison. Here we follow the evaluation criteria of Mikolajczyk et al. [1] which is based on the number of correct and incorrect matches for a given image pair. We did experiments on Japanese Female Facial Expression (JAFFE) database for performing the simulations. This database contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. We kept the face image without any expression as reference and with expressions as sensed image and tried for matching using different feature extraction methods. Our comparison parameters computational speed, repeatability, recall and precision are explained below.



Repeatability score.

The repeatability score is defined as the ratio of number of key points in reference image and the number of correspondent points in the sensed image.  $SA = R^+/R^*$ , with  $R^+$  the number of correspondences and  $R^*$  the number of all detected features in a reference image.

We made some detector-descriptor combinations using some scale and rotation invariant detectors. They are SIFT, SURF, DAISY, FAST, BRISK and our proposed modified BRISK detector. The comparison of performances when they are applying with face images are shown below.

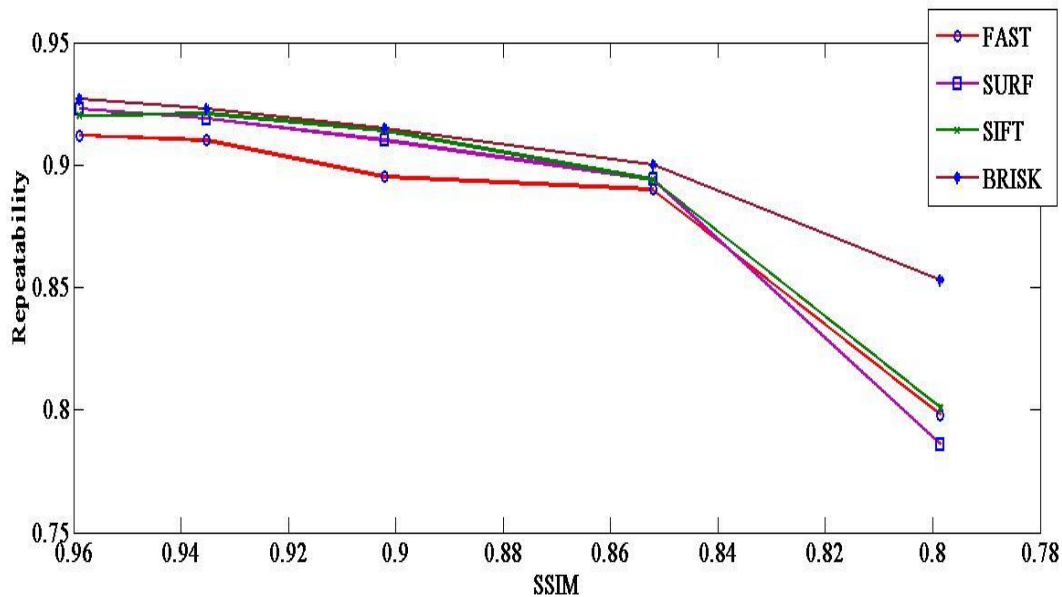


Figure 15 Repeatability vs SSIM plot of various descriptor

As explained above repeatability is a measure of the number of points in the reference image which is repeating in the detected image. For the perfect matching and good transformation function number of matching points should be high. That means repeatability should be high. From this graph it is observed that BRISK is the best descriptor in terms of repeatability.

<b>Detector/Descriptor</b>	<b>SIFT</b>	<b>SURF</b>	<b>FAST</b>	<b>DAISY</b>	<b>BRISK</b>
<b>Avg Features</b>	478	376	516	n.a	<b>647</b>
<b>Detector ms/image</b>	1.210	0.640	0.097	n.a	<b>0.45</b>
<b>Descriptor ms/feature</b>	1.08	0.25	n.a	0.33	<b>0.061</b>
<b>Storage bytes/feature</b>	128	64	n.a		<b>64</b>

Table 1 Comparison table for different detector/descriptor

This table shows the BRISK detector-descriptor outperforms the other detector-descriptor in all respect. Here number of key-points detected, time per detector, and storage bytes are the things compared everywhere BRISK gives the best results. But lot of confusions came during matches with BRISK detector also. So we need a better descriptor. The next chapter is discussing about proposed modified BRISK descriptor.

