Spatial and Multi-Temporal Visual Change Detection with Application to SAR Image Analysis

by

Qian Xu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2014 by the
Graduate Supervisory Committee:

Lina Karam, Chair
Chaitali Chakrabarti
Daniel Bliss
Cihan Tepedelenlioglu

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

Thousands of high-resolution images are generated each day. Detecting and analyzing variations in these images are key steps in image understanding. This work focuses on spatial and multi-temporal visual change detection and its applications in multi-temporal synthetic aperture radar (SAR) images.

The Canny edge detector is one of the most widely-used edge detection algorithms due to its superior performance in terms of SNR and edge localization and only one response to a single edge. In this work, we propose a mechanism to implement the Canny algorithm at the block level without any loss in edge detection performance as compared to the original frame-level Canny algorithm. The resulting block-based algorithm has significantly reduced memory requirements and can achieve a significantly reduced latency. Furthermore, the proposed algorithm can be easily integrated with other block-based image processing systems. In addition, quantitative evaluations and subjective tests show that the edge detection performance of the proposed algorithm is better than the original frame-based algorithm, especially when noise is present in the images.

In the context of multi-temporal SAR images for earth monitoring applications, one critical issue is the detection of changes occurring after a natural or anthropic disaster. In this work, we propose a novel similarity measure for automatic change detection using a pair of SAR images acquired at different times and apply it in both the spatial and wavelet domains. This measure is based on the evolution of the local statistics of the image between two dates. The local statistics are modeled as a Gaussian Mixture Model (GMM), which is more suitable and flexible to approximate the local distribution of the SAR image with distinct land-cover typologies. Tests on real datasets show that the proposed detectors outperform existing methods in terms of the quality of the similarity maps, which are assessed using the receiver operating characteristic (ROC) curves, and in terms of the total error rates of the final change detection maps. Furthermore, we proposed a new similarity measure for automatic change detection based on a divisive normalization transform in order to reduce the computation complexity. Tests show that our proposed DNT-based change detector exhibits competitive detection performance while achieving lower computational complexity as compared to previously suggested methods.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Lina Karam for giving support, inspiration, strength, and encouragement throughout my Ph.D. study and life. Her patient guidance and stimulating research advice were essential for me to proceed through the doctorial program and complete my dissertation. She has also shown me how to be a strong and successful woman in the career. I have been feeling amazingly fortunate that I could work with Dr. Lina Karam in the past years.

Dr. Chaitali Chakrabarti has provided necessary assistance and encouragement. I am deeply grateful to her for giving me insightful suggestions and comments. I would like to extend my thanks to all the members of my committee, Dr. Cihan Tepedelenlioğlu and Dr. Daniel Bliss for their kind guidance and constructive feedback on my research.

I also thank the ECEE Graduate Program Chair, Dr. Joseph Palais, for giving me the opportunity to work as a teaching assistant. This has been one of the more rewarding experiences I have had during my time at ASU.

I am indebted to my lab members at Arizona State University and my friends. Your continuous assistance and friendship are very valuable to my life. I would specially like to thank Jinjin Li, Tong Zhu, Aditee Shrotre, Milind Gide and Srenivas Varadarajan for helping me overcome setbacks and stay sane through these difficult years. I also thank my dearest friends, Lin Bo, Dangdang Shao, Peng Wang, Ming Zhou, Qing Zhang for listening, companying and sharing.

Especially, I am grateful to my family for their unconditional love and care, especially to my parents. Without their innumerable sacrifices and support towards my life in general and my Ph.D. in particular, I would not have made it this far. You are always the sources of laughter, joy, and support.

Most importantly, none of this would have been achieved without my best friend, soul-mate, and husband. I truly appreciate Lifeng Miao for sticking by my side, understanding and encouraging me both academically and personally. You have made my life a dream a come true.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Detecting visual changes in an image or between images has become a vital and fundamental subject in computer vision. The typical tasks of computer vision include but are not limited to: 1) recognition, where the objective is to determine whether or not the image data contains some specific object, feature, scene or activity; 2) motion analysis, where an image sequence is processed to track an object or to detect the occurrence of an event/activity or estimate the velocity at various locations in the image; 3) 3D scene reconstruction, where the aim is to compute a 3D model of the scene from a number of images of that scene. Nevertheless, most of these tasks require detecting and understanding the variations in an image or between images. For example, in an image, different objects are delineated by strong boundaries, which correspond to significant changes in intensities, colors, texture, or other features. Thus, successfully detecting these changes could greatly help in detecting and recognizing objects. Or, for an image sequence, each image contains different content and it is important to detect the changes between each image in order to perform further analysis, such as motion estimation or event detection. In summary, change detection plays a very significant role in computer vision and image understanding.

Strong boundaries within an image are referred to as edges in the image processing and computer vision fields. These edges can occur along different spatial features (luminance, texture or color). Methods used to detect these strong boundaries are referred to as edge detection methods. On the other hand, variations between images are referred to as inter-view changes if the images correspond to views taken at the same time, or temporal changes if the views are taken at different times. In the latter case, methods that detect these changes are referred to as temporal change detection methods or change detection methods for short.

Edge detection finds a wide variety of applications including image segmentation, image enhancement, image fusion, compression, object detection and recognition, medical imaging and 3D reconstruction [1–5]. Edge-based image segmentation algorithms use edge cues to accurately determine region boundaries. For example, segmentation methods such as geometric active contours, gradient vector flow, and snakes use edge information to dictate their curve evolution [6–8].

The correct indication of boundaries between overlapping objects allows for accurate object identification and precise motion analysis for several machine vision applications [9, 10]. This initial procedure can often lead to further calculations such as area, perimeters, and shape classification of scene elements once they have been isolated from the image background [11]. As the human eye is very sensitive to edges, image enhancement algorithms, such as the Edge Detection-based Image Enhancement (EDIE) algorithm [12, 13] process the edge information of images differently than more homogenous information regions upon obtaining the raw output from an edge detector. Edge detection has arguably been used most extensively for object detection and recognition, as the features extracted by an edge detector can be used to form feature vectors for computer-aided classification systems [14–16]. Therefore, edge detection is a field of constant research in image processing and computer vision, and has received a lot of attention. A lot of research was conducted to design more robust edge detectors with improved performance in terms of accuracy, speed, and resource usage (such as memory and power consumption).

