Feature Detection using S-Transform

Manish Bansal

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Feature Detection using S-Transform

Thesis submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Manish Bansal

(Roll: 109CS0132)

under the guidance of

Prof. Banshidhar Majhi

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.
May 2013

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

May 10, 2013

Certificate

This is to certify that the work in the thesis entitled ***Feature Detection using S-Transform*** by ***Manish Bansal*** is a record of an original research work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Banshidhar Majhi
Professor

Acknowledgment

"If God brings you to it, He will bring you through it."
Thank you God for showing me the path...

I owe deep gratitude to the ones who have contributed greatly in completion of this thesis.

Foremost, I would also like to express my gratitude towards my project advisor, Prof. Bansidhar Majhi for believing in me and providing me with the opportunity to work in the challenging area of Image Processing. His dedication has motivated me to work for excellence.

I am grateful to Prof. Pankaj Sa for providing useful insights into my project and always being with me whenever I needed him.

I am also very thankful to Ms. Hunny Mehrotra for reviewing my thesis.

I would like to thank all my friends and lab-mates for their encouragement and understanding. Their help can never be penned with words.

Most importantly, none of this would have been possible without the love and support of my parents. My parents to whom this dissertation is dedicated to, has been a constant source of love, support, strength, motivation and inspiration. I would like to express my heart-felt gratitude to them.

Manish Bansal

Dedicated to my Parents

Abstract

Images are characterized by features. Machines identify and recognize a scene or an image by its features. Edges, objects, and textures are some of the features that distinguish one image from another. There could be many common features in similar images. But, in those commonalities there lies a distinction in terms of features known as subtle features. Numerous algorithms have been reported to extract features from images. Few of them are reliable. Some of them do well under a constrained environment. Many of them fail miserably under low intensity, noise etc. The prominent features are very well identified by many algorithms, whereas the subtle features are often overlooked. In this thesis an attempt has been made to develop an algorithm to extract very subtle features from a given image. A new method has been proposed on the principle of phase congruency to detect features in images. The proposed method uses S-Transform to calculate phase congruency. The proposed method is able to calculate the subtle features even in the very low intensity images. Finally, an application of the proposed method in fingerprint minutiae extraction has also been demonstrated.

Contents

Certificate ii

Acknowledgement iii

Abstract v

List of Figures viii

1 Introduction 1

2 S-Transform 5

2.1 Signals and their types . . . 5

2.2 Derivation of S-Transform . . . 6

2.2.1 Derivation from Short Time Fourier Transform . . . 8

2.2.2 Derivation from Wavelet Transform . . . 9

2.3 Properties of S-Transform . . . 9

2.4 Discrete S-Transform . . . 11

2.5 Advantages of S-Transform . . . 12

2.6 Summary . . . 13

3 Novel Feature Detection based on Phase Congruency 14

3.1 Phase Congruency . . . 15

3.2 Existing Work . . . 15

3.3 Proposed Approach . . . 16

3.3.1 Working of Proposed Algorithm *ComputePCImage* . . . 18

3.3.2 Working of Proposed Algorithm *ComputePC* . . . 22

4 Simulation and Results 24

4.1 Analysis of Proposed Edge Detector . . . 24

4.2 Application in Biometrics 28

5 Conclusion and Future Work **32**

Bibliography **33**

List of Figures

1.1 Image showing range of subjective brightness of human eye [1]. . . . 2

2.1 (a) Stationary signal, (b) Non-Stationary Signal, (c) Fourier Transform of Stationary Signal in (a), and (d) Fourier Transform of Stationary Signal in (b). 7

3.1 Few Fourier components of a square wave are shown. 14

3.2 Figure showing the working of the proposed model. 17

3.3 (a) An input image (b) Edges detected in the input image (a) using algorithm 2, i.e., ComputePCImage. 18

3.4 (a) Image of a Fingerprint, (b) Orientation Information of fingerprint image in (a) using Algorithm 4, i.e., ModifiedPCImage. 19

3.5 (a) Input fingerprint image to Algorithm 2, i.e., *ComputePCImage*, (b) Features detected using window size 3×3, (c) Features detected using window size 5×5, and (d) Features detected using window size 7×7. In (b),(c)&(d), we have taken negative of all the output images. 21

4.1 (a) Input image, (b) Edges detected by the Sobel Operator, (c)Edges detected by Canny Edge Detection Algorithm, (d) Raw output image from Kovesi's Phase Congruency, (e) Raw Output image by the proposed algorithm, and (f) Output Image after thresholding applied to image (e). 26

4.2 (a) Input image, (b) Edges detected by the Sobel Operator, (c)Edges detected by Canny Edge Detection Algorithm, (d) Raw output image from Kovesi's Phase Congruency, (e) Raw Output image by the proposed algorithm, and (f) Output Image after thresholding applied to the image (e). 27
4.3 (a) Input image, (b) Edges detected by the Sobel Operator, (c)Edges detected by Canny Edge Detection Algorithm, (d) Raw output image from Kovesi's Phase Congruency, (e) Raw Output image by the proposed algorithm, and (f) Output Image after thresholding applied to the image (e). 29
4.4 Various forms of authentication. Traditional methods of authentication using token based and knowledge based approaches (left). Use of biometrics to claim identity (right) 30
4.5 (a) Minutiae detected by Algorithm [2], (b) Minutiae detected by the algorithm 5 on same input image as (a). 31

Chapter 1

Introduction

Images are characterized by features. To identify features such as edges and their significance, one should strive for a dimensionless quantity like intensity invariance and orientation invariance. Since such dimensionless quantity would provide absolute significance of feature points irrespective of intensity and orientation, one could universally apply them to any image. With such quantities it would be possible to compare or match images independent of their local properties.

A lot of effort has already been put in this direction to detect invariant measures of high level structures in images. Hu [3] developed a series of invariant moments to recognize binary objects. Then, a lot of work was done on geometric invariance, i.e, identification of geometric properties of objects that remain invariant to imaging transformations. All the work in the area of geometric invariance has been summarized in the book by Mundy and Zisserman [4]. However, very little work has been carried out in the direction of identifying invariant quantities that might exist in low level or early vision for tasks such as feature detection and detection matching. Some work in this direction has been carried out by Koenderink and DooRN [5], who recognized the importance of differential invariants associated with motion fields and Florack et. al [6], who proposed differential invariants for characterizing a number of image contour properties.

Our visual system is robust and can identify significant features even under widely varying conditions. Our interpretation of an image is largely unchanged even if the order of illumination is changed by several orders of magnitude. Similarly our interpretation of an image is largely unchanged by changes in spatial magnification, though

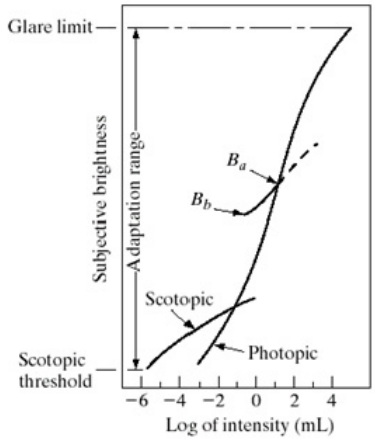


Figure 1.1: Image showing range of subjective brightness of human eye [1].

the degree of tolerance is not same as it for for illumination variance. Thus to detect image features of low level, illumination and contrast invariance are the main form of invariance needed and to some extent, image magnification. Finally, one has to decide whether a detected feature is an actual feature or not, depending on the value of quantity used for feature detection. Thus thresholding is needed. So, if one has an invariant measure of significance of features, the problem of deciding a threshold value is greatly eased.

The thresholding problem has plagued feature detection since a long time. Existing Gradient based edge detection methods developed by Sobel(1969), Marr and Hildreth [7], Canny [8], [9] and others are sensitive to variations in image illumination, blurring, and magnification. Empirical determination of image gradient values that correspond to significant edges is usually done. Efforts to determine threshold values automatically have not been been successful and such applicability of such methods is very limited [8], [10]. Developing feature detectors in spatial domain is difficult because it is hard to avoid characteristics, such as intensity gradients, contrast levels, or equivalent quantity, local to image.

Since development of invariant feature detectors is difficult in spatial domain, we migrate to frequency domain for developing invariant feature detectors. Morrone *et al.* [11] and Morrone and Owens [12] has developed model of feature perception called local energy model. This model postulates that features are perceived at points in an image where Fourier components are maximally in phase.

A lot of work on the local energy model has been done. Morrone and Burr [13] showed that the local energy model successfully explains psychological effects in human feature perception. Owens *et al.* [14] have proposed an edge detector that when applied to its own output produces no further change. Such edge detectors are projections in mathematical sense. They showed that the energy feature detector is a true projection and does not proliferate edges when applied to a line drawing. Venkatesh and Owens [15] examined feature classification based on local energy detection and showed that local energy is intrinsically capable of classifying features because of the use of odd and even filters. Feature classification allows for the elimination of certain types of features from the edge map, simplifying the task of object recognition.