# CHAPTER 4

## 4. Proposed Modified BRISK descriptor

### 4.1 Modified BRISK Descriptor

In original BRISK descriptor features are described by 1 single bit stream. Here the matching process is done by calculating the hamming distance by doing XOR operation. Then will find the 2 correspondence points in detected image which is at minimum distance from the reference image. In the next step ratio between these distances will calculate. If the ratio is more than 0.8 this points will be eliminated and it will not use for matching purposes. So for transform estimation the correspondence between these points will not considers. The information loss here can degrade the quality of the registered output. So to increase the total number matches one algorithm is proposed in this work. The algorithm proposed is shown below.

The original BRISK descriptor is a binary string made by concatenating the results of simple brightness comparison tests. Here the brightness comparison is doing in such a way that if the intensity value at the point  $p_i$  is greater than the intensity value at point  $p_j$  the value will be 1 else it is zero. Like this the comparison will do for  $N(N-1)/2$  sampling pairs ( $p_i, p_j$ ). Because this method threshold at exactly the intensity value of the candidate point  $p_i$ , this descriptor may become highly to be sensitive to noise, particularly in near-uniform image regions. We know that most of the facial regions are relatively uniform, it is very much useful to improve the efficiency and quality of the underlying descriptors in these areas.

This section extended binary code of BRISK descriptor to 3-value codes in which if the difference value between  $I(p_i, \sigma_i)$  and  $I(p_j, \sigma_j)$  is coming in a zone of width  $\pm t$  are quantized to zero, ones above this are quantized to +1 and ones below it to -1,

i.e. the indicator  $b$  is replaced by a ternary function:

$$b' = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) \geq I(p_i^\alpha, \sigma_i) + t \\ 0 & |I(p_j^\alpha, \sigma_j) - I(p_i^\alpha, \sigma_i)| < t \dots\dots\dots (16) \\ -1 & I(p_j^\alpha, \sigma_j) \leq I(p_i^\alpha, \sigma_i) - t \end{cases}$$

Here  $t$  is a user-defined threshold. This logic will give us a ternary pattern. For simplicity we will split this 3- value code into 2 binary codes. In the first (upper) binary code we will code '1' as 1 and '0' and '-1' as '0'. In the second (lower) stage we will code '1' and '0' as '0', '-1' as '1'. By using this logic we will 2 bit streams for representing a single key point. Subsequently treating these as two separate channels we will do matching by computing the hamming distance. The upper bit stream will compare with upper and the lower will compare with lower and the summation will be calculated. The minimum distance corresponds to the matching points.

Compared with the original BRISK descriptor, the modified BRISK descriptor will consume double the memory and more computation time. The experimental results shows for the modified BRISK descriptor that the accuracy in matching has increased significantly compared with its original version.

## 4.2 Proposed Algorithm to find out the possible reference image pair for face registration

The last 2 chapters discuss about different feature descriptors and image registration algorithms based on that. As I explained in the introduction chapter, the main problem we want to solve in this work is, how to make a pose invariant face recognition algorithm. By using image registration as a preprocessing step, pose invariant face recognition algorithm can be developed. Logically it is very much clear that this idea can improve the detection accuracy as registration will transform the sensed image to the coordinates of the reference image. In the practical case in the train database a number of face images will be there and detected image can be only one at a moment. So which image in the data base will be a reference image to the newly detected image? So here the only solution is to find out the possible face images in the database which could be the reference image to the detected image and register the detected image with all the suspected reference images.

The algorithm what we developed is described below.

**Step1.** Calculate complete feature descriptor vector for all train images in the database

**Step2.** Compute complete feature descriptor vector for the sensed image

**Step3.** Compute the ratio between number of matched key-points and the total number of key-point

**Step4.** Take two images in the train set whose corresponding feature vector gives the maximum ratio of the newly sensed image.