Detecting regions of change in images of a scene taken at different times is of widespread interest due to a large number of applications in diverse disciplines. Important applications of change detection include video surveillance [17–19], remote sensing [20–22], medical diagnosis and treatment [23–25], civil infrastructure [26, 27], underwater sensing [28–30], and driver assistance systems [31, 32]. Recently, with the availability of high-resolution remote sensing images, a lot of research focused on designing good change detection algorithms for remote sensing images because of the great benefit these bring in monitoring the Earth's surface. Usually, changes occurring after a natural or anthropoid disaster are abrupt and seldom predictable. As a result, the knowledge of the dynamics of either natural resources or man-made structures is a valuable source of information in decision making to fight the disaster. In this context, satellite and air-borne remote sensing sensors have proved particulary useful in addressing change-detection applications that are related to environmental monitoring, agricultural surveys, urban studies and flooding monitoring [33]. In recent years, especially after the availability of high-resolution synthetic aperture radar (SAR) images acquired by high-resolution space-borne systems, such as the European Space Agency (ESA) first Earth Remote Sensing Satellite (ERS-1), TerraSAR-X and TanDEM-X, multi-temporal change detection using high-resolution SAR images has received a lot of interest [34]. This is also due to the fact that the SAR imagery system can be operated under all-weather conditions. However, this new generation of SAR sensors enabled a significant increase in data volume and date rates. As a consequence, the large number of acquired images makes manual analysis very time consuming and impractical, and thus this increased the need for reliable automatic change detection algorithms for SAR images.

1.1 Problem Statement

Edge detection, as the one of the key stages of image processing and computer vision, has received a lot of attention. A wide range of algorithms were proposed and implemented to tackle the problems posed by edge detection. They include the Roberts [35], Prewitt [36], Sobel [37], Frei-Chen [38], Canny [39,40], and Laplacian of Gaussian (LoG)/Marr-Hildreth methods [41] [42], to name a few. Also, there have been few attempts to map some of the existing edge detection algorithms onto silicon [43–52]. Among all existing edge detectors, the Canny edge detector has best edge detection performance in terms of signal-to-noise ratio, edge localization and only one response to a single edge and as a result, it is generally considered to be the edge detection standard [50]. Its superior performance is due to the fact that the Canny algorithm performs hysteresis thresholding which requires computing high and low thresholds based on the entire image statistics. Unfortunately, this feature makes the Canny edge detection algorithm not only more computationally complex as compared to other aforementioned edge detection algorithms, but also necessitates additional pre-processing computations to be done on the entire image. This renders the Canny edge algorithm not practical especially for large-size images and when the memory resources are constrained. Also, a lot of popular computer vision and image processing algorithms are block-based in nature and the original Canny algorithm cannot be easily combined with block-based algorithms as it necessitates computing thresholds based on the entire image. Furthemore, a direct implementation of the Canny algorithm has high latency, which is not suitable for real-time applications. This problem is discussed and tackled in detail in Chapter 3.

Change information of the earth's surface is becoming more and more important in monitoring the local, regional and global resources and environment [53]. Especially, due to the all-weather operating ability and the availability of high-resolution SAR imagery, the large collection of SAR imagery makes it possible to analyze spatio-temporal patterns of environmental elements and the impact of human activities on the environment. Although plenty of successful application cases have been reported for SAR image change detection, there are enormous challenges in analyzing SAR imagery to derive useful and timely change information in relation to the earth's environment and human activities. Firstly, it has been generally agreed that change detection is a complicated and challenging process [53]. Since the SAR images are taken at different times, radiometric conditions can be quite different and thus can affect the accuracy of change detection algorithms. Radiometric conditions can be influenced by many factors such as different imaging seasons or dates, different solar altitudes, different acquisition angles, different meteorologic conditions and different cover areas of cloud, rain or snow etc. No existing approach is optimal and applicable to all cases. Secondly, one critical issue in multi-temporal SAR images is the detection of changes occurring after a natural or anthropic disaster. Since the changes produced by these events are abrupt and seldom predictable, they are often difficult to model, even for the same kind of change. For example, an earthquake can have different features depending on when or where it happens. Also, the changes of interest are all mixed up with normal changes if the time gap between the two acquisitions is too long [54]. Finally, at present, for most change detection algorithms, the degree of automation is low while the computational complexity is high, which prevents real-time applications. In Chapter 4 to Chapter 6 of this report, some of these challenges are discussed and novel solutions are proposed for improved multi-temporal SAR change detection.

1.2 Contributions

In Chapter 3 , we present a novel distributed Canny edge detection algorithm which has the ability to compute edges of multiple blocks at the same time. To support this, an adaptive threshold selection method that predicts the high and low thresholds of the entire image while only processing the pixels of an individual block, is proposed. This results in three benefits: 1) a significant reduction in the latency; 2) better edge detection performance; 3) the possibility of pipelining the Canny edge detector with other block-based image codecs; 4) significantly reduced memory resource requirements for large-size images. In addition, a low complexity non-uniform quantized histogram calculation method is proposed to compute the block hysteresis thresholds. The proposed algorithm is scalable and has very high detection performance. We show that our algorithm can detect all psycho-visually important edges in the image for various block sizes. We also show that the proposed algorithm results in superior edge detection performance as compared to the original Canny detector, especially in the presence of noise. Finally, the algorithm is mapped onto a Xilinx Virtex-5 FPGA platform and tested using ModelSim. The FPGA implementation takes only 0.721ms (including the SRAM read/write time and the computation time) to detect edges of 512×512 images in the USC SIPI database when clocked at 100 MHz.The proposed distributed Canny edge detector enables parallel implementations supporting fast real-time edge detection of images and videos including those with full-HD content.

