

# Comparision of Iris Identification by Using modified SIFT and SURF keypoint Descriptor

*A Thesis submitted in partial fulfillment of the requirements for the degree of  
Bachelor of Technology in “Electronics and Instrumentation Engineering”*

By

**SharadAgarwal (109EI0316)**

**Ritesh Kumar Nayak (109EI0166)**

Under the Esteemed guidance of

**Prof. U. C. Pati**



Department of Electronics and Communication Engineering

**National Institute of Technology**

**Rourkela-769008 (ODISHA)**

# Comparision of Iris Identification by Using modified SIFT and SURF keypoint Descriptor

*A Thesis submitted in partial fulfillment of the requirements for the degree of  
Bachelor of Technology in “Electronics and Instrumentation Engineering”*

By

**SharadAgarwal (109EI0316)**

**Ritesh Kumar Nayak (109EI0166)**

Under the Esteemed guidance of

**Prof. U. C. Pati**



Department of Electronics and Communication Engineering

**National Institute of Technology**

**Rourkela-769008 (ODISHA)**



DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

ODISHA, INDIA-769008

---

# CERTIFICATE

---

This is to certify that the thesis entitled “**Comparison of Iris Identification by Using modified SIFT and SURF keypoint Descriptor**”, submitted by **SharadAgarwal (Roll No: 109EI0316)** and **Ritesh Kumar Nayak (Roll No: 109EI0166)**, in partial fulfillment of the requirements for the award of **Bachelor of Technology in Electronics and Instrumentation Engineering** during session 2012-2013 at National Institute of Technology, Rourkela is abonafide record of research work carried out by them under my supervision and guidance. The candidates have fulfilled all the prescribed requirements. The Thesis which is based on candidates’ own work has not been submitted elsewhere for a degree/diploma. In my opinion, the thesis is of standard required for the award of a bachelor of technology degree in Electronics and Instrumentation Engineering.

**Place: Rourkela**

**Prof. U. C. Pati**

**Date : 13th May, 2013**

**Dept. of Electronics and Communication Engineering**

**National institute of Technology, Rourkela**

## ACKNOWLEDGEMENTS

I would like to thank NIT Rourkela for giving me the opportunity to use their resources and work in such a challenging environment. My heart pulsates with the thrill for tendering gratitude to those persons who helped me in completion of the project.

We would like to take this opportunity to express our gratitude and sincere thanks to our respected supervisor **Prof. U. C. PATI** for his guidance, insight, and support he has provided throughout the course of this work. We would like to articulate our profound gratitude and indebtedness to **Prof. L. P. ROY** for teaching “Digital Image Processing” in such depth and setting up a strong fundamental base in Image Processing that enabled us achieve this success. We would like to thank all faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T.Rourkela for their extreme help throughout course.

An assemblage of this nature could never have been attempted without reference to and inspiration from the works of others whose details are mentioned in the reference section. We acknowledge our indebtedness to all of them.

Last but not the least our sincere thanks to all our friends, who have patiently extended all sorts of help for accomplishing this undertaking.

Date: 13th May 2013

NIT Rourkela

**SHARAD AGARWAL (109EI0316)**

**RITESH KUMAR NAYAK (109EI0166)**

**Dedicated  
To  
My beloved parents**

## ABSTRACT

A wide variety of systems require reliable personal recognition schemes to either confirm or determine the identity of an individual requesting their services. The purpose of such schemes is to ensure that only a legitimate user, access the rendered service. A biometrics system is essentially a pattern recognition system, which makes a personal identification by determining the authenticity of a specific physiological or behavioral characteristic possessed by the user. Iris serves as one of the excellent biometric traits due to the stability and randomness of its unique features. After localization of the iris, *Scale Invariant Feature Transform* (SIFT) is used to extract the local features. But SIFT is found out to be computational complex. So in this paper another keypoint descriptor, *Speeded Up Robust Features* (SURF), is tested and then modified which compare the performance of different descriptor and hence gives promising results with very less computations. We finally carry out a comparison of both the descriptors performance wise.

# CONTENTS

ACKNOWLEDGEMENTS .....	a
ABSTRACT.....	i
CONTENTS.....	ii
<b>INTRODUCTION</b> .....	1
<b>1.1 Background on Biometrics</b> .....	2
<b>1.2 Human Iris as preferred Biometrics</b> .....	4
<b>1.3 Objective</b> .....	5
<b>1.4 Thesis Organization</b> .....	5
<b>LITERARTURE REVIEW</b> .....	7
<b>2.1 Literature Review</b> .....	8
<b>IRIS LOCALISATION</b> .....	9
<b>3.1 Image Acquisition</b> .....	10
<b>3.2 Image Preprocessing</b> .....	10
3.2.1 Removal of Specular Highlights.....	10
3.2.2 Pupil Detection.....	12
3.2.3 Iris Detection.....	13
3.2.4 Removal of noise due to eye-lid occlusion .....	15
<b>FEATURE EXTRACTION (SIFT)</b> .....	16
<b>4.1 Scale Invariant Feature Transform</b> .....	17
4.1.1 Constructing a Scale Space.....	17
4.1.2 Laplacian of Gaussian (LoG) Approcimation.....	19
4.1.3 Finding Keypoints.....	20
4.1.4 Getting Rid of Bad Keypoints.....	22

4.1.5	Orientation Assignment .....	24
4.1.6	Generating SIFT Features .....	25
<b>FEATURE EXTRACTION (SURF) .....</b>		<b>26</b>
<b>5.1</b>	<b>Speeded-Up Robust Features .....</b>	<b>27</b>
5.1.1	Feature Extraction .....	27
5.1.2	Interest Point Description .....	30
<b>FEATURE MATCHING .....</b>		<b>33</b>
<b>6.1</b>	<b>Feature Matching .....</b>	<b>34</b>
<b>RESULTS .....</b>		<b>37</b>
<b>7.1</b>	<b>Results .....</b>	<b>38</b>
<b>CONCLUSION AND FUTURE WORK .....</b>		<b>42</b>
<b>8.1</b>	<b>Conclusion .....</b>	<b>43</b>
<b>8.2</b>	<b>Future Work .....</b>	<b>43</b>
<b>PUBLICATIONS .....</b>		<b>44</b>
<b>REFERENCES .....</b>		<b>45</b>

## LIST OF FIGURES

Figure 1-1 Original image along with the binarized image and complemented image .....	4
Figure 1-1-2 Step by step methodology for iris recognition.....	5
Figure 3-1 Original image along with the binarized image and complemented image .....	11
Figure 3-2 Hole Filled Image (I).....	11
Figure 3-3 Pupil Center (Magnified) .....	12
Figure 3-4 Left to right: (a) Eroded Hole Filled Image, (b) Pupil Radius Detection (I-J), (c) Original Image with Pupil Center and Radius Shown .....	13
Figure 3-5 Original Image and the Filtered Image .....	14
Figure 3-6 Histogram Equalized Image .....	14
Figure 3-7 Iris Outer Boundary.....	15
Figure 4-1 Scale Space.....	19
Figure 4-2 Scale Space and DoG .....	21
Figure 4-3 Local extrema detection by comparison with 26 neighboring pixels .....	22
Figure 4-4 Beginning from left, a: maxima/minima keypoints, b: Only high contrast keypoints, c: Edge keypoints removed.....	23
Figure 5-1 Calculation of Integral Image $\Sigma=A-B-C+D$ .....	27
Figure 5-2 Second order derivatives approximated by box filters.....	28
Figure 5-3 Instead of Using Iterative Reduction of size of Images (left) we can use gradually Up- Scaled filters by using integral images (right) .....	30
Figure 5-4 Calculation of Dominant Orientation for a interest point .....	31
Figure 5-5 Haar responses in the sub-regions.....	32

## LIST OF TABLES

1. Table 7.1 Feature vector by SIFT algorithm for 1 keypoint (128 dimensions)	38
2. Table 7.2 Feature vector by SURF algorithm for 1 keypoint (64 dimensions)	39
3. Table 7.3 Number of keypoints generated for various samples of iris images of same and different persons (By SIFT)	39
4. Table 7.4 Number of keypoints generated for various samples of iris images of same and different persons (By SURF)	39
5. Table 7.5 Comparison of match scores for the SIFT algorithm	40
6. Table 7.6 Comparison of match scores for the SURF algorithm	41

# Chapter 1

## INTRODUCTION

## 1.1 Background on Biometrics

Quite often we find that ‘the need to prove our identity’ becomes a necessity and in some cases unavoidable to get access into something which is presumably secure. Personal identification, regardless of method, is ubiquitous in our daily life. May it be gaining access to your own bank account, logging into your PC, it may be while trying to draw some cash from an ATM or getting through a high-security entrance doorway, it is inescapable to prove your identity.