**Step5.** Do registration for both images and compute the output image.

**Step6.** feed the output image to a face recognition algorithm.

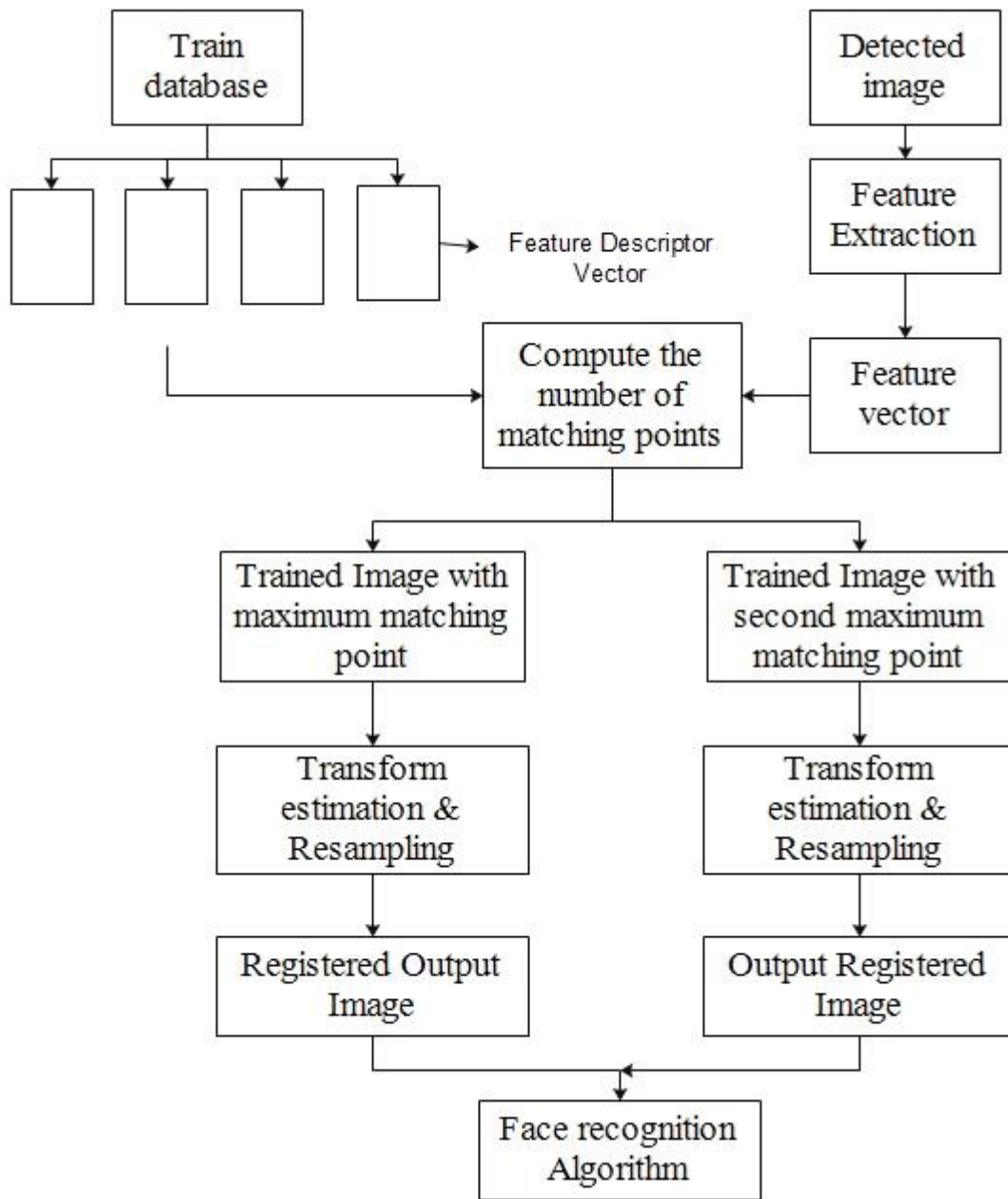


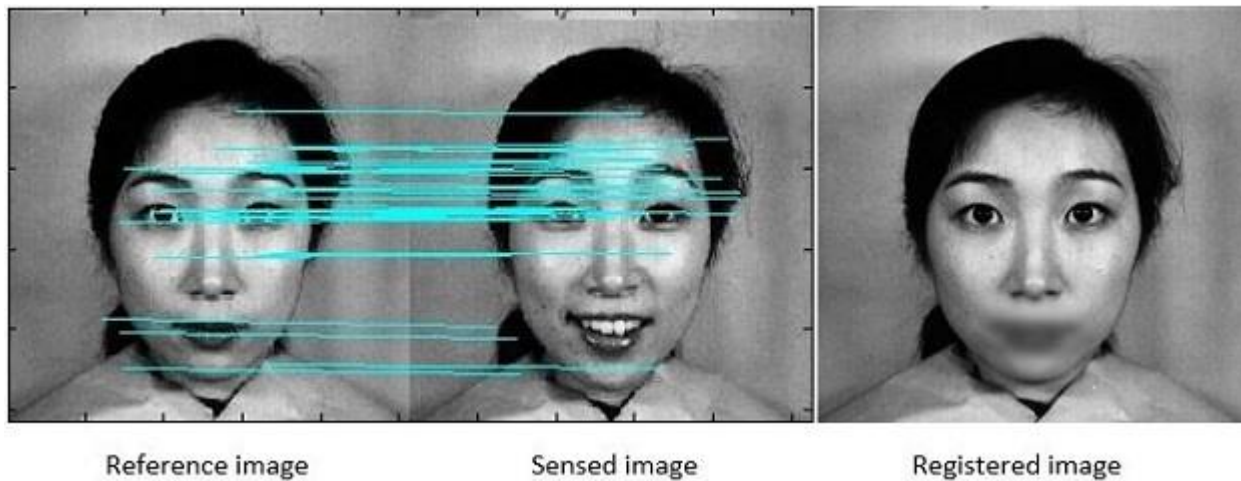
Figure 16 Proposed algorithm for feature based face image registration

### 4.3 Output of registration algorithm with different feature extractors

In this work different combination of feature detectors and descriptors have been made. After the transform