After a review of the exiting state-of-art multi-temporal change detection methods in Chapter 4, novel statistical similarity measures for automatic change detection using multi-temporal SAR images are presented in Chapter 5 and 6. A similarity measure based on an analysis of the local distribution of SAR images using a Gaussian Mixture Model and the KL distance is presented in Chapter 5. The locally adaptive Gaussian Mixture Model is more suitable and flexible than the single parametric mathematical distribution models to approximate the distribution of the actual SAR image, which shows a varied scene presenting several distinct land-cover typologies. Also, novel spatial-domain and wavelet-domain change detection methods are presented in Chapter 5 based on the proposed similarity measure. Compared to existing detectors with higher order statistics both in the spatial and wavelet domains, the proposed change detection methods are shown to exhibit a better change detection performance in terms of the quality of the similarity maps, which are assessed using the receiver operating characteristic (ROC) curves, and in terms of the total error rates of the final change detection maps.

In Chapter 6, we propose a new image comparison method that is inspired by the recent success of the divisive normalization transform (DNT) as a statistically and perceptually motivated image representation. This DNT-based change detection algorithm exploits desirable properties of the DNT, namely that the DNT was empirically shown to produce approximately Gaussian marginal distributions and to reduce the statistical dependencies of the original linear wavelet representation. This results in a one-parameter Gaussian statistical model, which allows efficient change detection. Tests on real data show that our proposed DNT-based change detector exhibits competitive detection performance while achieving lower computational complexity as compared to previously suggested methods.

1.3 Organization

This thesis is organized as follows. Chapter 2 presents background material about edge detection. This chapter covers basic concepts related to edge, edge detection and existing edge detection algorithms. Chapter 3 presents a novel distributed Canny edge detection algorithm and corresponding performance analysis for clean and noisy images. Chapter 4 gives an overview of the general change detection system and existing methods for multi-temporal SAR images. Chapter 5 presents a new GMM-based similarity measure for automatic change detection using multi-temporal SAR images. Novel spatial-domain and wavelet-domain change detection methods are also presented in Chapter 5 based on the proposed similarity measure. In Chapter 6, we propose a new reduced-complexity similarity measure for automatic change detection based on the divisive normalization transform (DNT). Finally, Chapter 7 summarizes the contributions of this work and discusses possible future research directions.

Chapter 2

EDGE DETECTION BASICS

2.1 Edge and Edge detection definition

Edges are a class of singularities which commonly appear in image processing [55] and locations in an image at which intensity values change abruptly [56]. The process of determining edge pixels within an image is known as edge detection [57]. In practical, an edge detector is usually defined as a mathematical operator of small spatial extent that responds in some way to edges, usually classifying every pixel in an image as either corresponding to an edge pixel or not [58]. This binary decision output of an edge detector is usually referred to as an edge map. Edge detection is an important step in several applications, especially when dealing with feature-based image processing and scene analysis systems [59]. This is because edges determine the structure of objects in images [11]. Edge detection is a useful, low-level form of image processing for obtaining a simplified image [60]. Consequently, edge detection has been used as a preprocessing for many computer vision and image processing tasks, such as image enhancement, image segmentation, image fusion, information hiding, compression, and object detection and recognition, etc. The qualify of the edge detection output often dictates the results of further processing [30]. Thus, accurately locating edge pixel location in an image is of importance and the development of accurate edge detection in both clean and noisy environments is a must. For many years, researchers have been dedicated in designing good detectors. In this chapter, we will discuss the classical edge detection algorithms.

2.2 Existing Edge Detection Methods

Since edges are defined as points in an image at which intensity values vary sharply, most common edge detectors are derived from computing the directional derivative of the image. There are two approaches. The one involves calculating the first partial derivative, also called gradient, in the horizontal and vertical directions of the image [61], including the Roberts, Prewitt, Sobel, Frei-Chen and Canny algorithms. Typically, these first derivative-based methods detect edges by locating the maxima of the magnitude of the gradient of the image. In essence, they attempt to match local image neighborhoods with predetermined edge patterns to determine edge points [62]. The other approach involves calculating the second derivative information ,also called Laplacian, of the image [61], including the Laplacian of Gaussian/Marr-Hildreth algorithm. These second derivative-based methods detect edges by locating zero crossings of the Laplacian of the image. All the aforementioned edge detectors are largely based on discrete convolution operations using specifically designed directional kernels, noted as F_x and F_y, to traverse the image and thus detect edges [61].

2.2.1 First Derivative-Based Edge Detector

The first derivative-based edge detectors are commonly derived form computing the gradient of the image. The gradient G of an image I is defined as a directional change in the intensity or color in an image and is given by the formula:

$$G = \nabla I = \frac{\partial I}{\partial x}\widehat{x} + \frac{\partial I}{\partial y}\widehat{y} \tag{2.1}$$

$$G_x = \frac{\partial I}{\partial x}\widehat{x} = F_x * I \tag{2.2}$$

$$G_y = \frac{\partial I}{\partial y}\widehat{y} = F_y * I \tag{2.3}$$

where G_x and G_y are the horizontal and vertical gradient, respectively; and F_x and F_y are the first directional derivative kernels. The gradient magnitude image $|G|$, which corresponds to the likelihood of edge information, can be calculated by $|G| = \sqrt{(G_x)^2 + (G_y)^2}$ Also, the gradient direction can be computed by $\theta_G = tan^{(-1)}(\frac{G_y}{G_x})$. Typically, the kernels used in gradient-based edge detection algorithms do not measure the true gradient but rather an approximation of the gradient, as they contain some embedded smoothing to combat noise. The kernels are, however, closely related to discrete differentiators, and in many cases can be separated into two 1-D filters [61].

- Roberts Edge Detector

The directional kernels used by the Roberts edge detector [41] [35] are given by:

$$F_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, F_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{2.4}$$

From the equation, it is clear that these kernels are simple 2D discrete differentiators and oriented diagonally. More specifically, the F_x kernel oriented at +45° while the F_y kernel in this case detects edges oriented at −45°. For the Robert edge detector, there is no embedded smoothing stage. This feature makes the Robert detector be capable to locate the accurate edge position but very sensitive to noise. Beside, it is worthwhile to notice that the kernels have even length dimensions; and this yields offset, where filter output is not being centered on a central pixel. The output binary edge map is produced by calculating the gradient using the Roberts edge detection kernels and thresholding its magnitude by some predetermined constant [41].