Conventionally we identify ourselves and gain access by physically carrying access cards, keys or by remembering passwords, secret security codes, and personal identification numbers (PINs). But unfortunately these methods have serious loopholes. Access cards and keys can be lost, duplicated or stolen whereas passwords, secret codes and PINs can be forgotten or observed by someone else. So they fail to satisfy the security requirements [1].

At this juncture Biometrics based Personal Identification System stands right at the frontier of finding an easy to go and secure option for recognizing an individual. Biometrics involves measurement of personal, physical or biological parameters and provides a more reliable answer to the question “Are you the one who you are claiming to be? Amongst a multitude of such options like measurement of weight, hair color, finger prints, voice, skin complexion etc.

A good biometrics is characterized by unique features so that the probability of two persons being wrongly matched reduces to minimum [2]. The biometrics should be such that the features used for comparison remains stable throughout life and are easily captured without imposing many constraints on the user. The most basic qualities that the

physiological or behavioral characteristics of humans must possess to qualify as preferred biometrics are listed as follows:

- i. *Universality*: Every individual must possess this characteristic;
- ii. *Distinctiveness*: Such characteristics of any two persons must be sufficiently different;
- iii. *Permanence*: The characteristics should be such that they remain stable throughout life;
- iv. *Collectability*: Quantitative measurement of characteristics must be possible.  
(For instance characteristics should not any of such kind: love, hatred, mental state, excitement level etc.)

Over and above all, one should take into account the following factors while designing a biometric system:

- i. *Performance*: Refers to the accuracy and speed of the biometric system;
- ii. *Acceptability*: Points at the willingness of people to accept the biometric system in their daily lives;
- iii. *Circumvention*: Refers to how easily one can fake one's identity while using the biometric system.

A practical biometric system must be robust to environmental and operational factors and should pose no harm to its users.

Every biometric system is subjected to error of two kinds [3]:

- i. *False match*: Mistaking measurement of biometrics from two different individuals to be the same. It is characterized by False Match Rate (FMR);

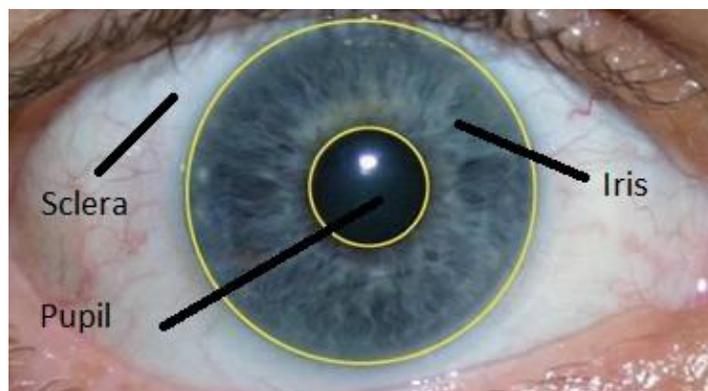
- ii. *False Non-Match*: Mistaking measurement of biometrics from a single individual to be from two different individuals. It is characterized by False Non-Match Rate (FNMR)

Every biometric system has to make a trade-off between the two, i.e. if a system is designed to be more tolerant to input variations in environmental conditions and noise factor the False Match Rate (FMR) increases, and if a system is designed to be more secure than the False Non-Match Rate (FNMR) increases. So one has to design one's biometric system keeping in mind the most specific and important requirement of problem at hand.

## 1.2 Human Iris as preferred Biometrics

Human iris is a diaphragm type structure responsible for the adjustment of pupil size and hence controls the amount of light entering the pupil by expansion and contraction of ciliary muscles owing to the variation in light to which the eye is subjected to. Iris is also well protected behind the eye-lids.

The iris shares its boundary with pupil at its inner side and sclera at its outer side as shown in Fig. 1.1.



**Figure 1-1 Original image along with the binarized image and complemented image**

Human iris is a more preferred biometric these days because of its unique spatial features which don't vary with passing time [4]. Its invasiveness also favors its use as a biometric data. An iris recognition system operates by taking an input image from the user, preprocessing the image to find the region of interest, extracting features, forming a biometric template and authenticating an individual based on the result of comparison of this template with other iris templates [5]. The step by step methodology adopted for iris recognition is as depicted in Fig. 1.2.



**Figure 1-2 Step by step methodology for iris recognition**

### **1.3 Objective**

The basic objective of this thesis is to carry out a detailed analysis and comparison of iris feature extraction using two algorithms - Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF), and then to draw a relative conclusion about the efficiency of both the algorithms depending upon the results obtained. Herein we use a Euclidean Distance based feature matching technique to match the features obtained from both of the algorithms

### **1.4 Thesis Organization**

The rest of the thesis is organized as follows. Next subsection gives a brief review on the methods of iris recognition and varied work done in this area. Chapter 2 details out the method used for preprocessing and localization of iris, followed by Chapter 3 which deals

with feature extraction using Scale Invariant Feature Transform (SIFT). The next chapter describes another algorithm called Speeded-Up Robust Feature (SURF) for the extraction of iris features. Feature matching is discussed in Chapter 5 and Chapter 6 gives the simulation results for all the methods used in the given thesis.

# Chapter 2

## LITERATURE REVIEW

## 2.1 Literature Review

Flom and Safir have proposed the idea of first iris based biometric system [6]. But they have done it for highly constrained conditions like, image should be taken with ideal headrest, the person should gaze directly onto the image acquisition system and manual operator.

The first iris recognition system was developed by Daugman at University of Cambridge. Daugman [7] has used multi-scale quadrature wavelets to extract texture phase structure information of the iris to generate a 2048 bit iriscodes and compared the difference between a pair of iris representations by computing their Hamming distance via the XOR. Wildes [5], Boles and Boashash [8] are among several others who proposed different algorithms for iris recognition. The authors in [8] have represented the unique iris features using wavelet transform zero crossings.

In [9], Nosrati et al. have used a 4 step methodology to extract circular shapes from noisy backgrounds. In the first step, they perform median filtering to smooth the image and remove salt and pepper noise. In the next step laplacian filter is used to sharpen the blurred edges and highlight the smaller details. 3<sup>rd</sup> step involves application of Canny Edge Detector for image segmentation followed by the 4<sup>th</sup> step in which Circular Hough Transform (CHT) is used to detect circular shapes.

In [5] the authors have used Hough Transform for iris segmentation, Laplacian of Gaussian (LoG) to produce feature templates and normalized correlation for matching.

# Chapter 3

## IRIS LOCALISATION

## 3.1 Image Acquisition

A camera is used to acquire the image of the iris. But care must be taken to impose less as less constraints as possible on the user of the biometrics system.

## 3.2 Image Preprocessing

The conventional steps involved are:

- i. To remove the effect of specularities lying in the pupillary area,
- ii. to localize the inner and outer iris boundaries,
- iii. to remove the eyelids portion usually modeled as noise from the localized iris area.

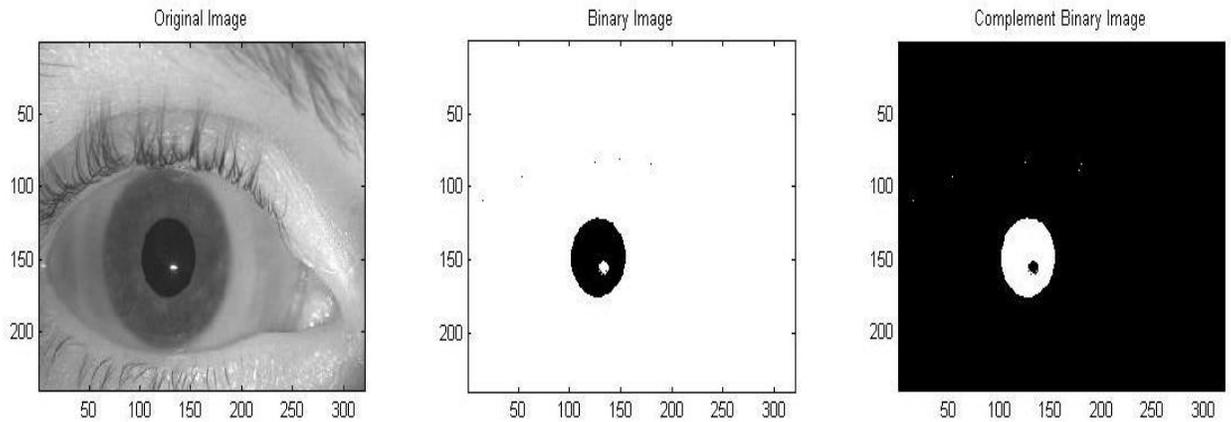
### 3.2.1 Removal of Specular Highlights

First of the entire actual image is converted into binary image by adaptive thresholding.