Owens [16] demonstrated that points in image, at which local energy function has a local maximum, are stable with respect to large class of image variations. Morrone *et al.* [17] proposed a novel method for scale selection used in edge detection, where the scale size varied dynamically with the convolution output, i.e., the the stronger the output, the smaller the spatial scale. Robbins and Owens [18] proposed a method for detection of 2D image features that relied upon maximal 2D order in the phase domain of the image signal. Points of maximal phase congruency correspond to all the different types of 2D features.

The work done so far in the direction of feature detection from local energy model depends mostly on finding points of maximal phase congruency from maxima in local energy. Local energy is a dimensional quantity, proportional to phase congruency. However, local energy is dependent on local contrast. Thus, to identify whether a local energy value corresponds to feature is again dependent on the choice of threshold value.

Kovesi [19] proposed a method to find Phase Congruency, a dimensionless measure for feature detection, independent of local contrast. However, Phase Congruency has not been successfully used for feature detection because of the following reasons:

1. Since Phase Congruency is a normalized quantity, it is highly sensitive to noise.

2. The existing methods of finding Phase Congruency is ill-conditioned if all frequency components are very small, or if there is only one frequency component

present in the signal.

3. The existing methods for calculating Phase Congruency do not provide good localisation of features.

Sensitivity of Phase Congruency to noise is the biggest problem associated with Phase Congruency. This thesis aims to detect Phase Congruency directly, without using the notation of Local Energy. In this work, a method has been proposed to calculate Phase Congruency using S-Transform. Since, the proposed method is dependent on S-Transform, it provides good localization of features. Moreover, the way Phase Congruency is calculated, makes it independent of trivial cases of having only one or very few Fourier components. As far as noise is considered, the proposed method can also be combined with the denoising techniques while calculating Phase Congruency. However, such denoising techniques would work only for additive noises.

This thesis is organized as follows. In Chapter 2, we introduce the S-Transform and its properties. We then discuss the advantages of S-Transform over other multiresolution techniques. In Chapter 3, we discuss feature detection algorithm based on Phase Congruency. We begin this chapter with the definition of Phase Congruency. We describe the existing work on Phase Congruency briefly. It is followed by the proposed algorithm, described in detail with diagram. Finally, Results and Simulation are presented in Chapter 4. In section 4.1 of this chapter, we provide the comparative analysis of the proposed algorithm with the existing edge detection techniques. In section 4.2, application of the proposed method for feature detection is discussed in the field of biometrics. Finally, Chapter 5 presents the concluding remarks, with the scope for further research work.

Chapter 2

S-Transform

2.1 Signals and their types

A signal is a function that conveys information about the behavior or attributes of some phenomenon. In the context of image processing, a signal is a physical quantity which varies with space and contains information about space.

Signals may broadly be classified into the following two types:

1. Stationary Signals

2. Non-Stationary Signals

Stationary signal are the signals which have all the frequency components present at all times of the signal. Non-stationary signals are the signals in which all the frequency components are not present at all the times in the signal.

An example of stationary signal is shown in Figure 2.1(a). This signal has all frequency components at all points. This signal is given by

$$f(t) = cos(2\pi 10t) + cos(2\pi 25t) + cos(2\pi 50t) + cos(2\pi 100t) \tag{2.1}$$

To analyze this type of signal, one can simply apply Fourier transform and get all the frequency components of the signal (See Figure 2.1(c)). Since, all frequency components are present at all times, the original signal can be successfully reconstructed from inverse of Fourier transform.

An example of a non-stationary signal is given in Figure 2.1(b). This signal has four frequency components, whose lifespan is disjoint in time domain. The four frequency components contained in the signal are $10Hz, 25Hz, 50Hz$, and $100Hz$. Each of these frequency is present for a duration of 0.25 seconds and only one frequency component is present at any instant of time. Fourier transform is applied on this signal and is shown in Figure 2.1(d). Since, Fourier transform gives information only about the frequencies contained in the signal, and not about the time at which these frequencies are present in the signal, it cannot be used to reconstruct the original signal.

An image is a non-stationary signal. Image consists of edges which divide the it into regions. Smooth regions in the image have dominant low frequency components while edges have dominant high frequency components. Since, an image neither consists only of smooth regions, nor only of edges, but a mixture of both, an image is essentially a non-stationary signal.

To analyze a non-stationary signal such as image, we need multi-resolution techniques. Multi-resolution techniques give us time-frequency representation (TFR). TFR can be used to deduce the information about which frequency components are present at what time in the original signal.

Many multi-resolution techniques exist. Some of them are :

1. Short Time Fourier Transform

2. Wavelet Transform

3. S-Transform

S-Transform has many advantages over Short Time Fourier Transform and Wavelet Transform. A detailed discussion on this is presented in Section 2.5.

2.2 Derivation of S-Transform

The S-Transform was developed by R. G. Stockwell [20]. It is used to perform multi-resolution analysis on signals and it gives very good Time-Frequency Representation (TFR). It gives information about all the Fourier components that are present at a given point in a signal.

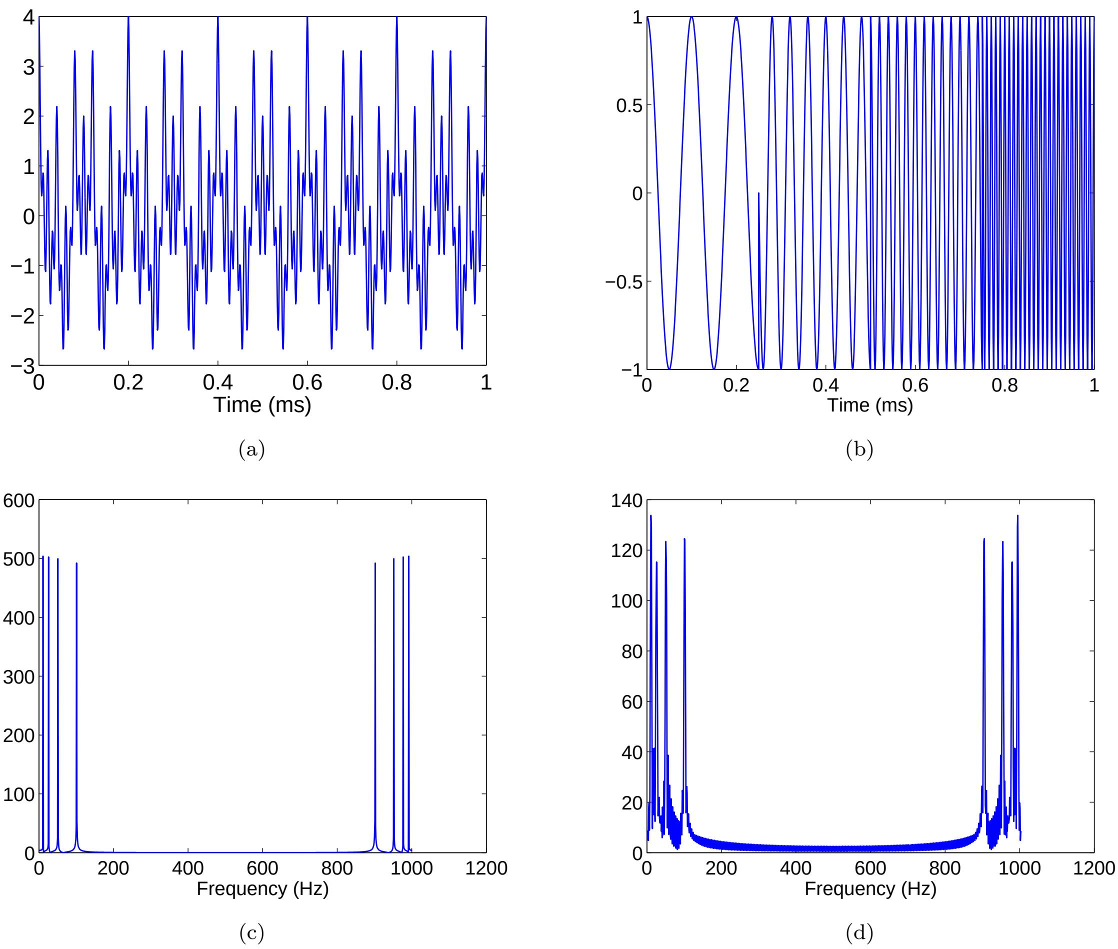


Figure 2.1: (a) Stationary signal, (b) Non-Stationary Signal, (c) Fourier Transform of Stationary Signal in (a), and (d) Fourier Transform of Stationary Signal in (b).