- Prewitt Edge Detector

The directional kernels used by the Prewitt edge detector [36] are as follows:

$$F_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}, F_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{2.5}$$

As show in the equation, the Prewitt edge detection kernels can be separated into two 1D filters, an arithmetic mean filter and a central difference filter. The F_x kernel averages along columns to smooth noise and differentiates along rows, while the F_y kernel averages along rows and differentiates along columns. As a result, F_x detects vertical edges while F_y detects horizontal edges. Similarly withe the Roberts detector, the final binary edge map is obtained by thresholding the gradient magnitude, which are calculated by using the Prewitt edge detection kernels.

- Sobel Edge Detector

Similarly, the directional kernels used by the Sobel edge detector [37] are given by:

$$F_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}, F_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{2.6}$$

Similarly with the Prewitt edge detector, the F_x and F_x kernels of Soble edge detector detect vertical and horizontal edges, respectively, and also can be separated into an mean and central difference filter pair. The only difference between the Prewitt and Sobel edge detection kernels is the choice of the 1D smoothing filter. The smoothing filter used by Sobel is a weighted mean filter which resembles a Gaussian function, as its filter coefficients are the binomial coefficients for n = 3. Again, the final binary edge map is obtained by thresholding the gradient magnitude.

- Canny Edge Detector

Compared to the edge aforementioned edge detectors (Roberts, Prewitt and Sobel), Canny edge detection algorithm have more pre-processing and post-processing steps; and thus its performance is superior to others. First, these edge detectors mentioned before are sensitive to noise since the image is not smoothed first. However, the Canny edge detector first smoothes the input image with a Gaussian filter and then computes the gradient of the smoothed image. Secondly, the non-maximum suppression step is employed by the Canny algorithm in order to remove pixels that are not considered to be part of an edge. Hence, only thin lines (candidate edges) will remain. Finally, instead of performing simply manual thresholding, Canny edge detection algorithms performs hysteresis thresholding which requires computing high and low thresholds based on the entire image statistics. As a result, the Canny edge detector is optimal for isolated step edges with white Gaussian noise and generally considered to be the edge detection standard [61].

Canny developed an approach to derive an optimal edge detector to deal with step edges corrupted by a white Gaussian noise. It was based on three criteria that are related to the detection performance. These criteria can be summarized as follows [39]:

1. Good detection. Ideally, this is accomplished by choosing the filter A which maximizes the signal-to-noise ratio (SNR), given as

$$SNR(f(x)) = \frac{A}{\sigma_n} \cdot \frac{|\int_{-\infty}^{0} f(x)\,dx|}{\int_{-\infty}^{\infty} f(x)^2\,dx} \tag{2.7}$$

   where A is the amplitude of the step edge and σ_n^2 is the mean-squared noise variance.

2. Good localization. The points marked as edge points by the filter should be as close as possible to the center of the true edge. This is achieved by maximizing

$$Localization(f(x)) = \frac{A}{\sigma_n} \cdot \frac{|f'(0)|}{\int_{-\infty}^{\infty} f'^2(x)\,dx} \tag{2.8}$$

   where $f'(x)$ is the first derivative of the filter $f(x)$, and $f'(0)$ is the first derivative at location $x = 0$.

3. Only one response to a single edge. This is achieved by maximizing the mean distance x_max between two adjacent peaks in the noise response, given by

$$x_{max}(f(x)) = 2\pi\sqrt{\frac{\int_{-\infty}^{\infty} f'^2(x)\,dx}{\int_{-\infty}^{\infty} f''^2(x)\,dx}} \tag{2.9}$$

   where $f''(x)$ is the second derivative of the filter $f(x)$.

Unfortunately, there is a tradeoff between the detection performance (SNR) and localization; this tradeoff can be achieved through the spatial scaling of $f(x)$. So, the optimal filter is determined by maximizing the product of the detection term $SNR(f(x))$ and localization term $localization(f(x))$ under the constraint of the single response term $x_{max}(f(x))$. Numerically, the optimal detector for step edges in the presence of white Gaussian noise can be approximated by the first derivative of a Gaussian function as follows:

$$f(x) = -\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}} \tag{2.10}$$

The original Canny algorithm [39] consists of the following steps:

1. Calculating the horizontal gradient G_x and vertical gradient G_y at each pixel location by convolving with gradient masks.

2. Computing the gradient magnitude $|G|$ and direction θ_G at each pixel location.

3. Applying Non-Maximal Suppression (NMS) to thin edges. This step involves computing the gradient direction at each pixel. If the pixel's gradient direction is one of 8 possible main directions (0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°), the gradient magnitude of this pixel is compared with two of its immediate neighbors along the gradient direction and the gradient magnitude is set to zero if it does not correspond to a local maximum. For the gradient directions that do not coincide with one of the 8 possible main directions, an interpolation is done to compute the neighboring gradients.

4. Computing high and low thresholds based on the histogram of the gradient magnitude for the entire image. The high threshold is computed such that a percentage P_1 of the total pixels in the image would be classified as strong edges. In other words, the high threshold

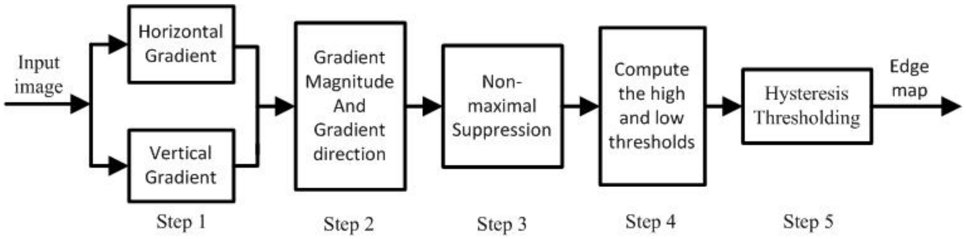


Figure 2.1: Block Diagram of the Canny Edge Detection Algorithm.

corresponds to the point at which the value of the gradient magnitude cumulative distribution function (CDF) equals to 1-P_1. The low threshold is computed as a percentage P_2 of the high threshold. The values of P_1 and P_2 are typically set as 20% and 40%, respectively [39, 61].