Steps to find the binary image:

- i. The entire image is divided into blocks of size 'w X w',
- ii. the average intensity of each block is found out,
- iii. the minimum average intensity is taken as the threshold,
- iv. the input image is compared with this threshold to obtain the binary image.

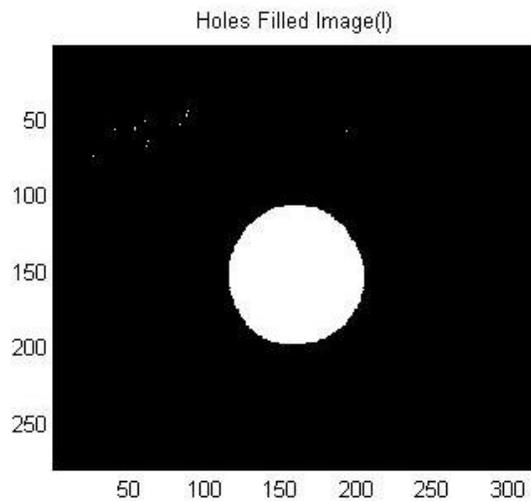
Fig.3.1. shows the original image along with the result obtained after binarization of the image and then complementing it.



**Figure 3-1 Original image along with the binarized image and complemented image**

But the image obtained after binarization contains specular highlights. So these spots are detected and filled. Here we have used *imfill()* function from MATLAB to fill the holes.

Fig.3.2 shows the hole filled image.

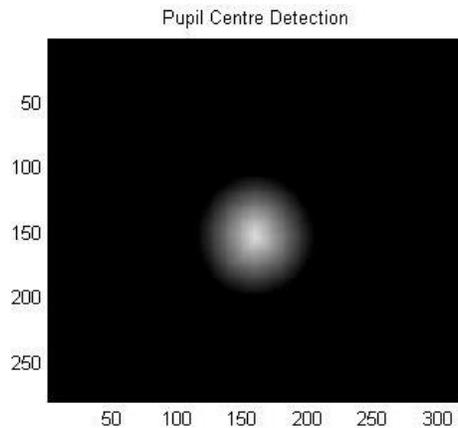


**Figure 3-2 Hole Filled Image (I)**

### 3.2.2 Pupil Detection

The hole filled image (I) is complemented and Euclidean distance of each pixel to the nearest non-zero pixel is calculated. The pixels in the white (non-zero region) would have zero Euclidean distance. They are represented by zero intensity values giving “black” region.

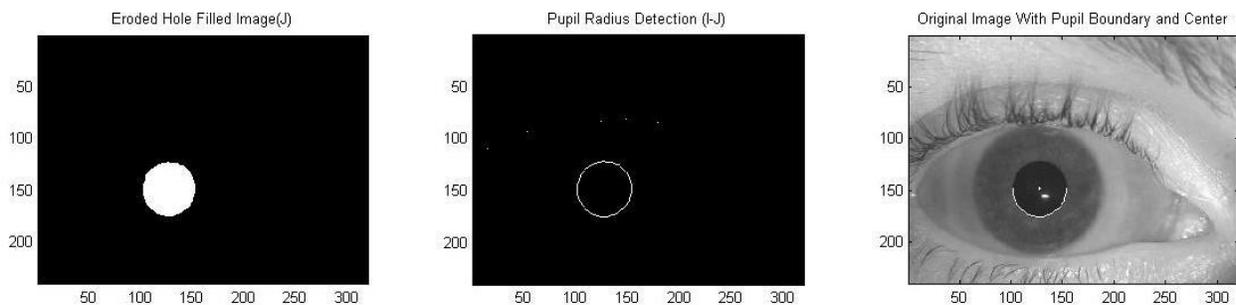
But when the pixels in the pupillary region are taken into consideration and the Euclidean distance of such pixels to the nearest non-zero pixel is calculated they would give some non-zero value. This non-zero value is too big. So while implementation through code we divide this non-zero value by some number to downscale it. These pixels are represents by this downscaled Euclidean distance. The pixel with the largest value i.e. the pixel giving the maximum Euclidean distance is found to be the center of the pupil. The center of the pupil is shown in Fig.3.3.



**Figure 3-3 Pupil Center (Magnified)**

The hole filled image (I) is eroded to give the eroded image (J). This operation is carried out in MATLAB through *imerode()* function. The boundary of the pupil is given by  $I - J$ .

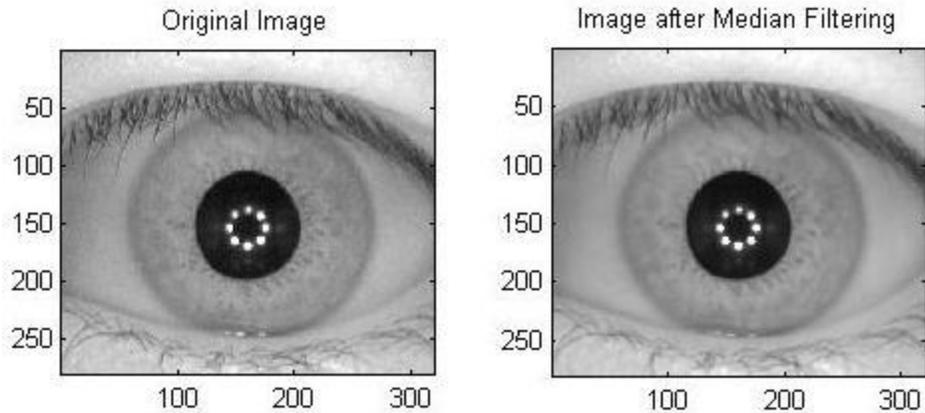
After having found the center of the pupil the distance of the center pixel to the nearest non-zero pixel gives the radius of the pupil ( $r_p$ ). This is the inner iris boundary. Fig.3.4 shows the steps in detecting the iris inner boundary.



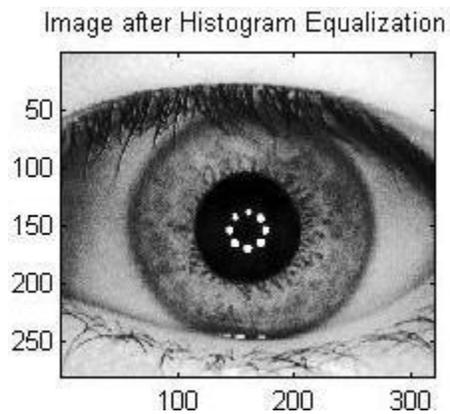
**Figure 3-4 Left to right: (a) Eroded Hole Filled Image, (b) Pupil Radius Detection (I-J), (c) Original Image with Pupil Center and Radius Shown**

### 3.2.3 Iris Detection

Median filter is used to blur the image which eliminates noise while preserving image boundaries. To have sharp image boundaries we perform histogram equalization to the blurred version of the image. Fig.3.5 shows the original image and the result obtained after median filtering the image. The result obtained after performing histogram equalization to the blurred version of the image is shown in Fig.3.6.

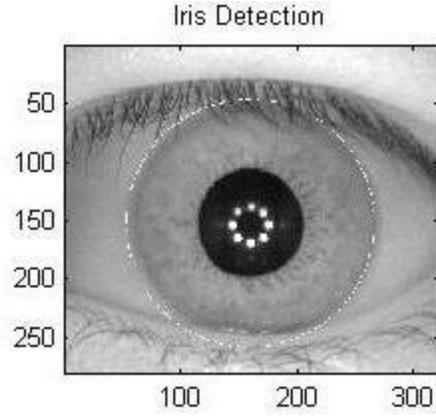


**Figure 3-5 Original Image and the Filtered Image**



**Figure 3-6 Histogram Equalized Image**

Concentric circles of varying radii (with pupil center as center) are drawn. The intensities of pixels along the perimeter of each circle are summed up. The circle having highest change in overall intensity from the previously drawn circle gives the iris outer boundary and hence the iris radius ( $r_i$ ). Fig.3.7 shows the obtained iris outer boundary.