estimation and image resampling registered image output is obtained. For comparing the similarity between the reference and registered image we used SSIM index and mutual information concept.

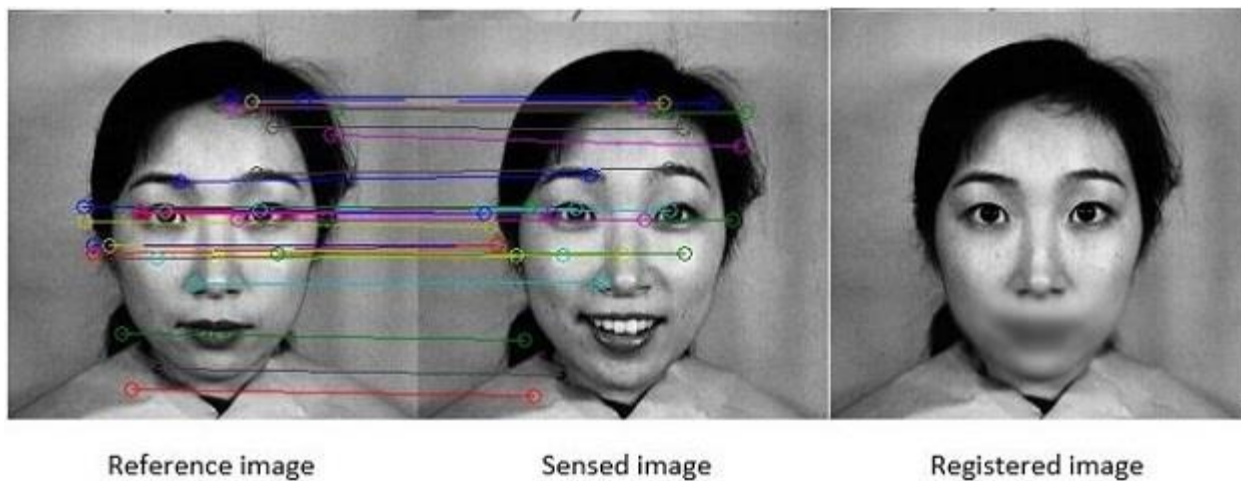
The output of feature based face image registration is shown below

#### SIFT-SIFT



The important matched features are showing in the figure. In between the detected and sensed images where the mismatch occurs, for those regions we got blur.