The S-Transform for continuous 1-dimensional signal $h(t)$ is given by :

$$S(\tau, f) = \int_{-\infty}^{\infty} h(t) \frac{| f |}{\sqrt{2\pi}} e^{-\frac{(\tau-t)^2 f^2}{2}} e^{-2i\pi f t} dt \tag{2.2}$$

S-Transform is an extension of Short Time Fourier Transform and Wavelet Transform and can be derived from both.

2.2.1 Derivation from Short Time Fourier Transform

The Fourier transform of continuous 1-dimensional signal h(t) is given by:

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-i2\pi f t} dt \tag{2.3}$$

If the time series $h(t)$ is windowed (or multiplied point by point) with a window function $g(t)$, then the resulting spectrum is

$$H(f) = \int_{-\infty}^{\infty} h(t) g(t) e^{-i2\pi f t} dt \tag{2.4}$$

The S-Transform can be found by first defining a particular window function, a normalized Gaussian

$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{-t^2}{2\sigma^2}} \tag{2.5}$$

and then allowing the Gaussian to be a function of translation τ and dilation (or window width) σ

$$S^*(\tau, f, \sigma) = \int_{-\infty}^{\infty} h(t) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(t-\tau)^2}{2\sigma^2}} e^{-i2\pi f t} dt \tag{2.6}$$

which for a particular value of σ is similar to the definition of Short Time Fourier Transform, given by,

$$STFT(\tau, f) = \int_{-\infty}^{\infty} h(t) w(t-\tau) e^{-i2\pi f t} dt \tag{2.7}$$

Taking width of the window σ to be proportional to the inverse of the frequency

$$\sigma(f) = \frac{1}{| f |} \tag{2.8}$$

one gets the S-Transform as :

$$S(\tau, f) = \frac{| f |}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(t) e^{-\frac{(t-\tau)^2 f^2}{2}} e^{-i2\pi f t} dt \tag{2.9}$$

2.2.2 Derivation from Wavelet Transform

The Continuous Wavelet Transform can be defined as a series of correlations of the time series with a function called a wavelet:

$$W(\tau, d) = \int_{-\infty}^{\infty} h(t) w(t - \tau, d) dt \tag{2.10}$$

The S-Transform of a function h(t) can be derived by a CWT with a specific mother wavelet multiplied by a phase factor

$$S(\tau, f) = e^{i2\pi f \tau} W(\tau, d) \tag{2.11}$$

where the mother wavelet is defined as

$$w(t, f) = \frac{| f |}{\sqrt{2\pi}} e^{-\frac{t^2 f^2}{2}} e^{-i2\pi f t} \tag{2.12}$$

2.3 Properties of S-Transform

1. **Absolutely Referenced Phase Information** : The phase factor $e^{i2\pi f t}$ in Equation 2.11 helps to get absolutely referenced phase information. This phase factor splits the mother wavelet into two parts, Gaussian window and oscillatory exponential kernel $e^{-i2\pi f t}$. The kernel remains stationary while Gaussian window moves. Kernel being stationary, localizes the real and imaginary components of spectrum independently, thus localizing amplitude and phase of spectrum independently.

2. **Relation to Fourier Transform** : The S-Transform is related to Fourier transform in the following way:

$$H(f) = \int_{-\infty}^{\infty} S(\tau, f) d\tau \tag{2.13}$$

Thus, this relationship can be used to calculate Inverse S-Transform.

$$h(t) = \int_{-\infty}^{\infty} \{\int_{-\infty}^{\infty} S(\tau, f) d\tau\} e^{i2\pi f t} df \tag{2.14}$$

3. **Instantaneous Frequency** : An extension of instantaneous frequency is provided by the S-Transform. S-Transform can be written in polar notation as

$$S(\tau, f) = A(\tau, f) e^{\Phi(\tau, f)} \tag{2.15}$$

where,

$$A(\tau, f) = \sqrt{Real(S(\tau, f)) + Im(S(\tau, f))} \tag{2.16}$$

and

$$\Phi(\tau, f) = tan^{-1}\{\frac{Im(S(\tau, f))}{Real(S(\tau, f))}\} \tag{2.17}$$

Thus, Instantaneous Frequency (IF) is given by,

$$IF(\tau, f_0) = \frac{1}{2\pi} \frac{d}{d\tau} \{2\pi\tau f_0 + \Phi(\tau, f_0)\} \tag{2.18}$$

4. **Linearity** : S-Transform is a linear operation. Thus,

$$ST\{g(t) + h(t)\} = ST\{g(t)\} + ST\{h(t)\} \tag{2.19}$$

Proof of Linearity :

$$ST\{g(t) + h(t)\} = S(\tau, f) = \frac{\mid f \mid}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \{g(t) + h(t)\} e^{-\frac{(t-\tau)^2 f^2}{2}} e^{-i2\pi f t} dt \tag{2.20}$$

which can be rewritten as

$$\begin{aligned} ST\{g(t) + h(t)\} = S(\tau, f) &= \{\frac{\mid f \mid}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(t) e^{-\frac{(t-\tau)^2 f^2}{2}} e^{-i2\pi f t} dt\} \\ &+ \{\frac{\mid f \mid}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(t) e^{-\frac{(t-\tau)^2 f^2}{2}} e^{-i2\pi f t} dt\} \end{aligned} \tag{2.21}$$

$$= \quad ST\{g(t)\} + ST\{h(t)\}$$

Thus,

$$ST\{g(t) + h(t)\} = ST\{g(t)\} + ST\{h(t)\} \tag{2.22}$$

2.4 Discrete S-Transform

The S-Transform of a discrete 2-dimensional signal $f(x, y)$ is given by:

$$S(x, y, k_x, k_y) = \sum_{\alpha=0}^{M-1} \sum_{\beta=0}^{N-1} F(\alpha + k_x, \beta + k_y) e^{-2\pi^2 (\frac{\alpha^2}{k_x^2} + \frac{\beta^2}{k_y^2})} e^{2\pi i(\alpha x + \beta y)} \tag{2.23}$$

Here,

- x corresponds to x-coordinate in space.
- y corresponds to y-coordinate in space.
- k_x corresponds to frequency along x-axis.
- k_y corresponds to frequency along y-axis.
- F is the Fourier transform of original image.

The algorithm to compute 2-dimensional S-Transform [21] of an image is given by Algorithm 1.

Algorithm 1: Compute2DST

Data: I: Input Image, M: Rows, N: Columns

Result: S: Resultant S-Transformed matrix on I

1 $F(\alpha, \beta) \leftarrow FFT(I(x, y))$;

2 **forall the** $(k_x, k_y)(k_x, k_y \neq 0)$ **do**

3 Compute the Frequency domain Gaussian localizing window at the current frequency $W(\alpha, \beta) \leftarrow (k_x, k_y) : e^{-2\pi^2(\frac{\alpha^2}{k_x^2}+\frac{\beta^2}{k_y^2})}$;

4 Shift the Fourier Spectrum $F(\alpha, \beta)$to$F(\alpha + k_x, \beta + k_y)$;

5 Compute the point-wise multiplication of $F(\alpha + k_x, \beta + k_y)$ and $W(\alpha, \beta)$, and denote it as $M_{k_x,k_y}(\alpha, \beta)$;

6 $S_{k_x,k_y}(x, y) \leftarrow IFFT(M_{k_x,k_y}(\alpha, \beta))$;

7 For the frequencies $(k_x, 0) and (0, k_y)$, the Gaussian window function becomes $e^{-2\pi^2\frac{\alpha^2}{k_x^2}}$ and $e^{-2\pi^2\frac{\beta^2}{k_y^2}}$ respectively. And compute steps 4-6. ;

8 For the frequency $(0, 0)$, $S_{0,0}(x, y) \leftarrow mean\{I(x, y)\}$;

2.5 Advantages of S-Transform

1. The Short Time Fourier Transform (STFT) has a fixed resolution but S-Transform gives a good time resolution for high frequency components and good frequency resolution for low frequency components, which is best suited for images. S-Transform is equivalent to applying several STFT with different sized windows. Thus, S-Transform is superior to STFT.

2. Wavelet Transform gives phase information local to translated window but S-Transform gives absolutely referenced phase information, which can be used for evaluating phase congruency. It has already been explained in Section 2.3, Property 1.

3. S-Transform can be used for denoising images containing additive noise. For this purpose, we can use the linearity property of S-Transform described in Section

2.3, Property 4.

4. S-Transform is directly related to Fourier Transform but Wavelet Transforms are not related to Fourier Transform. Relationship between S-Transform and Fourier Transform has already been explained in Section 2.3, Property 2. Thus, S-Transform is invertible but not all Wavelet Transforms are invertible.

5. S-Transform also provides superior time resolution compared to wavelet resolution.

2.6 Summary

The S-Transform is more powerful that other multi-resolution techniques like STFT and Wavelet Transform. The phase of the S transform referenced to the time origin provides useful and supplementary information about spectra that is not available from locally referenced phase information in the CWT [20]. The major disadvantage of S-Transform is its very high computational time complexity which makes it impractical in many cases.