**Figure 3-7 Iris Outer Boundary**

### 3.2.4 Removal of noise due to eye-lid occlusion

For a normal gaze it is observed that:

- i. Edge of the upper eye-lid covers half of the iris circle.
- ii. Lower eye-lid covers  $(1/4)^{\text{th}}$  of the iris circle.
- iii. Left and right iris circle regions are free from occlusion of eye-lids.

The regions that do not suffer occlusion are given by  $[35^{\circ}, 145^{\circ}]$  and  $[215^{\circ}, 325^{\circ}]$ . So by varying the values of  $r_i$  for varying  $\theta$ , the occlusion free portion of iris can be obtained.

$$r_i = \begin{cases} \frac{3}{4}r_i & \text{if } 0 < \theta < 35 \\ r_i & \text{if } 35 \leq \theta \leq 145 \\ \frac{1}{2}r_i & \text{if } 145 < \theta < 215 \\ r_i & \text{if } 215 < \theta < 325 \\ \frac{3}{4}r_i & \text{if } 325 < \theta < 360 \end{cases} \quad (3.1)$$

# Chapter 4

## **FEATURE EXTRACTION (SIFT)**

## 4.1 Scale Invariant Feature Transform

The expansion and contraction of pupil is a natural phenomenon, due to which the texture pattern of iris undergoes linear deformation. Thus, enhanced keypoint descriptor is required that can be performed in various different scale along with other transformations. Here, Scale Invariant Feature Transform (SIFT) is used as a local feature descriptor that provides features which are less sensitive to image distortions. First stage of feature extraction is to identify and construct a scale space that can be repeatedly assigned to the same object. This is done by using a cascade filtering approach that minimizes the feature extraction cost. Keypoints are detected using difference of Gaussian (DOG) images by Laplacian of Gaussian (LOG) approximation. Then keypoints are the local maxima/minima in the DOG images. After getting rid of bad keypoints which may originate due to low contrast features or edges, orientation is assigned to each of the keypoint. The orientation thus provided gives rotation invariance. Now to generate SIFT features, local image gradients are measured at selected scale in region around each keypoint to form descriptor vector. Detailed descriptions of steps outlined above are given in the following subsections.

### 4.1.1 Constructing a Scale Space

How you perceive real objects depends on the scale used to represent the object. A small sugar cube might be completely visible when viewed on tea-table, but try imagining the same in a huge platform, say the Milky Way. All difference is due to the difference in scale. This multi-scale nature of objects is quite common. And a scale space attempts to replicate this concept on digital images. While getting rid of these details, it has been

ensure that no new false details get introduced. The only way to do that is with the Gaussian Blur. So to create a scale space, the original image was taken and progressively blurred out images are generated.

The first stage of computation searches over all scales and image locations. It is being implemented efficiently by using difference-of-Gaussian (DoG) to identify potential interest points that are invariant to scale and orientation.

The number of octaves and scale depends on the size of the original image. However, 4 octaves and 5 blur levels are ideal for the algorithm. As more keypoints give better result, so the original image is doubled in size and anti-aliased a bit (by blurring it) and the algorithm produces more four times more key-points.

The scale space of image is defined as,

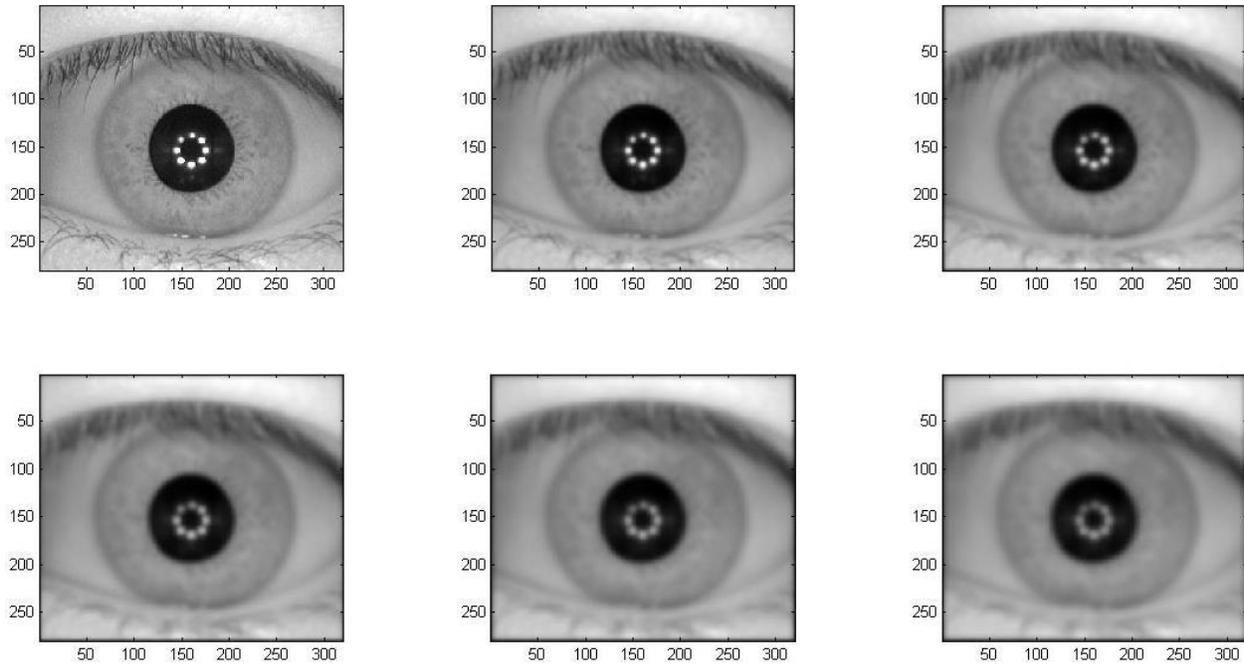
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4.1)$$

Where,  $I(x, y)$  is the input image and  $*$  is the convolution operation in  $x$  and  $y$ .

$G(x, y, \sigma)$  is the variable scale Gaussian blur operator defined as :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (4.2)$$

So in this step of SIFT, several octaves of the original images are generated and each octave's image size is half of the previous one where within a single octave, images are progressively blurred using the Gaussian Blur operator. In the next step, all these octaves are used to generate the Difference of Gaussian (DOG) images. Fig.4.1 shows the successive blurred of the image and hence the obtained scale space.



**Figure 4-1 Scale Space**

#### **4.1.2 Laplacian of Gaussian (LoG) Approximation**

In the Laplacian of Gaussian (LoG) operation, an image is taken and is blurred to get the blurred image. And then, the second order derivatives applied on it (or, the “laplacian”) are calculated. This locates edges and corners on the image. These edges and corners are best for locating keypoints. But the problem arises due to extreme sensitivity of second order derivative to noise. The blur helps in smoothing it out and stabilizes the second order derivative. So, to generate the Laplacian of Gaussian (LoG) images quickly, the scale space is used and the difference between two consecutive scales or the Difference of Gaussians are calculated. The DoG approximates the LoG images and hence converts a computationally complex procedure into mere subtraction which proves to be fast and efficient. We also found out that these approximations are “scale invariant”.