#### SURF-SURF



SURF is an approximated version of SIFT so it takes less computational time and gives good accuracy

### FAST-DAISY



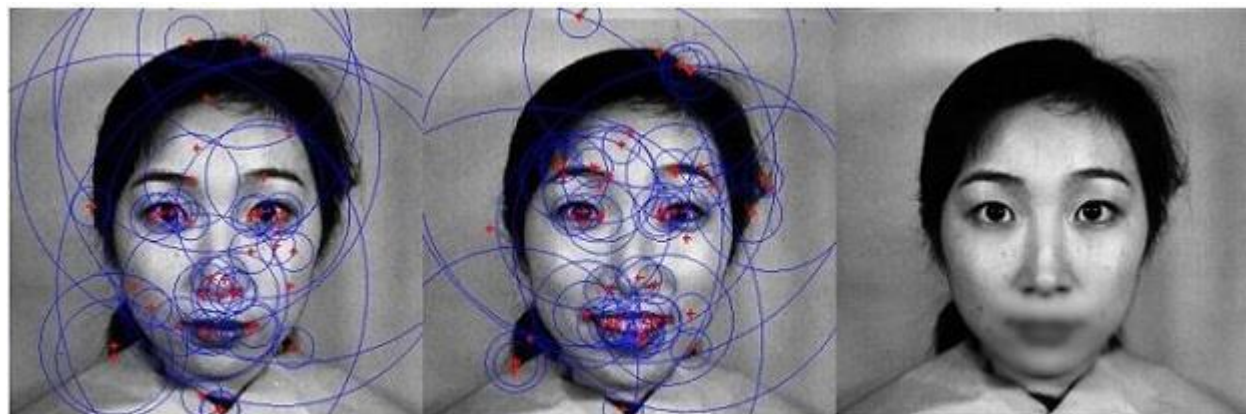
Reference image

Sensed image

Registered image

FAST is a very much fast detector but it is not robust like other descriptor. And the combination of FAST and DAISY also not giving proper results.