Chapter 3

Novel Feature Detection based on Phase Congruency

Images are characterized by features such as edges and object. At the edges and boundary of objects, the Fourier components of the images are in same phase. Alternatively, we can say that edges and objects, can be characterized by the phase of the Fourier components.

For example, consider square wave and its few Fourier components given in Figure 3.1. If we observe the square wave and its Fourier components, we will see that at the rising edge of the square wave, all the Fourier components are rising, i.e., they have a phase value of zero radians. We also see that at the falling edge of the square wave, the Fourier components are falling, i.e., they all have a phase value of π radians.

Similarly, if we consider a triangular wave and its Fourier components, we observe

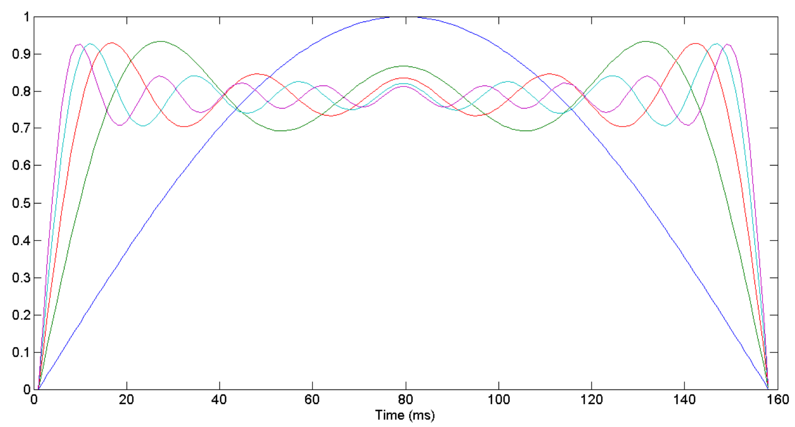


Figure 3.1: Few Fourier components of a square wave are shown.

that at the peak of the triangular wave, all the Fourier components of the wave are at their peak and have the same phase.

Thus, from these two examples, we can conclude that features of images or signals can be characterized by phase similarity of the Fourier components.

3.1 Phase Congruency

Phase Congruency is defined as the measure of degree of similarity of phase of Fourier components of the signal. It varies from zero to one. A Phase Congruency of zero implies that Fourier components of the signal are completely out of phase. And a Phase Congruency value of one implies that all the Fourier components of the signal have same phase.

3.2 Existing Work

The existing work on Phase Congruency by Kovesi [19] uses Wavelet Transform to calculate Phase Congruency. It uses Gabor Filters to calculate Phase Congruency. According to his work, Phase Congruency at a point x in the signal is defined as :

$$PC(x) = \frac{\Sigma_o (E_o(x) - T_o)^+}{\epsilon + \Sigma_o \Sigma_n A_{no}(x)} \tag{3.1}$$

where,

1. $E_o(x)$ is the energy along an orientation. It is calculated as :

$$E_o(x) = \sqrt{F_o^2(x) + G_o^2(x)} \tag{3.2}$$

$$F_o(x) = \Sigma I_o(x) * M_n^e(x) \tag{3.3}$$

$$G_o(x) = \Sigma I_o(x) * M_n^o(x) \tag{3.4}$$

   $M_n^e(x)$ and $M_n^o(x)$ are even and odd components of wavelet and $I_o(x)$ is given signal along a particular orientation.

2. T_o is the noise correction factor given by :

$$T_o = kA_1 \frac{1 - m^{-n}}{1 - m^{-1}} \tag{3.5}$$

and

$$A_1 = e^{\overline{log A_0(x,y)}} \tag{3.6}$$

3. ϵ is used to check the condition when only 1 Fourier component is present at a point.

4. Amplitude term $A_{no}(x)$ is given by :

$$\Sigma_o \Sigma_n A_{no}(x) = \Sigma_o \Sigma_n \sqrt{I_o(x) * M_n^e + I_o(x) * M_n^o} \tag{3.7}$$

3.3 Proposed Approach

We propose a different approach to compute Phase Congruency based on S-Transform. Since, the local energy is dependent on image characteristics such as illumination, contrast, etc., we propose a method to compute Phase Congruency that does not involve the calculation of local energy. We bypass the entire step of calculating local energy at each point. Instead, we first apply S-Transform locally to each point in the image. S-Transform gives us all the Fourier components at a particular point. We then calculate the phase value for each Fourier component at the point and take the standard deviation of phase values as the measure of Phase Congruency. The working of proposed method for edge detection is illustrated in Figure 3.2.

Since, the proposed algorithm works locally, it will make the feature detection process translation invariant. Moreover, the proposed algorithm 2 is also saved form the trivial cases of having only one frequency component. If a point has only one Fourier component, then S-Transform will associate high value to the Fourier component present at the point and all other Fourier components at the point would be associated with low value. But when we apply standard deviation to compute Phase Congruency, the value of standard deviation becomes high leading to low Phase Congruency. Figure 3.3 depicts for the edges detected using proposed method.

With slight modifications, the proposed algorithm can also be used to get the orientation information of each feature point. The modified algorithm is given in Algorithm 4. Figure 3.4 shows a fingerprint and its orientation information using Algorithm 4, i.e., $ModifiedPCImage$. With the orientation information of features

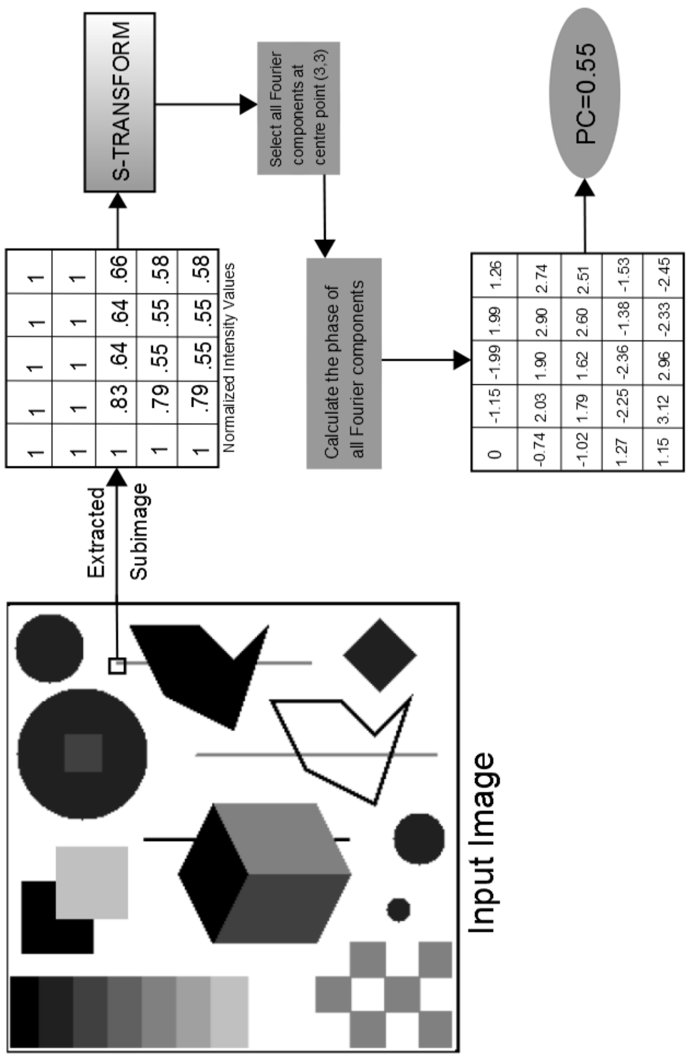
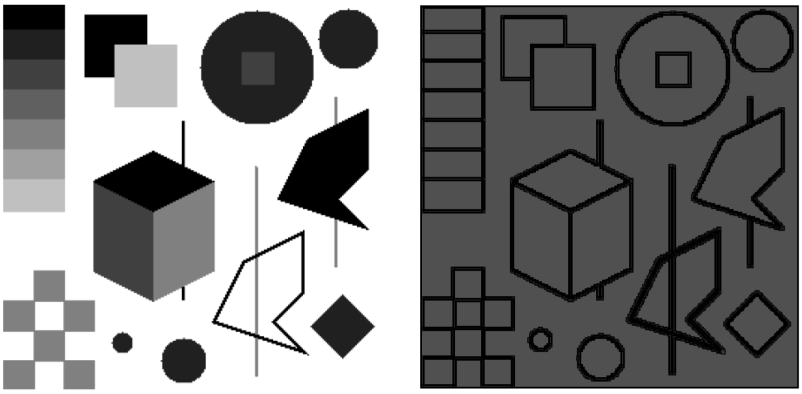


Figure 3.2: Figure showing the working of the proposed model.