5. Performing hysteresis thresholding to determine the edge map. If the gradient magnitude of a pixel is greater than the high threshold, this pixel is considered as a strong edge. If the gradient magnitude of a pixel is between the low threshold and high threshold, the pixel is labeled as a weak edge. Strong edges are interpreted as ”certain edges”, and can be immediately included in the final edge images. Weak edges are included if and only if they are connected to strong edges.

A block diagram of the Canny edge detection algorithm [39] is shown in Fig. 2.1. In this original Canny edge detection algorithm [39], the gradient calculation (Step 1) is performed by using Finite-Inpulse Response (FIR) gradient masks designed to approximate the following 2D sampled versions of the partial derivatives of a Gaussian function:

$$F_x(x,y) = -\frac{x}{\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} = (-xe^{-\frac{x^2}{\sigma^2}})(\frac{1}{\sigma^2}e^{-\frac{y^2}{\sigma^2}}) \tag{2.11}$$

$$F_y(x,y) = -\frac{y}{\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} = (-ye^{-\frac{y^2}{\sigma^2}})(\frac{1}{\sigma^2}e^{-\frac{x^2}{\sigma^2}}) \tag{2.12}$$

where σ is the standard deviation of the Gaussian function. The size of the gradient masks used by the Canny edge detector is usually implemented as a function of the chosen σ, with larger values of σ yielding larger masks. However, the best choice of σ is image-dependent and can be selected by the user based on knowledge of the present noise characteristics or the size of desired objects in the image [40]. The parameter σ can also be set by a separate application that estimates the noise and/or scale of objects in the image. The effect of the gradient mask size is illustrated in Section 3.3.1.

2.2.2 Second Derivative-Based Edge Detector

As described before, the second derivative information can also be used for edge detection since maxima in the first derivative of a signal is equivalent to zero crossing in the second derivative of the signal. For the 2D case, the Laplacian operator is defined as

$$L = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{2.13}$$

There are many ways to generate a discrete Laplacian kernel and the general implementation of the $n \times n$ Laplacian is given as

$$Laplacian(i,j) = \begin{cases} n^2 - 1, & \text{i=j=ceil(n/2)} \\ -1, & \text{otherwise} \end{cases} \tag{2.14}$$

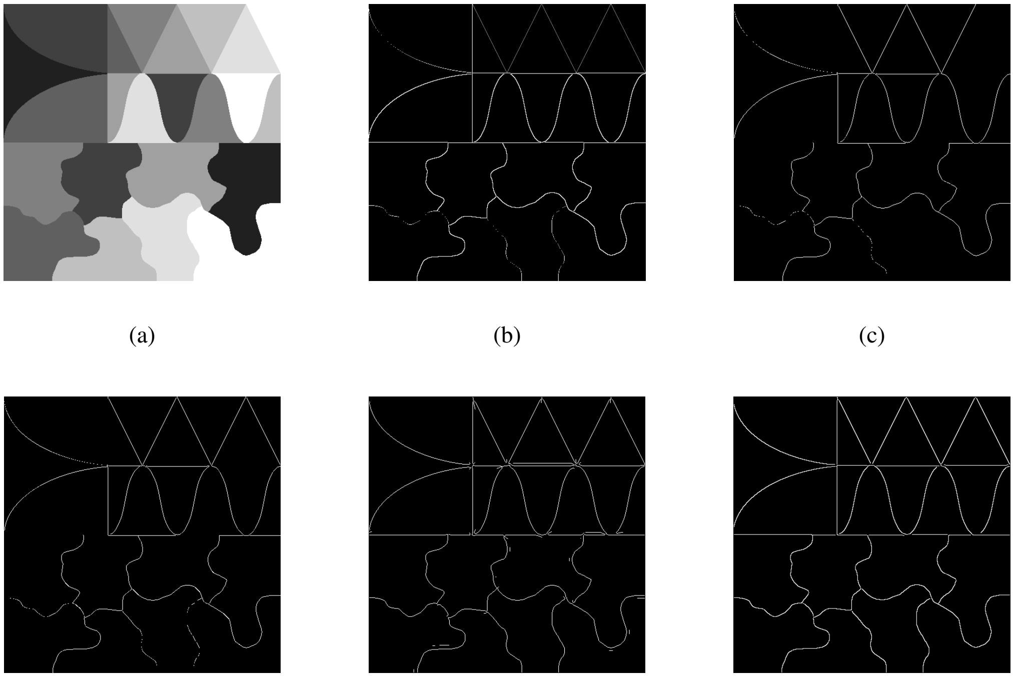
Compared to the first derivative operator, the second derivative of a function is even more sensitive to noise [63]. Therefore, the typical Laplacian-based edge detection algorithms measure the Laplacian of a smoothed version of the image rather than the original image to increase the robustness to noise. The Laplacian of Gaussian (LoG)/Marr-Hilldreth [41,42] edge detector algorithm is the one of the most famous Laplacian-based edge detection algorithms. It consists of the following steps:

1. Convolute the image with a 2-D Gaussian function in order to smooth the image.

2. Laplacian filtering step. Calculate the Laplacian of the smoothed image. The LoG kernel can also be calculated directly for a Gaussian function with standard deviation σ by sampling the function

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{\frac{x^2+y^2}{2\sigma^2}} \tag{2.15}$$

3. The output binary edge map is produced by determining the location of zero crossing of the Laplacian of the smoothed image, and thresholding the difference between pixels on either side of the zero crossings by some manual set constant.



(a) (b) (c)

(d) (e) (f)

Figure 2.2: (a) Original Image, Obtained Edge Maps by Using (b) Roberts, (c) Prewitt, (d)Sobel, (e) LoG, (f) Canny Edges Detection Algorithms.