### BRISK-DAISY



Reference image

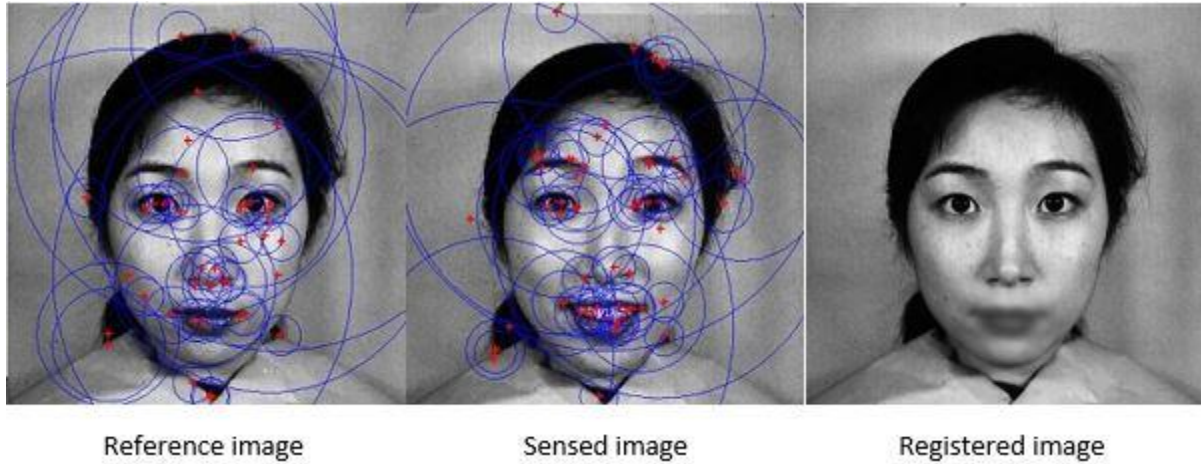
Sensed image

Registered image

Here the red dots are denotes the key-points. Blue circles are sampling regions

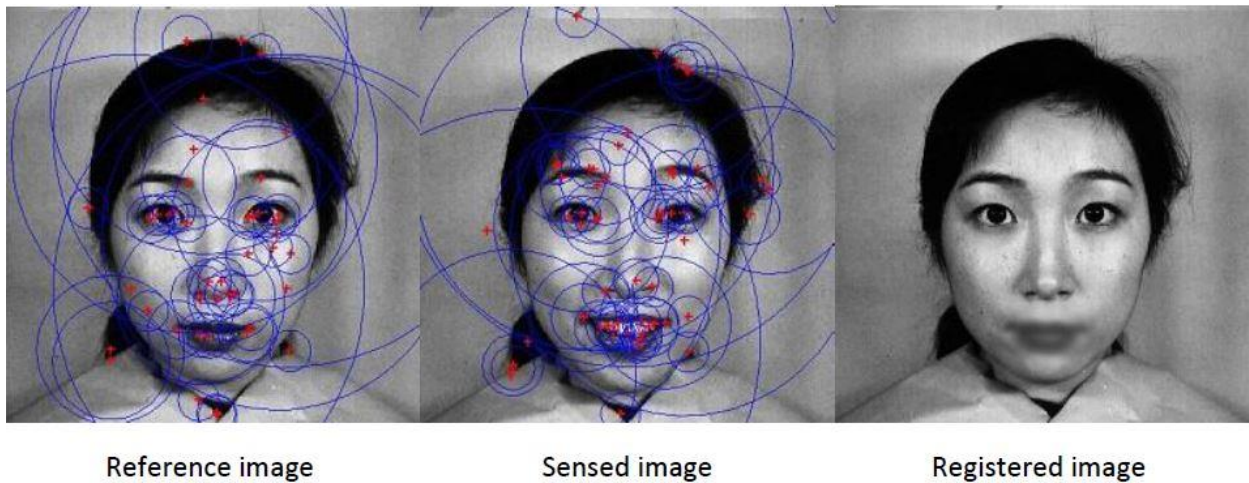


## BRISK-BRISK



The combination of BRISK descriptor with BRISK detector gives good results. But the total number of key-points discarded due to matching criteria [16] is significantly high here.

## BRISK-modified BRISK



Among all this combination give maximum good results

Detector-Descriptor combination	SSIM Index	Mutual Information
<b>SIFT-SIFT</b>	0.9803	1.79
<b>SURF-SURF</b>	0.9717	1.63
<b>FAST-DAISY</b>	0.9684	1.58
<b>BRISK-DAISY</b>	0.9831	1.91
<b>BRISK-BRISK</b>	0.9849	1.98
<b>BRISK-modified BRISK</b>	<b>0.9884</b>	<b>2.02</b>

Table 2 Performance analysis of different feature based face registration

Output of the another set of face registration is shown below



Reference Image

Sensed Image



FAST- DAISY

SURF-SURF



SIFT-SIFT

BRISK-DAISY



BRISK-BRISK

BRISK-modified BRISK

Detector-Descriptor combination	SSIM Index	Mutual Information
<b>SIFT-SIFT</b>	0.8754	1.33
<b>SURF-SURF</b>	0.8623	1.31
<b>FAST-DAISY</b>	0.8317	1.26
<b>BRISK-DAISY</b>	0.8963	1.37
<b>BRISK-BRISK</b>	0.9081	1.39
<b>BRISK-modified BRISK</b>	0.9130	1.42