(a) (b)

Figure 3.3: (a) An input image (b) Edges detected in the input image (a) using algorithm 2, i.e., ComputePCImage.

in the image, the detected features can be made rotation invariant with the concept of binning.

3.3.1 Working of Proposed Algorithm *ComputePCImage*

In, Algorithm 2, we mentioned the steps required to compute the Phase Congruency (PC) of an image. The algorithm takes an input image I of size $M \times N$ and returns an image C of same size, containing the Phase Congruency values for all points in the input image I. In step 1, we simply normalize the input image. We do normalization by dividing the intensity value in each pixel by the maximum possible intensity value. In case of 8 bit gray scale images, we use value 255 for normalization.

In step 2, we add some intensity (say, 0.1) to each point and then re-normalize it by value (1 + added intensity). The motivation behind this step is that a point with zero intensity has zero energy. And zero energy implies that no Fourier components exist at that point. So, at points with zero intensity, all Fourier components will have zero magnitude, thus implying that the phases of all the components would be same. If phases of all Fourier components are same, then by definition, Phase Congruency will be 1. Thus, such points would be falsely detected as feature points. So, to avoid detecting such points as feature points, we add some intensity at each point.

In step 3, we create an image C, which contains the Phase Congruency values for each point of the image. Since, we want to emphasize on edges and would be

(a) (b)

Figure 3.4: (a) Image of a Fingerprint, (b) Orientation Information of fingerprint image in (a) using Algorithm 4, i.e., ModifiedPCImage.

Algorithm 2: ComputePCImage

Data: I: Input Image, M: Rows, N: Columns

Result: C: Resultant Image with Detected Features, M: Rows, N: Columns

```
Normalize I(x, y), i.e, convert all intensity values between (0, 1) ;
Add some intensity value to all points of normalized image and then
re-normalize it. Let this image be J(x, y) ;
Create and initialize image C(x, y) to contain zeros ;
Define local window size ;
Define core points of J(x, y) to be those points on which if window is placed,
then window will not cross J(x, y) ;
forall the (x, y) in core points of J(x, y) do
    max ← 0 ;
    forall the rotangle in 0 to 2π increment σ do
        W(x, y) ← local window of J(x, y) in the direction of rotangle ;
        STW(x, y, kx, ky) ← STransform{W(x, y)} ;
        temp ← ComputePC{STW(x_0, y_0, kx, ky)} ;
        if temp < max then
            max ← temp ;
    C(x, y) ← max ;
```

representing them by the high intensity values, we initialize the image C with dark intensity values.

Algorithm 3: ComputePC

Data: I: Input Image, M: Rows, N: Columns

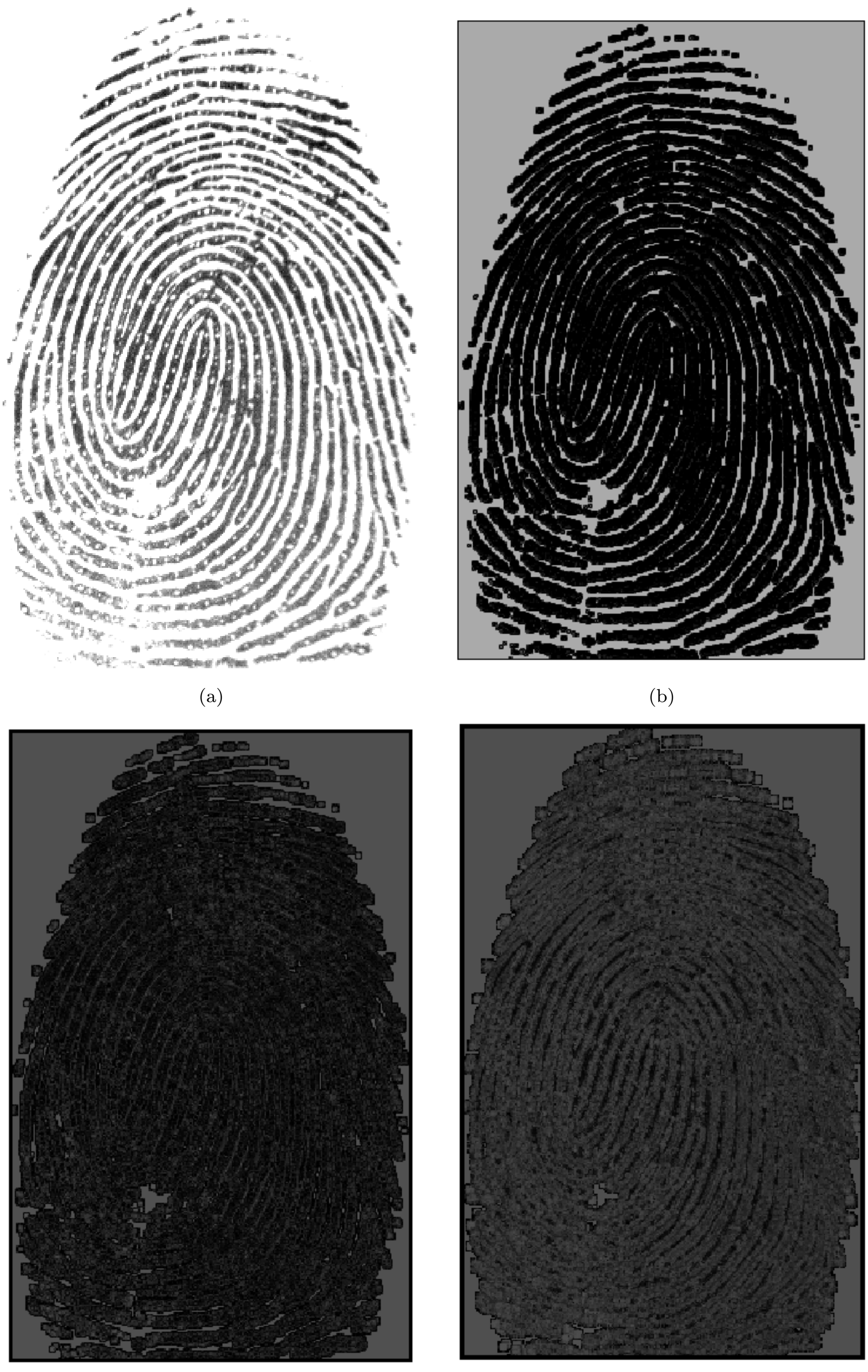
Result: Phase Congruency value, *pcval* for given Image

1 $minstd \leftarrow 1$;
2 $\Phi(k_x, k_y) \leftarrow Phase\{I(k_x, k_y)\}$;
3 **for** *each row r of* Φ **do**
4   Normalise r, i.e. $r \leftarrow r/\pi$;
5     $tempstd \leftarrow std(r)$;
6   **if** $tempstd < minstd$ **then**
7         $minstd \leftarrow tempstd$;
8 **for** *each column c of* Φ **do**
9   Normalise c, i.e. $c \leftarrow c/\pi$;
10     $tempstd \leftarrow std(c)$;
11   **if** $tempstd < minstd$ **then**
12         $minstd \leftarrow tempstd$;
13 $pcval \leftarrow (1 - minstd)$

Considering the efficiency parameters of the algorithm, and computational complexity of S-Transform, we would be applying S-Transform at each point locally. Applying S-Transform locally also provide us with the advantage that the feature points detected would be translation invariant. To apply S-Transform locally, we define a window size in step 4. We have worked with several window sizes and found out that 3×3 window size gives better performance as shown in Figure 3.5.

In step 5, we identify core points of the image. Boundary points of the image are all those points located near the boundary of the image, on which if local window is placed, it will cross the boundary of the image. All points of the image which are not boundary points are core points.

In step 6, we begin a loop, which will calculate the Phase Congruency for all core points in the image. The proposed algorithm does not calculate the Phase Congruency



(a) (b) (c) (d)

Figure 3.5: (a) Input fingerprint image to Algorithm 2, i.e., *ComputePCImage*, (b) Features detected using window size 3×3, (c) Features detected using window size 5×5, and (d) Features detected using window size 7×7. In (b),(c)&(d), we have taken negative of all the output images.

for boundary points. Since number of such points is very less, they can be ignored without significantly affecting the result.

In step 9, we extract the local window, W, at image location (x, y) of image J in the direction of orientation. Edges may oriented in any direction and they are detected when we apply the edge detection algorithm in a direction perpendicular to their direction of orientation. So, to make detection algorithm robust, we take local window at each position in the several possible direction of orientation. In step 10, we apply S-Transform on the locally detected window, W and get an image STW. In step 11, we get the Phase Congruency value for STW.

In for loop (steps $8-13$), we calculate maximum PC value for each point along orientations $0, \sigma, 2\sigma \ldots 2\pi$ and store it in the image C in step 14.