2.3 Experimental Results

Fig. 2.2 shows an example of results using the discussed edge detectors. This experiment shows that Roberts, Sobel and Prewitt detector outputs provide many disconnected, false and disconnected edge responses. They also fail to detect some coarser scale edges because of their relatively smaller kernels. The LoG edge detector also can't detect many important edges and produces many spurious edges. The Canny edge detector produces continuous, thin and correct edges. We can conclude that Canny edge detector produces the best output for this image. However, the Canny edge detector output still suffers from localization errors due to standard deviation and some edge losses due to smoothing.

Chapter 3

DISTRIBUTED CANNY EDGE DETECTION ALGORITHM

3.1 Introduction

Edge detection is the most common preprocessing step in many image processing algorithms such as image enhancement, image segmentation, tracking and image/video coding. Among the existing edge detection algorithms, the Canny edge detector has remained a standard for many years and has best performance [39]. Its superior performance is due to the fact that the Canny algorithm performs hysteresis thresholding which requires computing high and low thresholds based on the entire image statistics. Unfortunately, this feature makes the Canny edge detection algorithm not only more computationally complex as compared to other edge detection algorithms, such as the Roberts, Prewitt and Sobel algorithms, but also necessitates additional pre-processing computations to be done on the entire image. As a result, a direct implementation of the Canny algorithm has high latency and cannot be employed in real-time applications.

Many implementations of the Canny algorithm have been proposed on a wide list of hardware platforms. There is a set of work [43,45,46] on Deriche filters that have been derived using Canny criteria and implemented on ASIC-based platforms. The Canny-Deriche filter [45] is a network with four transputers that detect edges in a 256×256 image in 6s, far from the requirement for real-time applications. Although the design in [43] improved the Canny-Deriche filter implementation of [45] and was able to process 25 frames/s at 33 MHz, the used off-chip SRAM memories consist of Last-In First-Out (LIFO) stacks, which increased the area overhead compared to [45]. Demigny proposed a new organization of the Canny-Deriche filter in [46], which reduces the memory size and the computation cost by a factor of two. However, the number of clock cycles per pixel of the implementation [46] varies with the size of the processed image, resulting in variable clock-cycles/pixel from one image size to another with increasing processing time as the image size increases.

There is another set of work [47–49] on mapping the Canny edge detection algorithm onto FPGA-based platforms. The two FPGA implementations in [47] and [48] translate the software design directly into VHDL or Verilog using system-level hardware design tools, which results in a decreased timing performance as shown later in Section 3.4.3 of this work. The parallel implementation in [49] operates on only 4 pixels in parallel, resulting in an increase in the number of memory accesses and in the processing time as compared to the proposed algorithm. Furthermore, in order to reduce the computational complexity, all of these implementations compute the high and low thresholds off-line and use the same fixed threshold values for all the images, which result in a decreased edge detection performance. In [64], a self-adapt threshold Canny algorithm is proposed for a mobile robot system where a low and high threshold values are computed for each input image and an implementation for an Altera Cyclone FPGA is presented.

Recently, the General Purpose Graphic Processing Unit (GPGPU) has emerged as a powerful and accessible parallel computing platform for image processing applications [65, 66]. Studies of GPGPU accelerated Canny edge detection have been presented [50–52]. All of these implementations are frame-based and do not have good edge detection performance since they use the same fixed pair of high and low threshold values for all images. Furthermore, as shown later in the paper, their timing performance is inferior compared to the proposed algorithm in spite of being operated at a very high clock frequency.

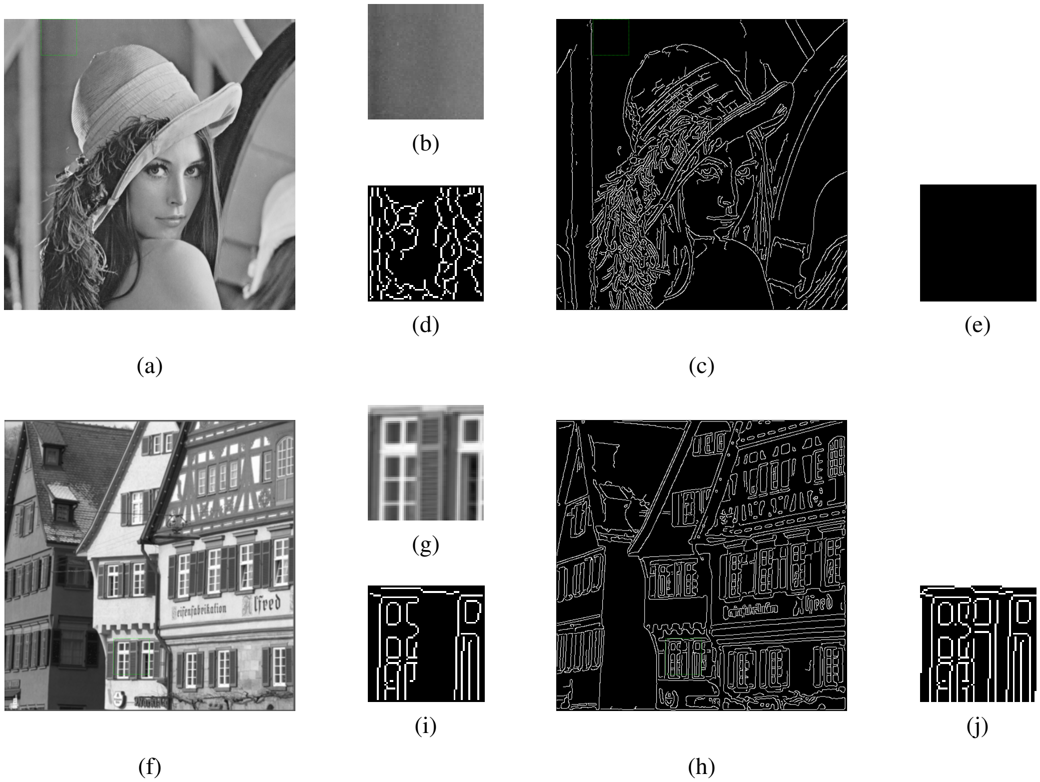
In the original Canny method, the computation of the high and low threshold values depends on the statistics of the whole input image. However, most of the above existing implementations (e.g., [47–49], [50–52]) use the same fixed pair of high and low threshold values for all input images. This results in a decreased edge detection performance as discussed later in this report. The non-parallel implementation [64] computes the low and high threshold values for each input image. This results in increased latency as compared to the existing implementations (e.g., [47–49, 64], [50–52]). Furthermore, the non-parallel implementations ([47–49]) result in a decreased throughput as compared to the parallel implementations ([49], [50–52]). The issue of increased latency and decreased throughput is becoming more significant with the increasing demand for large-size high-spatial resolution visual content (e.g., High-Definition and Ultra High-Definition).