To detect stable keypoint locations in the scale space, Difference of Gaussian (DOG) function is convolved with the image. The Difference of Gaussian (DOG) for two nearby scales of an iris image  $I$  is computed as,

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (4.4)$$

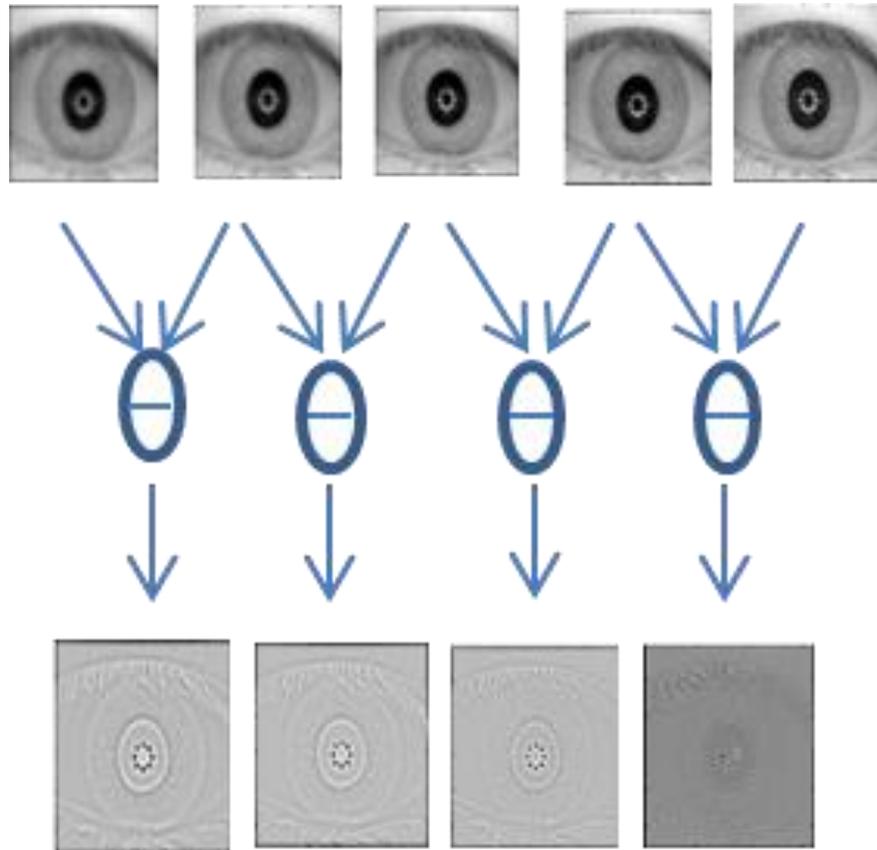
Where ' $k$ ' is a constant multiplicative factor which is used for changing the scale and  $x, y$  are the coordinates of a pixel in image  $I$ .

The progressive blurring of images and generation of scale space by performing subtraction is as shown in Fig. 4.2.

So in this 2<sup>nd</sup> step of SIFT, two consecutive images in an octave are picked and one is image is subtracted from the other. Then the next consecutive pair is taken and the process repeats. Same steps are followed for all the octaves. The resulting images are an approximation of scale invariant laplacian of gaussian (which is good for detecting keypoints). There are a few “drawbacks” due to the approximation taken, but they won't affect the algorithm.

### 4.1.3 Finding Keypoints

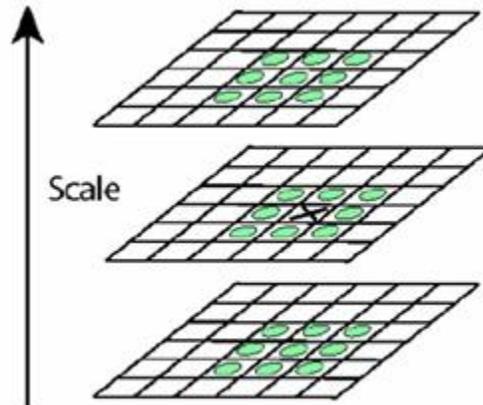
The first step is to coarsely locate the maxima and minima. DOG images are used to detect interest points with the help of local maxima and minima across different scales.



**Figure 4-2 Scale Space and DoG**

The first step is to coarsely locate the maxima and minima. DOG images are used to detect interest points with the help of local maxima and minima across different scales. DOG images were iterated through each pixel and all its neighbors were checked. The check was done within the current image, and also with the one above and below it. Each pixel in DOG image was compared to 8 neighbors in the same scale and 9 neighbors in the neighboring scales. The pixel was selected as a candidate keypoint if it was found to be local maxima or minima in the neighbor of the pixel. The comparison of a pixel with its 26 neighbors is shown in Fig.4. 'X' marks the current pixel. Neighbours are marked by

the green circles. This way, a total of 26 checks are made. Fig.4.3 shows the finding of local maxima and minima by comparison with neighbouring pixels.



**Figure 4-3 Local extrema detection by comparison with 26 neighboring pixels**

The algorithm of SIFT is best suited for generating 2 such extrema images. So, exactly 4 DoG images are required and to generate 4 DoG images, 5 Gaussian blurred images are required. Hence the 5 level of blurs in each octave was done in the first step of the algorithm.

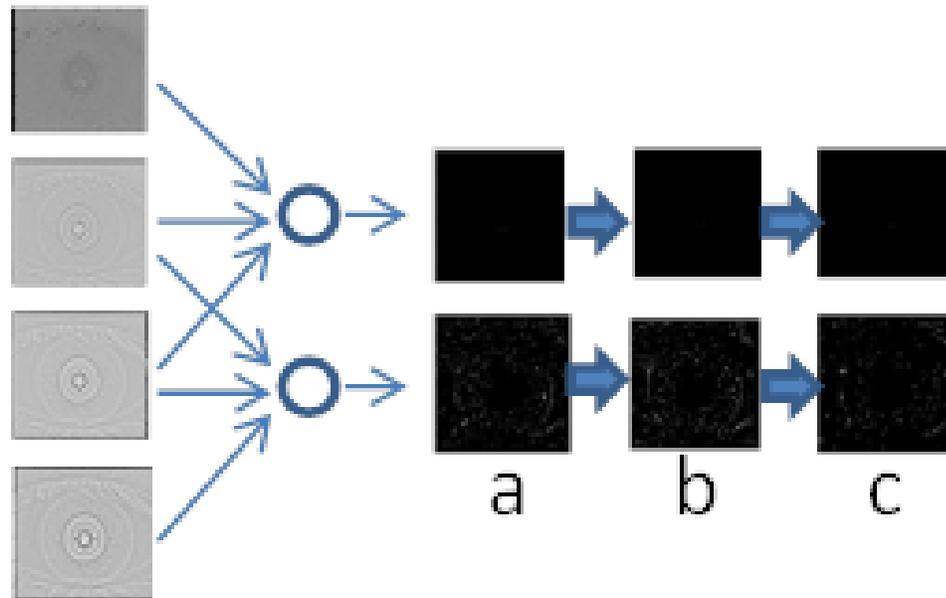
So in this 3<sup>rd</sup> step of SIFT, the maxima and minima in the DoG images generated in the previous step were detected which was done by comparing neighbouring pixels in the current scale, the scale above and the scale below.

#### **4.1.4 Getting Rid of Bad Keypoints**

A lot of keypoints are generated in the previous step . Some of them are found to lie along an edge, or they lack in contrast. In both of the cases, they are not useful as

features. So to get rid of them, the approach similar to the one used in the Harris Corner Detector [10] is used for removing edge features. For low contrast features, If the magnitude of the intensity (i.e., without sign) at the current pixel in the DOG image (that is being checked for minima/maxima) is less than a certain value, it is rejected. For removing features along the edges, the Hessian Matrix [11] is used to check if a point is a corner or not. The two gradients at the keypoint both perpendicular to each other is calculated and based on the image around the keypoint, if both the gradients will be big enough then it is pass as a keypoint, otherwise it is rejected.

So in this 4<sup>th</sup> step of SIFT, the number of keypoints were reduced. This helps increase efficiency and also the robustness of the algorithm. Keypoints were rejected if they had a low contrast or if they were located on an edge. In the next step, an orientation to all the keypoints that passed both tests will be assigned. Fig.4.4 shows the steps that lead to obtaining optimum keypoints.



**Figure 4-4** Beginning from left, a: maxima/minima keypoints, b: Only high contrast keypoints, c: Edge keypoints removed

### 4.1.5 Orientation Assignment

After step4, legitimate key points have been tested to be stable. The next step is to assign an orientation to each keypoint which will provide rotation invariance. Gradient directions and magnitudes are collected around each keypoint and most prominent orientation(s) is assigned to the keypoint in that region. The size of the orientation collection region around the keypoint depends on its scale. Figure 3 shows the steps of removal of bad keypoints.