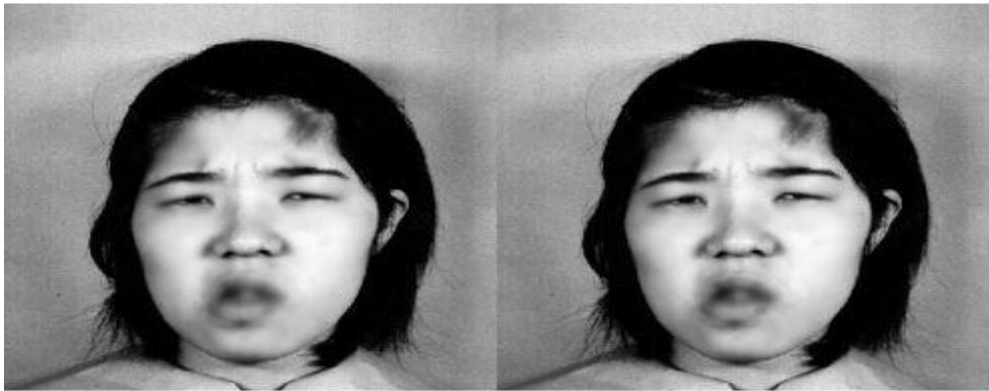
*Table 1 Performance analysis of different feature based face registration*

Output of the another set of face registration is shown below



Reference Image

Detected Image



FAST-DAISY

SURF-SURF



SIFT-SIFT

BRISK-DAISY



BRISK-BRISK

BRISK-modified BRISK

Detector-Descriptor combination	SSIM Index	Mutual Information
<b>SIFT-SIFT</b>	0.8664	2.13
<b>SURF-SURF</b>	0.8492	2.11
<b>FAST-DAISY</b>	0.8335	2.06
<b>BRISK-DAISY</b>	0.8812	2.17
<b>BRISK-BRISK</b>	0.8903	2.19
<b>BRISK-modified BRISK</b>	0.9086	2.22

Table 4 Performance analysis of different feature based face registration

Here for analyzing the performance of different feature based face registration algorithms we used 2 parameters. SSIM and Mutual Information. Basically both are measuring the similarity between 2 images. SSIM is more concentrating about the structure of the images. MI is not considering the pixel to pixel relationship, and it is seeing the gray level value correspondence. So SSIM is the best method. Among all descriptors, from this comparison it is observed that BRISK-modified BRISK combination is the best combination.

The head pose database consist of 15 sets of images of different poses. The head orientation and poses of the face images are different in each and every images. We tried to perform registration operation with this data bases also. The average structural similarity index between the reference and sensed image, reference and registered image for each set of images with different detector descriptor combinations are calculated and compared. The results of comparison is shown in the table below.

	ssim	SIFT-SIFT ssim	BRISK- DAISY ssim	BRISK- BRISK ssim	BRISK- modified BRISK ssim
	Reference and sensed	Reference & output	Reference & output	Reference & output	Reference & output
Person 01	0.6832	0.7539	0.7629	0.7826	<b>0.7912</b>
Person 02	0.6352	0.7289	0.7326	0.7629	<b>0.7752</b>
Person 03	0.6531	0.7436	0.7538	0.7769	<b>0.7836</b>
Person 04	0.6956	0.7863	0.7926	0.8169	<b>0.8265</b>
Person 05	0.7025	0.7758	0.7923	0.8235	<b>0.8365</b>
Person 06	0.7325	0.7963	0.8026	0.8321	<b>0.8462</b>
Person 07	0.6725	0.7269	0.7324	0.7598	<b>0.7635</b>
Person 08	0.6485	0.7362	0.7423	0.7632	<b>0.7796</b>
Person 09	0.6215	0.7263	0.7395	0.7426	<b>0.7592</b>
Person 10	0.6435	0.7326	0.7463	0.7632	<b>0.7795</b>
Person 11	0.7235	0.7465	0.7596	0.7765	<b>0.7896</b>
Person 12	0.6952	0.7638	0.7782	0.7963	<b>0.8036</b>
Person 13	0.7149	0.7769	0.7896	0.7935	<b>0.8129</b>
Person 14	0.6635	0.7596	0.7695	0.7863	<b>0.7936</b>
Person 15	0.6836	0.7756	0.7836	0.7926	<b>0.8014</b>

*Table 5 comparison of registration algorithms using different feature detector-descriptor combination using head pose database*

This table shows that the conclusion obtained from JAFFE database is correct. That means BRISK-modified BRISK detector descriptor combination outperforms all other combinations in terms of quality of the registered output (similarity between the reference and registered output is maximum in the case of BRISK-modified BRISK combination).