Our focus is on reducing the latency and increasing the throughput of the Canny edge detection algorithm so that it can be used in real-time processing applications. As a first step, the image can be partitioned into blocks and the Canny algorithm can be applied to each of the blocks in parallel. Unfortunately, directly applying the original Canny at a block-level would fail since it leads to excessive edges in smooth regions and loss of significant edges in high-detailed regions. In this report, we propose an adaptive threshold selection algorithm which computes the high and low threshold for each block based on the type of block and the local distribution of pixel gradients in the block. Each block can be processed simultaneously, thus reducing the latency significantly. Furthermore, this allows the block-based Canny edge detector to be pipelined very easily with existing block-based codecs, thereby improving the timing performance of image/video processing systems. Most importantly, conducted conformance evaluations and subjective tests show that, compared with the frame-based Canny edge detector, the proposed algorithm yields better edge detection results for both clean and noisy images. The block-based Canny edge detection algorithm is mapped onto an FPGA-based hardware architecture. The architecture is flexible enough to handle different image sizes, block sizes and gradient mask sizes. It consists of 32 computing engines configured into 8 groups with 4 engines per group. All 32 computing engines work in parallel lending to a 32-fold decrease in running time without any change in performance when compared with the frame-based algorithm. The architecture has been synthesized on the Xilinx Virtex-5 FPGA. It occupies 64% of the total number of slices and 87% of the local memory, and takes $0.721ms$ (including the SRAM read/write time and the computation time) to detect edges of 512×512 images in the USC SIPI database when clocked at $100MHz$.

The rest of this chapter is organized as follows. Section 3.2 presents the proposed distributed Canny edge detection algorithm which includes the adaptive threshold selection algorithm and a non-uniform quantization method to compute the gradient magnitude histogram. The effects of the gradient mask size and the block size on the performance of the proposed distributed Canny edge detection scheme are discussed and illustrated in Section 3.3. In Addison, quantitative conformance as well as subjective testing results are presented in Section 3.4 in order to illustrate the edge detection performance of the proposed distributed Canny algorithm as compared to the original Canny algorithm for clean as well as noisy images. The FPGA synthesis and performance evaluation are presented in Section 3.4.3.

3.2 Distributed Canny Edge Detection Algorithm

As discussed in Section 2.2.1, the classical Canny edge detection algorithm sets the high and low thresholds based on the distribution of the gradients at all the pixel locations of an image. Thus, directly applying the original Canny algorithm to a block of the image would fail to detect


(b)

(d) (e)

(a) (c)

(g)

(i) (j)

(f) (h)

Figure 3.1: (a), (f) Original 512 × 512 Lena And Houses Images; (b), (g) Zoomed-In Blocks of the Lena And Houses Images; (d), (i) Edge Maps of Block Images in (b) And (g) Obtained by Applying the Classical Canny Edge Detector to Blocks Separately; (c), (h) Edge Maps Using the Canny Edge Detector Applied to the Entire Lena Image And Houses Image; (e), (j) Zoomed-In Blocks of the Edge Maps (b) & (j).

the desired edges since the statistics of a block may differ significantly from the statistics of the entire natural image. For example, Fig. 3.1 shows two types of blocks, namely smooth block and edge block, extracted from the Lena image and Houses image, respectively. Despite the fact that the smooth block (Fig. 3.1 (b)) does not contain any significant edges, applying the original Canny algorithm to this block results in the detection of an excessive number of edges as shown in Fig. 3.1 (d). This is due to the fact that the Canny algorithm assumes that a percentage P_1 of the total pixels in the block are strong edges. On the other hand, for an edge block (Fig. 3.1 (g)), the majority of pixels are edges. Since P_1 is much smaller than the actual percentage of edge pixels in the edge block, it results in missing edges as shown in Fig. 3.1 (i). For comparison, Fig. 3.1 (c), (e), (h) & (j) show the performance of the Canny edge detector when applied to the entire image. In latter case, the Canny edge detector is able to achieve the expected edge detection performance in the smooth

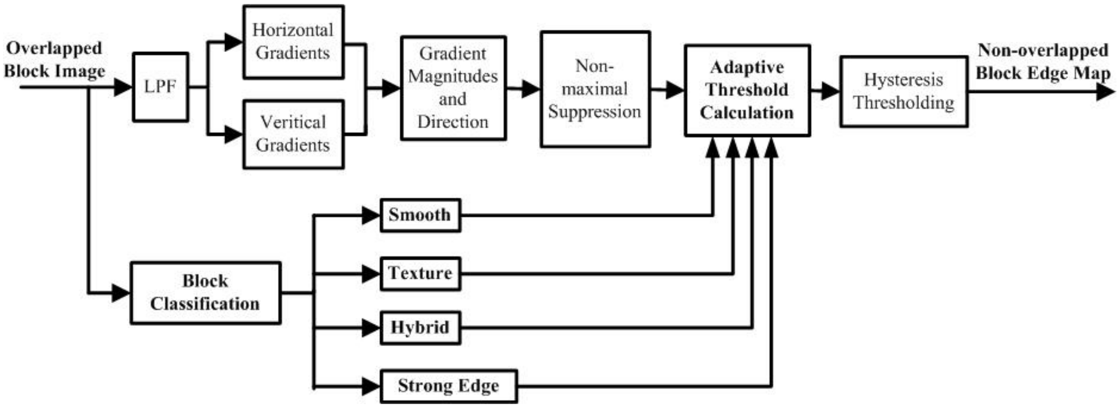


Figure 3.2: Proposed Distributed Canny Edge Detection Algorithm.