To determine the keypoint orientation, a gradient orientation histogram is formed in the neighborhood of keypoint. The scale of keypoint is used to select Gaussian smoothed image  $L$ . For each Gaussian smoothed image,  $L(x,y)$ , magnitude ( $m(x, y)$ ) and orientation ( $\theta(x, y)$ ) are computed as:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (4.5)$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y))) \quad (4.6)$$

Then gradient orientation histogram is formed around each keypoint. The histogram has 36 bins for 360 orientations and each sample is weighted by gradient magnitude and Gaussian weighted circular window with  $\sigma = 1.5$  times of scale of keypoint before adding it to histogram. Peaks in the histogram correspond to the orientation and any other local peak within 80% of largest peak is used to create a new keypoint with the computed orientation. This new keypoint has the same location and scale as the original but its

orientation is equal to the other peak and so an orientation can split up one keypoint into multiple keypoints. This will increase stability during the matching process.

#### **4.1.6 Generating SIFT Features**

Once orientation has been selected, unique fingerprint for the keypoint has to be generated. The feature descriptor is computed as a set of orientation histograms on  $4 \times 4$  pixel neighborhoods. The orientation histograms are relative to the keypoint orientation is shown in the Figure. First of all a  $16 \times 16$  window around the keypoint is taken which is broken into sixteen  $4 \times 4$  windows. Within each  $4 \times 4$  window, gradient magnitudes and orientations are calculated and are put into an 8 bin histogram. Any gradient orientation in the range 0-44 degrees add to the first bin, 45-89 add to the next bin and so on the amount added to the histogram bin depends on the magnitude of the gradient. Unlike the past, the amount added also depends on the distance from the keypoint and so gradients that are far away from the keypoint will add smaller values to the histogram. Doing this for all 16 pixels, 16 totally random orientations are compiled into 8 predetermined bins and this is done for all the sixteen  $4 \times 4$  regions. So this generates SIFT feature descriptor of  $4 \times 4 \times 8 = 128$  elements. These 128 numbers form the feature vector and the keypoint is uniquely identified by this feature vector. The feature vector is invariant to rotation, scale variations and is immune to illumination changes.

# Chapter 5

## **FEATURE EXTRACTION (SURF)**

## 5.1 Speeded-Up Robust Features

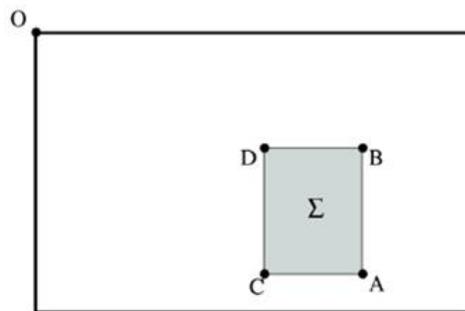
SURF is a robust and scale invariant feature vector detection algorithm which was first proposed by Herbert Bay et al. in the year of 2006 [12]. SURF is found to be computationally simpler than SIFT.

### 5.1.1 Feature Extraction

Familiarization with the concept of integral images is necessary to begin with SURF. The integral images help in computation of box type convolution filters. Integral image  $I_{\Sigma}(x)$  at a location  $(x,y)^T$ , is given by sum of all the pixels in the input image  $I$  within a rectangular region formed between origin and  $x$ .

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (5.1)$$

Upon calculation of the integral image it takes three additions to calculate intensity of any rectangular area as shown in Figure 5.1. So the calculation time is independent of filter size which serves as an advantage in case of filters of larger size.



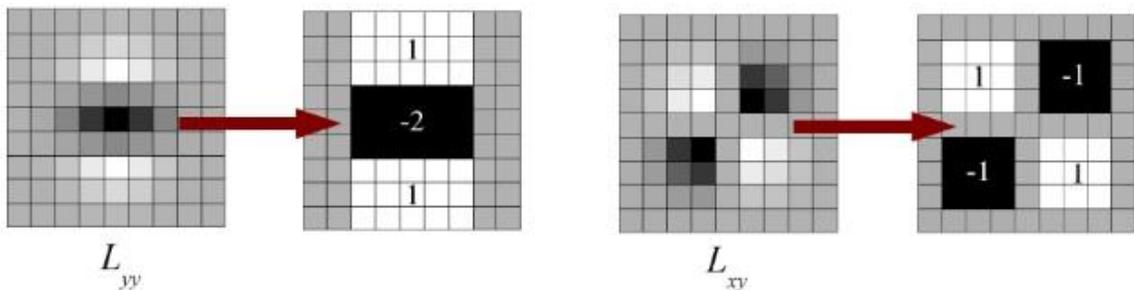
**Figure 5-1 Calculation of Integral Image  $\Sigma=A-B-C+D$**

We use Hessian matrix to detect the interest points. The locations where the Hessian determinant is maximum gives rise to an interest point. The Hessian matrix as given as follows:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (5.2)$$

Where,  $H(x, \sigma)$  is the Hessian matrix at  $x$  at scale  $\sigma$ .  $L(x, \sigma)$  is the Laplacian of Gaussian (LoG), i.e. the convolution of Gaussian second order derivative with the image in point  $x=(x,y)$ .

In practice the Gaussian second order derivatives are approximated with box filters as shown in Figure 15. There is a slight decrease in performance due to this approximation in terms of loss of repeatability under image rotations. But the advantage of fast convolutions outweighs the disadvantage. These approximated second order derivatives can be very easily computed by using integral images. The use of integral images makes the calculation time independent of the filter size. The 9X9 box filters shown in Fig. 5.2 are approximations of Gaussians with  $\sigma=1.2$  and this represents the lowest scale space.



**Figure 5-2 Second order derivatives approximated by box filters**

The approximated LoG are represented as  $D_{xx}, D_{xy}$  and  $D_{yy}$ . So the determinant of the Hessian matrix is given as:

$$\det(H(x, \sigma)) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (5.3)$$

$w$  balances for the approximation in the Gaussian second order derivative.

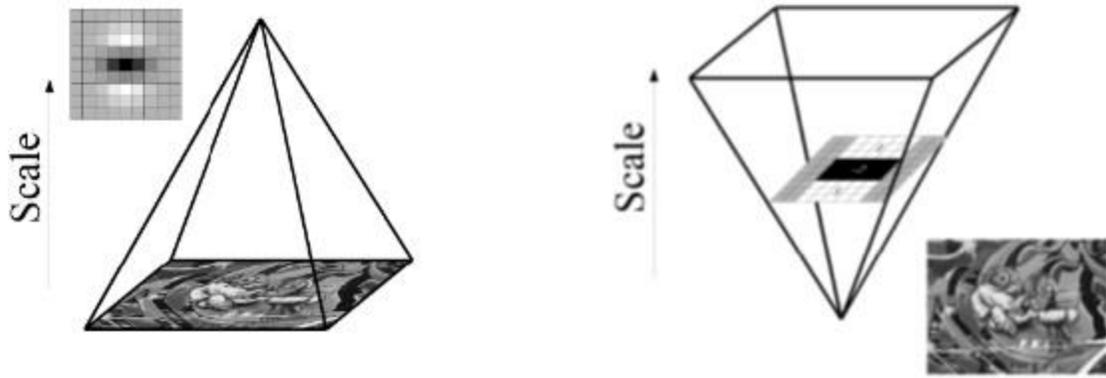
$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912\dots \approx 0.9$$

The weight  $w$  should change for theoretical correctness but in practice we keep  $w$  fixed and equal to 0.9.

We find out interest point at different scales so that matching of image can be done at any scale. Scale spaces are implemented as image pyramids. The images are smoothed again and again with Gaussian second derivative and then are sub-sampled to obtain a higher level of the scale pyramid. Two consecutive layers are then subtracted which gives the interest points [9].

At this point we have two options, either we can reduce the image size at each level or we can keep the image size fixed and apply filter of appropriate size i.e. we can upscale the filter size. We follow the later approach because the use of integral images while computing box type of filters brings in independency in size and thus can be computed at exactly same speed. The two approaches at setting up of scale space are shown in Fig.5.3.

The 9x9 filter shown in Figure 3 is taken as the initial scale which is referred to as  $s=1.2$  scales and it approximates Gaussian with  $\sigma=1.2$ . The further levels are obtained by applying increasingly larger filters to the image. The scale space thus is divided into octaves where in each octave we apply progressively up-scaled filters to the same image.