3.3.2 Working of Proposed Algorithm *ComputePC*

Algorithm 3 takes an input image I representing different Fourier components contained at a position. This algorithm returns returns a PC value which is maximum along any row or column of the image I.

In step 4, we normalize each row to ensure that phase values lie between 0 and 1. Since, all the phase values would be between 0 and 1, standard deviation value will also lie between 0 and 1 and thus Phase Congruency value would be limited between 0 and 1. In step 5, we calculate the standard deviation for a particular row. In steps $6-7$, we compare the present value of standard deviation with the minimum value of standard deviation and update the minimum value, if it is greater. We repeat the process similarly for all columns in for loop(steps $8-12$). Since, standard deviation is a measure of dissimilarity among phase values and because we need a measure of similarity of phase values, we define Phase Congruency as $(1 - minstd)$.

Algorithm 4: ModifiedPCImage

Data: I: Input Image, M: Rows, N: Columns

Result: C: Resultant Image with Detected Features, O: Resultant image containing orientation values of points in image ,M: Rows, N: Columns

Normalize $I(x, y)$, i.e, convert all intensity values between $(0, 1)$;
Add some intensity value to all points of normalized image and then re-normalize it. Let this image be $J(x, y)$;
Create and initialize image $C(x, y)$ to contain zeros ;
Create image $O(x, y)$ to contain orientation values ;
Define local window size ;
Define core points of $J(x, y)$ to be those points on which ii window is placed, then window will not cross $J(x, y)$;
forall the (x, y) *in core points of* $J(x, y)$ **do**
    $max \leftarrow 0$;
    $orientation \leftarrow -1$ **forall the** *rotangle in* 0 *to* 2π *increment* σ **do**
        $W(x, y) \leftarrow$ local window of $J(x, y)$ in the direction of $rotangle$;
        $STW(x, y, kx, ky) \leftarrow STransform\{W(x, y)\}$;
        $temp \leftarrow ComputePC\{STW(x_0, y_0, kx, ky)\}$;
        if $temp < max$ **then**
            $max \leftarrow temp$;
            $orientation \leftarrow rotangle$;
    $C(x, y) \leftarrow max$;
    $O(x, y) \leftarrow orientation$;

Chapter 4

Simulation and Results

This chapter discusses the application of the proposed algorithms for edge detection. Section 4.1 provides a comparative analysis of the proposed algorithm 2 with the existing methods. Section 4.2 discusses the application of algorithm 2. This section proposes an Algorithm 5 to detect minutiae features from fingerprint and compares the result with an existing work.

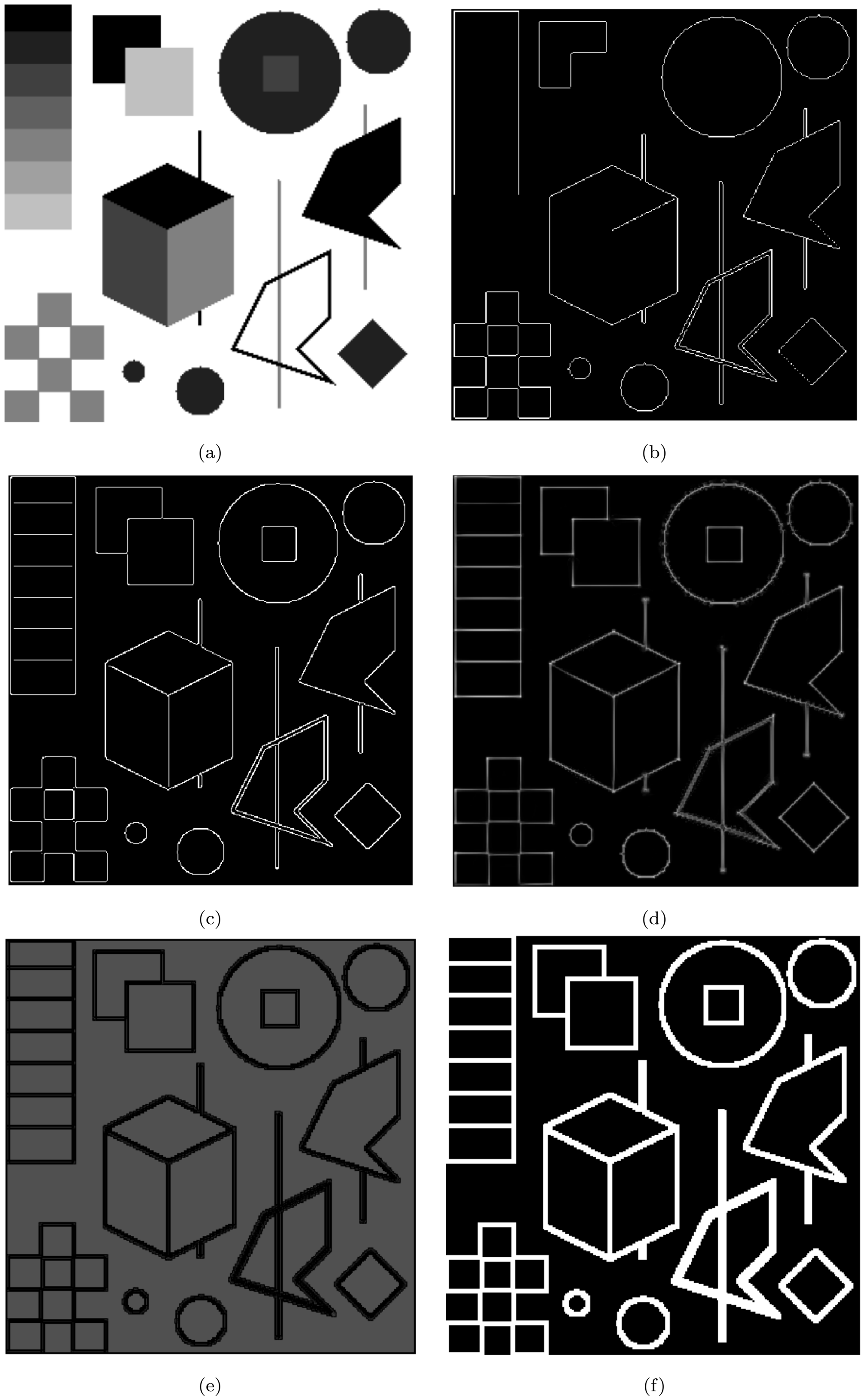
4.1 Analysis of Proposed Edge Detector

The performance of the proposed algorithm is demonstrated on three test images in this section. For comparison, the output of the Sobel (1969), Canny [9] and Kovesi [22], [23], [24], [25] are also presented. The purpose of this comparison is to illustrate some of the qualitative differences between the mentioned detectors. Canny edge detector used automatic values for thresholding. Kovesi's method used the following parameters : Local frequency is obtained using two octave bandwidth filters over four scales. Six number of orientations are used. Wavelength of smallest scale filter is 3 pixels. Scaling factor between successive filters is 2.1. Filters are constructed in frequency domain instead of creating them in spatial domain and transforming to frequency domain. Threshold used is 0.5. The sharpness of sigmoid function used to weight phase congruency for frequency spread is 10. The proposed method used a local window of size 5×5 and a threshold value of 0.7.

Figure 4.1 (a) is the input image to the various edge detection methods, i.e., Sobel, Canny, Kovesi's method, and the proposed method. Figure 4.1 (b) shows the edges detected by the Sobel operator. Clearly, Sobel operator fails to detect most of the edges. Figure 4.1 (c) shows the edges detected by the Canny's method. The problem with Canny edge detection algorithm is that for each edge in the original image, it detects two edges. Figure 4.1 (d) shows the output of the Kovesi's method [26]. In this image, one can notice that not all detected edges have equal strength. Moreover, some edges near corners are also dull. Figure 4.1 (e) shows the edges detected by Algorithm $ComputePCImage$. This image is the output of Algorithm $ComputPCImage$ applied on input image 4.1 (a). Here, edges are clearly demarked from the rest of the images. Moreover, if we just apply thresholding on this image, we get image 4.1 (f). This result is completely independent of the local image intensity. In Canny edge detection algorithm [8] and Kovesi's work [26], raw output is processed using non maximal suppression and hysteresis thresholding. Such techniques are dependent on the intensity values which might again defeat the entire purpose of detecting intensity invariant features. The Algorithm $ComputePCImage$ is completely independent of intensity of the original image.