## CHAPTER 5

### 5. Conclusion and future scope

Pose invariant face recognition algorithms have a big demand now a days. Because pose invariance is the most desirable quality for an automatic face recognition algorithm. This work suggests that using image registration as a preprocessing step could be a best step for achieving pose invariance. Illumination changes are one of the problem for face recognition. So here intensity based registration algorithms are completely avoided. Features are less affected by illumination and contrast variation. So here feature based registration algorithms are preferred. Here scale and rotation invariant detector-descriptors like SIFT, SURF, FAST, DAISY, and BRISK are used. Using this detector descriptor combination correspondence between the points of reference and detected images has been established. A good detector should be able to find out the same key-points in both sensed and reference image even if it undergoes some deformation. Repeatability is the parameter which measures number of features which is common to both images. Like this we compared all detectors and observed that BRISK descriptor outperforms all detectors in terms of computational accuracy and less complexity. But in this descriptor also matching all common points are difficult. So a new modified BRISK descriptor has been proposed in this work. The similarity between registered output and reference image has been estimated using SSIM and mutual information. Here BRISK-modified BRISK detector combination gives the best result. An algorithm is also proposed in this work to find out the possible pairs in the reference image corresponds to the detected image.

#### **Future Work**

In this work face images with front poses which under gone local deformations are considered for image registration. We can extend this work using more number of reference images which is taken at different view angles so that the face image of a person with different view angle also can be recognized.

## References

- [1]. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [2]. M. Bartlett, J. Movellan, and T. Sejnowski, "Face recognition by independent component analysis," *Neural Networks, IEEE Transactions on*, vol. 13, no. 6, pp. 1450–1464, nov 2002.
- [3]. J. Lu, K. Plataniotis, and A. Venetsanopoulos, "Face recognition using lda-based algorithms," *Neural Networks, IEEE Transactions on*, vol. 14, no. 1, pp. 195–200, jan 2003.
- [4]. Di Huang, Caifeng Shan, Yunhong Wang, and Liming Chen, "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey" *IEEE Transactions on systems, man and cybernetics—Part C: applications and reviews*, Vol. 41, No. 6, november 2011
- [5]. Brown, L.G, "A Survey Of Image Registration Techniques," *ACM Computing Surveys(CSUR)*, pp-113-119,2010.
- [6]. Barbara Zitova, Jan Flusser, "Image Registration Methods A Survey," *Image and Vision Computing*, pp-997-1000, 2003.
- [7]. Lowe, D.: Distinctive image features from scale-invariant keypoints, cascade filtering approach. *IJCV* 60 (2004) 91 – 110
- [8]. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded Up Robust Features" Elsevier preprint, 2008.
- [9]. Edward Rosten and Tom Drummond "Machine learning for high-speed corner detection" *ECCV 2006, Part I, LNCS 3951*, pp. 430–443, 2006.©Springer-Verlag Berlin Heidelberg 2006
- [10]. Engin Tola, Vincent Lepetit, and Pascal Fua "DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo" *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 32, No. 5, May 2010
- [11]. Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart "BRISK: Binary Robust Invariant Scalable Keypoints" 2011 *IEEE International Conference on Computer Vision* 978-1-4577-1102-2/11/\$26.00 © 2011 IEEE



- [12]. M. Ehlers, D.N. Fogel, High-precision geometric correction of airborne remote sensing revisited: the multiquadric interpolation, Proceedings of SPIE: Image and Signal Processing for Remote Sensing 2315 (1994) 814–824.
- [13]. Daniel Russakoff, Carlo Tomasi, Torsten Rohlfing, Calvin R. “Image similarity using mutual information of regions” 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III pp 596-607
- [14]. Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity” IEEE Transactions on Image Processing, Vol. 13, No. 4, April 2004
- [15]. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: ICCV. Volume 1. (2001) 525 – 531