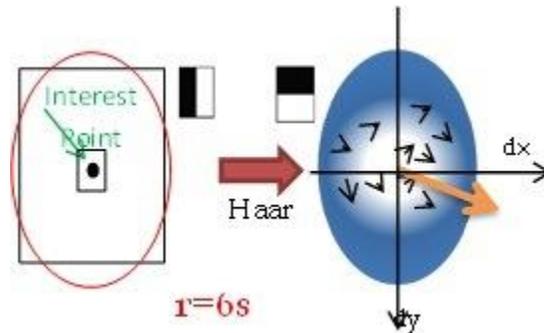
**Figure 5-3 Instead of Using Iterative Reduction of size of Images (left) we can use gradually Up-Scaled filters by using integral images (right)**

The first octave thus contains filters of sizes  $9 \times 9$ ,  $15 \times 15$ ,  $21 \times 21$  and  $27 \times 27$  i.e. with a gradual increase of 6. The next octave consists of filters of increasing sizes with a difference of 12. Similarly further octaves can be formed but it is found that the number of feature points decreases with each higher level octave. In this paper we have stopped at the first octave since the results were pretty good even with the first octave.

### 5.1.2 Interest Point Description

First of all a circular window of size  $6s$  ('s' refers to the scale at which the interest point was detected, i.e. '1' in our case) was drawn with the interest point at the center as shown in Figure 5.4.

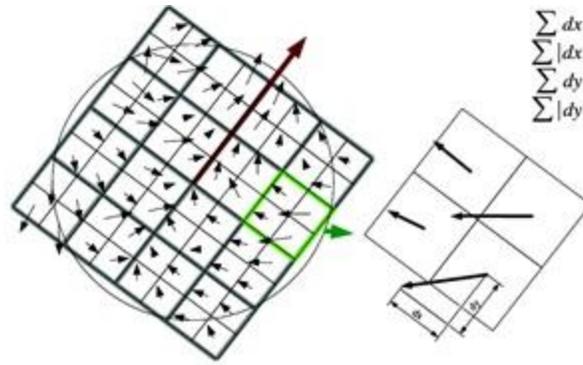
The wavelet responses in  $x$  and  $y$  direction of the points (sampled at 's') in the circle about the interest point are calculated. The wavelet responses are calculated using the concept of integral images with a window size  $4s$  and then we weighted the responses with Gaussian kernel. The weighted wavelet responses are then represented as points in space with  $dx$  along abscissa and  $dy$  along ordinate.



**Figure 5-4 Calculation of Dominant Orientation for a interest point**

We then took an angular window of angle= $60^\circ$ . And then swept this window about the entire circle. While doing so we found out the sum of the  $dx$  and  $dy$  of all the points in that window. This gave us the local orientation vector. The longest of such vector gives the orientation of the interest point.

After having found out the orientation of all the interest points, we drew a square region around the interest point and oriented along the direction found prior to this. The interest region (square region around each interest point) of size  $20s$  is split into  $4 \times 4$  sub regions. For each sub-region we calculate the haar wavelet responses at points sampled at  $5 \times 5$ . Here we use integral images with filter size  $2s$ . The  $dx$  response is along the abscissa and  $dy$  along the ordinate. Combined together they give rise to a vector as shown in Fig.5.5.



**Figure 5-5 Haar responses in the sub-regions**

The  $dx$  and  $dy$  along each sub region are summed up to give  $\sum dx$  and  $\sum dy$ . We also calculated  $\sum |dx|$  and  $\sum |dy|$  to have information about the polarity changes in intensity. So each sub-region resulted in a descriptor vector of 4 dimensions viz.  $v=(\sum dx, \sum |dx|, \sum dy, \sum |dy|)$ . Doing this for all sub-regions we got a descriptor vector with 64 dimensions. This 64 dimensional feature vector corresponds to each keypoint. Similar 64 dimensional vectors is obtained for each interest point/keypoint.

# Chapter 6

## **FEATURE MATCHING**

## 6.1 Feature Matching

Our image matching is based on finding the Euclidean distance between keypoints of corresponding images to be matched and find out matched pair of keypoints by the method of comparing with a given threshold.

For each detected keypoint we have got a 128 dimensional feature vector in case of SIFT and a 64 dimensional feature vector in case of SURF. The feature matching process described here is applicable to features extracted by SIFT and SURF invariably.

Let we have  $n$ -dimensional feature vectors  $(x_1, x_2, \dots, x_n)$  for each keypoint in both the images.

Case 1: The images to be compared have same number of keypoints

Let the feature vectors of both the images be defined as such:

Image 1

$$V1: (x_{11}, x_{12}, \dots, x_{1n})$$

$$V2: (x_{21}, x_{22}, \dots, x_{2n})$$

...

$$Vn: (x_{n1}, x_{n2}, \dots, x_{nn})$$

Image 2

$$W1: (y_{11}, y_{12}, \dots, y_{1n})$$

$$W2: (y_{21}, y_{22}, \dots, y_{2n})$$

...

$$WN: (y_{n1}, y_{n2}, \dots, y_{nn})$$

We considered a keypoint in the first image and then found out the Euclidean distance between this keypoint and another keypoint in the second image.

The formula for finding the Euclidean distance is as given below:

$$d1 = \sqrt{(x_{11} - y_{11})^2 + (x_{12} - y_{12})^2 + \dots + (x_{1n} - y_{1n})^2} \quad (6.1)$$

Similarly we found the Euclidean distance between the first keypoint of the first image and other keypoints of the second image.

$$d2 = \sqrt{(x_{11} - y_{21})^2 + (x_{12} - y_{22})^2 + \dots + (x_{1n} - y_{2n})^2} \quad (6.2)$$

$$d3 = \sqrt{(x_{11} - y_{31})^2 + (x_{12} - y_{32})^2 + \dots + (x_{1n} - y_{3n})^2} \quad (6.3)$$

...

$$dn = \sqrt{(x_{11} - y_{n1})^2 + (x_{12} - y_{n2})^2 + \dots + (x_{1n} - y_{nn})^2} \quad (6.4)$$

After having found out the Euclidean distance between keypoint '1' of first image and all the keypoints of second image, we stored the obtained distances in an array.

To find out matching for the first keypoint in the second image we adopt a 'hit and trial' method wherein we set a threshold value say 't'. Then by comparing each of the obtained distances ( $d1, d2, \dots, dn$ ) with this threshold we found out the distance which is less than or equal to the threshold i.e.  $Wx$  is the matched feature vector in the second image if  $dx \leq t$ , else we discard the interest point.

This gave us the matched keypoint in the second image. In any case where we obtained multiple keypoints which satisfied the above criteria, we considered that keypoint which gave the minimum Euclidean distance. Every time we found out a matched pair, we

removed the interest points from our next iteration and followed the above procedure for a new keypoint in the first image.

To check if we have found out a suitable match for the first image we again set a threshold say ' $k$ '. Then we calculated a ratio given as such:

$$r = \frac{\text{number of matched pair of keypoints}}{\text{total number of keypoints in each image}}$$

We compared this ratio with the threshold ' $k$ '. If ' $r$ ' exceeded ' $k$ ' then we concluded that we have found out a match, else not.

Case 2: The images to be compared have different number of keypoints

We followed entirely similar procedure except for the calculation of ratio ' $r$ '. In this very case the ratio was calculated as per the formula given below:

$$r = \frac{\text{number of matched pair of keypoints}}{\min(m, n)}$$

Where,  $m = \text{number of keypoints in the first image}$

$n = \text{number of keypoints in the second image}$

# Chapter 7

## RESULTS

## 7.1 Results

The feature matching for Iris Recognition System is divided into two phase (1) Training Phase (2) Testing Phase. Here CASIA database has been used to test the algorithm and we have used a sample of five people with two sets of iris images each, to generate the match scores (r). A MATLAB implementation of the whole algorithm is done and output is recorded in the form of tables. The sample images are taken in different lighting condition and with absence of spectacles.

Table 7.1 and table 7.2, respectively lists down the 128 dimensional feature vector generated in SIFT algorithm and the 64 dimensional vector generated in SURF algorithm for a single keypoint in a structured manner.