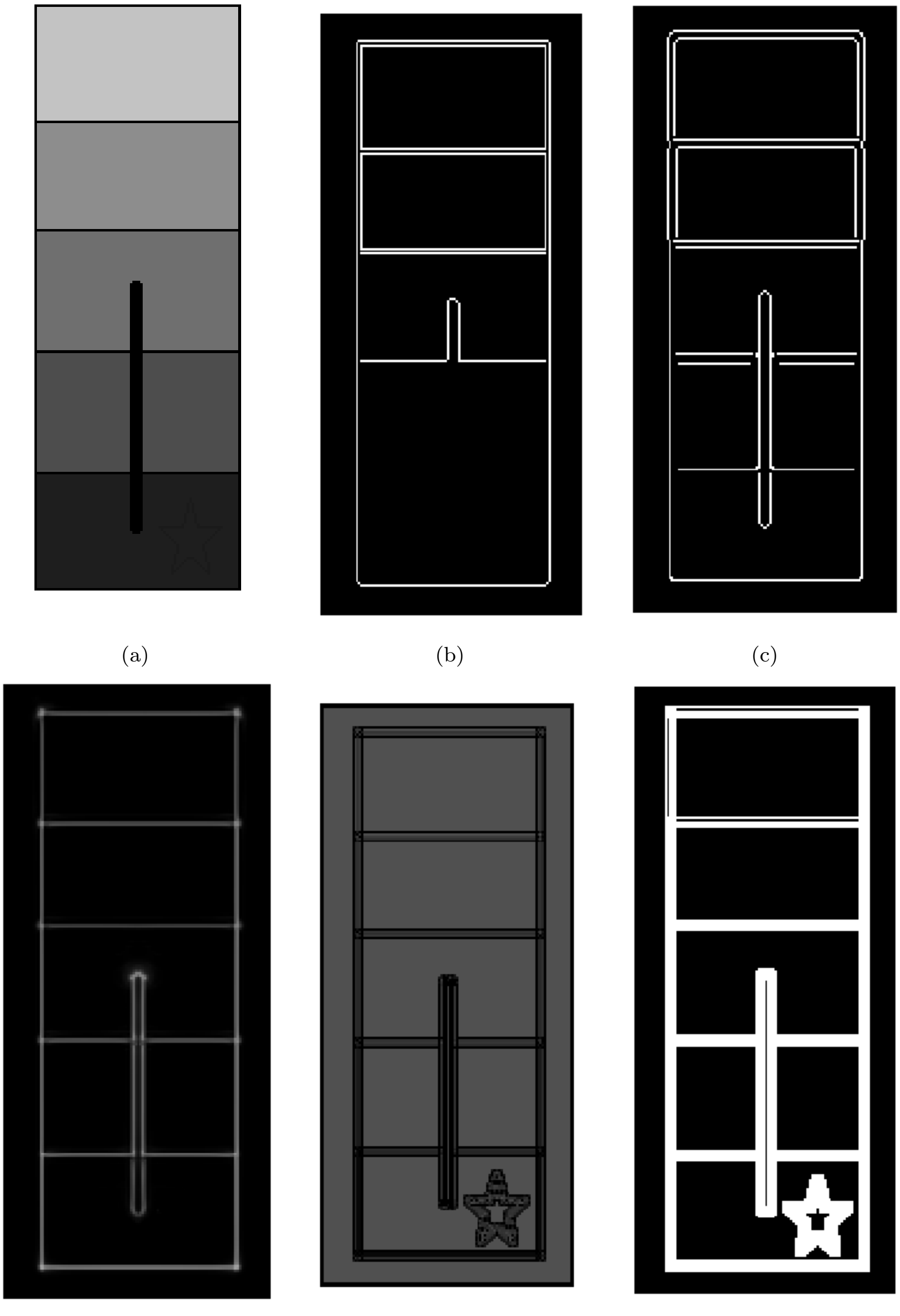
Figure 4.2 shows application of different edge detection algorithms on image containing subtle features. Figure 4.2 (a) is the input image. This image contains a star in the last box besides the line. The hollow star is marked by a boundary having very slight change in the intensity with background. The existing algorithms of Sobel, Canny, Kovesi fail to detect this star (See Fig 4.2 (b),(c),(d)). The proposed algorithm 2 easily detects the star contained in the last box (See Fig 4.2 (e)). We then apply a threshold value of 0.8 on Phase Congruency to get the thresholded image in Fig 4.2 (f). Thus, the proposed algorithm outperforms the existing algorithms when it comes to detect subtle features.

Figure 4.3 shows the application of various edge detection algorithms on the shaded input image. The input image in Figure 4.3 (a) is derived from shading the upper part of input image in Figure 4.1 (a). The Canny and Sobel edge detection algorithms fail to detect any feature in the dark region in the upper part of input image. The existing work of Kovesi is able to detect some features in the darker region. However, the proposed algorithm performs best in detecting features in the darker region. Here,



(a) (b) (c) (d) (e) (f)

Figure 4.1: (a) Input image, (b) Edges detected by the Sobel Operator, (c)Edges detected by Canny Edge Detection Algorithm, (d) Raw output image from Kovesi's Phase Congruency, (e) Raw Output image by the proposed algorithm, and (f) Output Image after thresholding applied to image (e).



(a) (b) (c)

(d) (e) (f)

Figure 4.2: (a) Input image, (b) Edges detected by the Sobel Operator, (c)Edges detected by Canny Edge Detection Algorithm, (d) Raw output image from Kovesi's Phase Congruency, (e) Raw Output image by the proposed algorithm, and (f) Output Image after thresholding applied to the image (e).

some of the features of the input image 4.1 (a) are lost during the shading (squares inside the circles in top portion of the image) and no edge detection algorithm can detect those features without apriori knowledge of the features. The proposed algorithm being too sensitive also detects the slight variation in intensity in the middle of image as feature.

4.2 Application in Biometrics

Biometrics is science of establishing identity of an individual based on certain unique characteristics which are possessed only by the individual. Biometrics provide solution to identity management to recognise individual [27]. The basic advantage of biometrics is that, it can't be stolen,forgotten or misplaced. Moreover, the biometric systems are difficult to fool, since the traits needed for such system belong to a person uniquely. Some traditional and biometrics systems used for authentication are shown in Figure 4.4. The underlying functioning of most of the biometric systems is input image from user, preprocessing of the image to find region of interest, feature extraction, and authentication of individual [28].

We tried to apply the proposed algorithm 4 for minutiae extraction from fingerprints. We proposed an Algorithm 5 for minutiae extraction. The algorithm does not need most of the pre-processing techniques used in traditional minutiae detection algorithms. It replaces all the pre-processing techniques by the proposed algorithm 4.

We applied the algorithm 5 on the fingerprint. We compared it with the algorithm [2]. The results are shown in Figure 4.5.

We perform a qualitative analysis of the proposed minutiae extraction algorithm with [2]. We observe that the proposed algorithm 5 perform better. It can detect more number of bifurcations. The Algorithm [2] detects all the features on the edges itself. However, our algorithm performs better.