**TABLE 7.1**

**Feature vector by SIFT algorithm for 1 keypoint (128 dimensions)**

0.075946	0.119733	0.021478	0.248749	0.045974	0.248749	0.111472	0.038742
0.188389	0.106774	0.029003	0.112657	0.043014	0.133405	0.005393	0.074532
0.033739	0.01196	0.010863	0.00158	0.013169	0.02533	0.000377	0.041668
0.002147	0.000156	0.003212	2.29E-05	0.007451	0.00142	0	0.010559
0.000243	0.009783	0.004161	0.004693	0.005482	0.001397	0	0.002056
0.000207	0.151548	0.007706	0.028033	0.004847	0.001636	0.000243	0.042234
0.005212	0.064666	0.092566	0.027104	0.025132	0.002591	0.018447	0.081004
0.01822	0.042878	0.170866	0.098036	0.102012	0.180325	0.17111	0.028093
0.084325	0.00787	0.187175	0.074851	0.073374	0.063828	0.048994	0.012772
0.248749	0.000255	0.093363	0.017042	0.009	0.214758	0.00889	0.223019
0.140306	0	0.03275	0.001235	0.003317	0.048503	4.19E-05	0.084575
0.000166	0.0002	0.002434	0.000178	0.007622	0.001797	0	0.009371
0.000888	0.098779	0.014713	0.179889	0.016981	0.040578	3.22E-05	0.003667
0.002383	0.248749	0.017734	0.248749	0.012929	0.119291	0.043454	0.026527
0.006499	0.095196	0.083241	0.048844	0.06921	0.045779	0.156963	0.014192
0.008493	0.020089	0.063459	0.03698	0.248749	0.005422	0.169125	0.002142

**TABLE 7.2****Feature vector by SURF algorithm for 1 keypoint (64 dimensions)**

0	0.283488	0	0.393852	0	0.382678	0	0.243815
0	0.054808	0	0.019717	0	-0.00923	0	-0.04113
0	0.283649	0	0.395733	0	0.394562	0	0.246529
0	0.069094	0	0.168204	0	0.070803	0	0.073617
0.005087	0.022695	0.07088	0.020442	0.095681	0.017797	0.031032	0.031595
0.00229	0.001196	0.007316	-0.0033	0.000244	-0.00191	-0.00489	-0.00435
0.005087	0.031279	0.07088	0.044291	0.095681	0.058777	0.031032	0.041245
0.003305	0.027641	0.01439	0.102718	0.008622	0.055017	0.006661	0.031074

In the rest 4 tables following the text, PxSy symbolizes, x<sup>th</sup> person's y<sup>th</sup> sample. Table 7.3 lists down the number of keypoints generated by applying SIFT algorithm to the iris samples of different individuals. Similar results for SURF algorithm are shown in table 7.4.

**TABLE 7.3****Number of keypoints generated for various samples of iris images of same and different persons (By SIFT)**

P1S1	P1S2	P2S1	P2S2	P3S1	P3S2	P4S1	P4S2	P5S1	P5S2
330	421	343	501	406	499	285	277	592	534

**TABLE 7.4****Number of keypoints generated for various samples of iris images of same and different persons (By SURF)**

P1S1	P1S2	P2S1	P2S2	P3S1	P3S2	P4S1	P4S2	P5S1	P5S2
99	108	92	90	87	105	66	78	148	124

The results listed in table 7.5 and table 7.6 contains the match scores of features generated by SIFT and SURF respectively. For SIFT a global threshold of 0.14 was taken which reduces the false acceptance and false rejection rate. Similarly for SURF, a global threshold of 0.26 was chosen and was found to give better results than SIFT. From the tables, it was inferred that 2 persons were falsely accepted for a match with a match score greater than 0.14 and 1 person was falsely rejected with a match score less than 0.14. But in SURF it was found that only 1 person was falsely accepted with match score exceeding the threshold.

**TABLE 7.5**

**Comparison of match scores for the SIFT algorithm**

	P1S1	P1S2	P2S1	P2S2	P3S1	P3S2	P4S1	P4S2	P5S1	P5S2
P1S1	1	0.16	0.13	0.10	0.11	0.13	0.12	0.16	0.10	0.1
P1S2	0.11	1	0.15	0.12	0.09	0.09	0.10	0.10	0.09	0.08
P2S1	0.12	0.09	1	0.18	0.10	0.09	0.12	0.12	0.09	0.10
P2S2	0.08	0.10	0.14	1	0.09	0.08	0.09	0.09	0.08	0.07
P3S1	0.11	0.08	0.11	0.11	1	0.17	0.10	0.14	0.1	0.1
P3S2	0.12	0.09	0.11	0.10	0.19	1	0.12	0.11	0.05	0.09
P4S1	0.12	0.11	0.13	0.12	0.12	0.10	1	0.21	0.11	0.11
P4S2	0.13	0.10	0.11	0.09	0.12	0.08	0.16	1	0.06	0.17
P5S1	0.08	0.07	0.09	0.06	0.10	0.07	0.10	0.08	1	0.22
P5S2	0.09	0.09	0.10	0.07	0.09	0.10	0.08	0.11	0.25	1

**TABLE 7.6****Comparison of match scores for the SURF algorithm**

	P1S1	P1S2	P2S1	P2S2	P3S1	P3S2	P4S1	P4S2	P5S1	P5S2
P1S1	1	0.28	0.19	0.17	0.18	0.19	0.19	0.16	0.23	0.22
P1S2	0.30	1	0.25	0.17	0.22	0.23	0.17	0.18	0.24	0.23
P2S1	0.23	0.18	1	0.31	0.25	0.22	0.18	0.21	0.19	0.23
P2S2	0.25	0.21	0.27	1	0.16	0.15	0.13	0.20	0.20	0.25
P3S1	0.13	0.11	0.15	0.13	1	0.29	0.15	0.14	0.22	0.15
P3S2	0.22	0.21	0.22	0.21	0.27	1	0.23	0.20	0.21	0.16
P4S1	0.17	0.23	0.25	0.19	0.31	0.32	1	0.40	0.22	0.18
P4S2	0.24	0.21	0.24	0.23	0.21	0.25	0.27	1	0.20	0.25
P5S1	0.19	0.21	0.11	0.18	0.23	0.24	0.19	0.18	1	0.28
P5S2	0.17	0.17	0.19	0.24	0.18	0.16	0.10	0.18	0.27	1

# Chapter 8

## CONCLUSION AND FUTURE WORK

## **8.1 Conclusion**

The implementation of SURF and SIFT showed that the SURF was time effective owing to its use of integral images in approximation to Gaussian second order derivatives. Also the descriptor vector generated in SURF is 64 dimensional in comparison to SIFT having a descriptor vector of length 128. So the feature matching is also faster and computationally less complex. SURF is found to be faster than SIFT by nearly 3 times, and has recall accuracy not worse than SIFT. Also SURF is good at handling images which has been degraded while taking the images.

## **8.2 Future Work**

Much work has not been done in testing with color images with 3D viewpoint and varying illumination. So a systematic study in this direction can be helpful in generating better descriptors which can override the constraints of varying illumination, color variance and rotation.

## **PUBLICATIONS**

A Journal Paper on “**Comparision of Iris Identification by using modified SIFT and SURF keypoint keypoint descriptor**” is communicated to **International Journal of Science, Education and Research (IJSER)** journal.

## REFERENCES

- [1] R. Bolle and S. Pankanti. *“Biometrics Personal Identification in Networked Society,”* Kluwer Academic Publishers, Norwell, MA, USA, 1998.
- [2] J. Daugman. “The importance of being random: statistical principles of iris recognition,” *Pattern Recognition*, 36(2):279 – 291, 2003.
- [3] A. K. Jain, P. Flynn, and A. A. Ross. *“Handbook of Biometrics,”* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
- [4] J. G. Daugman. “High confidence visual recognition of persons by a test of statistical independence,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.
- [5] R. P. Wildes. “Iris recognition: An emerging biometric technology,” *Proc. IEEE*, vol. 85, no. 9, pp. 1348–1363, Sep. 1997.
- [6] W. W. Boles and B. Boashash. “A human identification technique using images of the iris and wavelet transform,” *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1185–1188, Apr. 1998.
- [7] P. Gupta, H. Mehrotra, A. Rattani, A. Chatterjee and A.K. Kaushik. “Iris recognition using corner detection,” *23rd International Biometric Conference, Montreal, Canada, 2006.*
- [8] D. G. Lowe. “Distinctive image features from scale invariant keypoints,” *International Journal on Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008.
- [10] C. Harris and M. Stephens. “A combined corner and edge detection,” *proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [11] R.C. Gonzalez and R.E. Woods. *“Digital Image Processing (3rd Edition).”* Prentice Hall, 2007