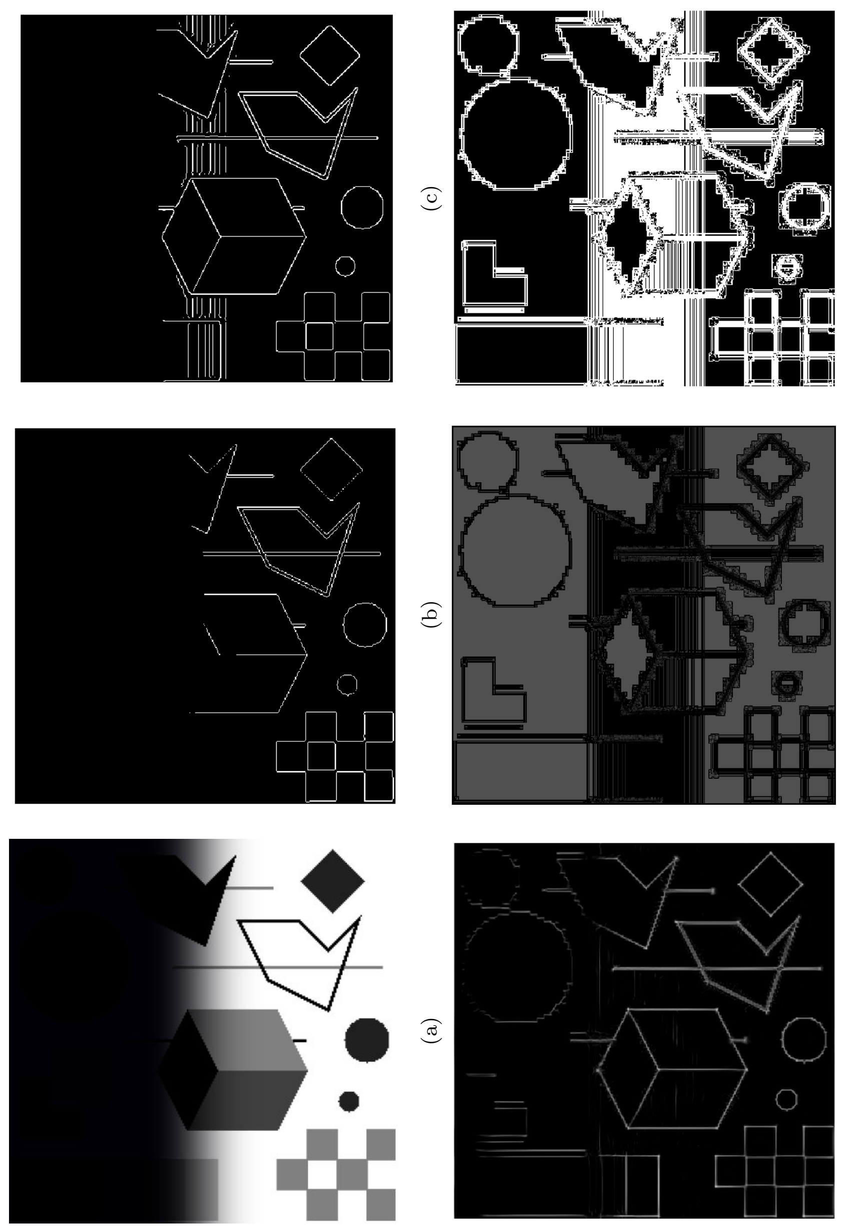


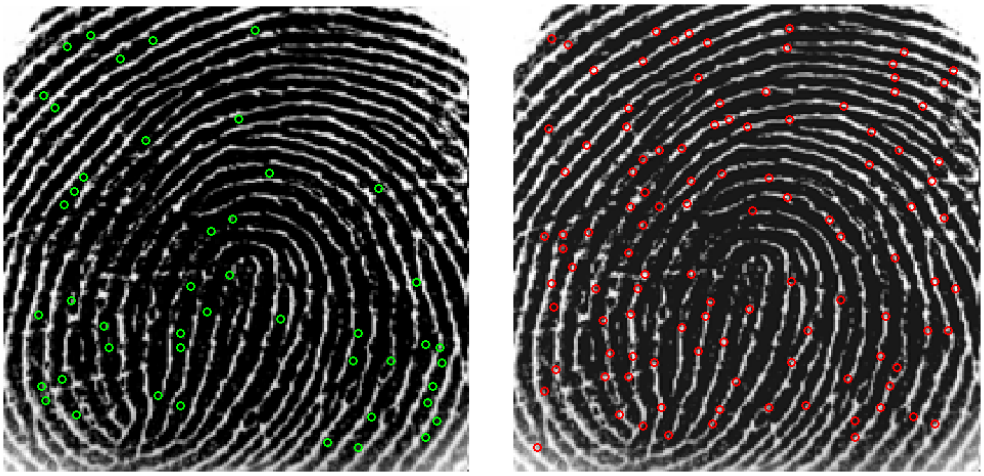
Figure 4.3: (a) Input image, (b) Edges detected by the Sobel Operator, (c)Edges detected by Canny Edge Detection Algorithm, (d) Raw output image from Kovesi's Phase Congruency, (e) Raw Output image by the proposed algorithm, and (f) Output Image after thresholding applied to the image (e).

Figure 4.4: Various forms of authentication. Traditional methods of authentication using token based and knowledge based approaches (left). Use of biometrics to claim identity (right)

Algorithm 5: ComputeMinutiae

Data: I: Input Image, M: Rows, N: Columns
Result: Minutiae Points with Orientation

Apply Algorithm 4 on input image I, i.e, $(PCImage, OrientImage) \leftarrow ModifiedPCImage\{I\}$;
Threshold the image $PCImage$, i.e, $binaryPC \leftarrow Thresholding\{PCImage\}$;
forall the (x, y) *in* $binaryPC$ **do**
    if $binaryPC(x, y) = 1$ **then**
        if $No.\ of\ 1 - value\ neighbors = 1$ **then**
            Mark (x, y) as termination point;
        if $No.\ of\ 1 - value\ neighbors >= 3$ **then**
            Mark (x, y) as bifurcation point;
Filter spurious minutiae using Euclidean distance. If $distance < threshDist$, minutiae is spurious;
Remove extreme minutiae using region of interest;
Extract the orientation information of remaining minutiae points from $OrientImage$;
Return the remaining minutiae points with their orientation;



(a) (b)

Figure 4.5: (a) Minutiae detected by Algorithm [2], (b) Minutiae detected by the algorithm 5 on same input image as (a).

Chapter 5

Conclusion and Future Work

This thesis proposes novel feature detection technique based on Phase Congruency using S-Transform. The first contribution is made to develop an approach for efficient feature detection using S-Transform. The proposed approach is powerful enough to extract the subtle features as well. The second contribution is made to modify the proposed algorithm to extract the orientation information of the detected features. This information is very useful for various applications in biometrics. Finally, the application of the proposed algorithm is shown on fingerprint minutiae extraction.

To conclude with thesis, the proposed work have been critically analyzed and few limitations have been observed. Further research work may be carried out on these limitations to improve the proposed work. The complexity of S-Transform poses a serious challenge from computational point of view. Thus, there is a stringent requirement to reduce the complexity of S-Transform. There is scope to try Discrete Orthonormal S-Transform (DOST) instead of S-Transform in the proposed approach. For feature detection in noisy images, the proposed algorithm may not perform well (due to its sensitivity), so appropriate denoising techniques need to be applied before applying the proposed algorithm. Such denoising technique can be developed during the implementation of the S-Transform to improve performance without increasing the overall complexity.

Bibliography

[1] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Dorling Kindersley India Pvt. Ltd., New Delhi, India, 2012.

[2] Florence Kussener. Fingerprint extraction. http://www.mathworks.in/matlabcentral/fileexchange/16728-fingerprint-application/content/FingerPrint/html/fingerprint.html, 2007.

[3] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transaction on Information Theory*, 8:179–187, 1962.

[4] Joseph L. Mundy and Andrew Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[5] Jan J. Koenderink and VAN DooRN. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22:773–791, 1975.

[6] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. General intensity transformations and differential invariants. *Journal of Mathematical Imaging and Vision*, 4:171–187, 1994.

[7] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of Royal Society of London*, 207:187–217, 1980.

[8] John Francis Canny. Finding edges and lines in images. Technical report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1983.

[9] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[10] M. K. Kundu and S. K. Pal. Edge detection based on human visual response. *International Journal of Systems Science*, 19:2523–2542, 1988.

[11] M. C. Morrone, J. Ross, D. C. Burr, and R. Owens. Mach bands depend on spatial phase. *Nature*, 324:250–253, 1986.

[12] M. Concetta Morrone and R. A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303–313, 1987.

[13] M. C. Morrone and D. C. Burr. Feature detection in human vision : a phase-dependent energy model. *Proceedings of the Royal Society of London*, 235:221–245, 1988.

[14] R. Owens, S. Venkatesh, and J. Ross. Edge detection is a projection. *Pattern recognition letters*, 9:233–244, 1989.

[15] Svetha Venkatesh and Robyn Owens. On the classification of image features. *Pattern recognition letters*, 11:339–349, 1990.

[16] R. A. Owens. Feature-free images. *Pattern Recognition Letters*, 15:35–44, 1994.

[17] M. Concetta Morrone, Anacleto Navangione, and David Burr. An adaptive approach to scale selection for line and edge detection. *Pattern Recognition Letters*, 16:667–677, 1995.

[18] B. Robbins and R. Owens. 2d feature detection via local energy. *Image and Vision Computing*, 15:353–368, 1997.

[19] Peter Kovesi. Phase congruency : A low-level image invariant. *Psychological Research*, 64:136–148, 2000.

[20] R. G. Stockwell, L. Mansinha, and R. P. Lowe. Localization of he complex spectrum : The s transform. *IEEE Transactions on Signal Processing*, 44:998–1001, 1996.

[21] H.Zhu, R. A. Brown, R. J. Villanueva-Oller, M. L. Lauzon, J. R. Mitchell, and A. G. Law. Progressive imaging : S-transform order. *ANZIAM Journal*, 45:1002–1016, 2003.

[22] Peter Kovesi. Symmetry and asymmetry from local phase. *Tenth Australian Joint Conference on Artificial Intelligence.*

[23] Peter Kovesi. Image features from phase congruency. *A Journal of Computer Vision Research*, 1, 1999.

[24] Peter Kovesi. Edges are not just steps. *The Fifth Asian Conference on Computer Vision*, pages 822–827, 2002.

[25] Peter Kovesi. Phase congruency detects corners and edges. *The Australian Pattern Recognition Society Conference*, pages 309–318, 2003.

[26] Peter Kovesi. Image features from phase congruency. Technical report, NedLands, W.A. 6009, 1995.

[27] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:4–20, 2004.

[28] A. K. Jain, P. Flynn, and A. A. Ross. *Handbook of Biometrics.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.