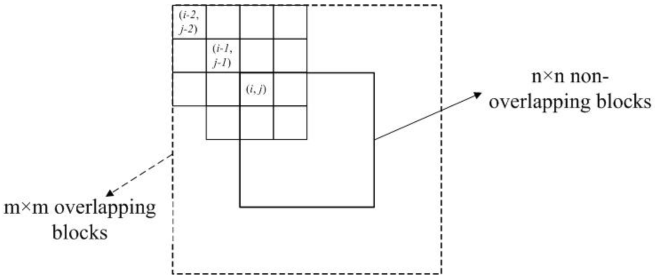


Figure 3.3: An Example of the Structure of An $m \times m$ Overlapping Block, Where $m = n + L + 1$ for An $L \times L (L = 3)$ Gradient Mask And When the Image Is Initially Divided into $n \times n$ Non-Overlapping Blocks.

regions (Fig. 3.1 (e)) and in the edge regions (Fig. 3.1 (j)).

In order to improve the performance of the edge detection at the block level and achieve the same performance as the original frame-based Canny edge detector when this latter one is applied to the entire image, a distributed Canny edge detection algorithm is proposed. A diagram of the proposed algorithm is shown in Fig. 3.2. In the proposed distributed version of the Canny algorithm, the input image is divided into $m \times m$ overlapping blocks and the blocks are processed independent of each other. For an $L \times L$ gradient mask, the $m \times m$ overlapping blocks are obtained by first dividing the input image into $n \times n$ non-overlapping blocks and then extending each block by $(L+1)/2$ pixels along the left, right, top, and bottom boundaries, respectively. This results in $m \times m$ overlapping blocks, with $m = n + L + 1$. The non-overlapping $n \times n$ blocks need to be extended in order to prevent edge artifacts and loss of edges at block boundaries while computing the gradi-

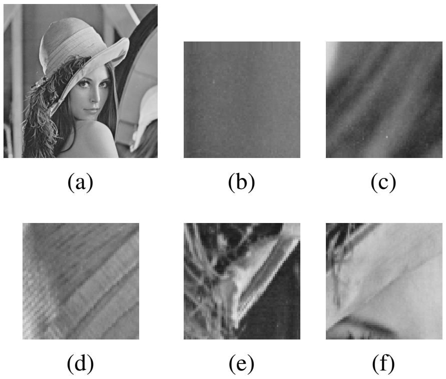


Figure 3.4: (a) Original 512 × 512 Lena Image; b) Uniform Block; (c) Uniform /Texture Block; (d) Texture Block; (e) Edge/Texture Block; (f) Medium Edge Block of the Lena Image Shown Blocks Are of Size 64 × 64.

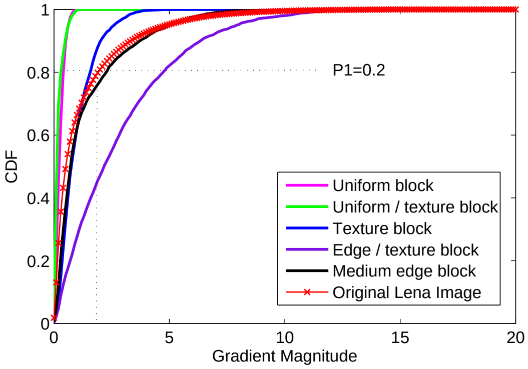


Figure 3.5: Normalized Gradient Magnitude CDFs for the 512 × 512 Lena Image And CDFs for the 64 × 64 Uniform Block, Uniform/Texture Block, Texture Block, Edge/Texture Block, Medium/Edge Block Shown in Fig. 3.4.

ents and due to the fact that the NMS operation at boundary pixels requires the gradient values of the neighboring pixels of the considered boundary pixels in a block. Fig. 3.3 shows an example of non-overlapping block and its extended overlapping block version in the case when a 3×3 gradient mask. In order to perform NMS for the border pixel (i, j), the gradient information of the adjacent pixels $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i+1, j-1)$ are needed. In order to compute the gradient of the adjacent pixels $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i+1, j-1)$ for the 3×3 gradient mask, the block has to be extended by 2 (where $(L-1)/2+1=2$) pixels on all sides in order to generate a block of size $(n+4)\times(n+4)$. Thus, m equals to $n+4$ for this example. Note that, for each block, only edges in the central $n\times n$ non-overlapping region are included in the final edge map, where $n=m-L-1$. Steps 1 to 3 and Step 5 of the distributed Canny algorithm are the same as in the original Canny algorithm except that these are now applied at the block level. Step 4, which is the hysteresis high and low thresholds calculation, is modified to enable parallel block-level processing without degrading the edge detection performance.

As discussed above, exploiting the local block statistics can significantly improve the edge detection performance for the distributed Canny edge detector. For this purpose, in order to learn the statistics of local blocks, we use a training database consisting of natural images. For each image in the database, we first divide the image into $n\times n$ non-overlapping blocks and classify the blocks into six types, uniform, uniform/texture, texture, edge/texture, medium edge, and strong edge block, by adopting the block classification method of [67]. This classification method utilized the local variance of each pixel using a 3×3 window that is centered around the considered pixel in order to label it as of type edge, texture, or uniform pixel. Then, each block is classified based on the total number of edge, texture, and uniform pixels in the considered block. Fig. 3.4 shows an example where the 512×512 Lena image is divided into blocks, and each block is classified according to this classification technique. Fig. 3.4 (b), (c), (d), (e) and (f) provide, respectively, examples of uniform, uniform/texture, texture, edge/texture, medium edge block for the 512×512 Lena image with a block size of 64×64. Note that there are no strong edge blocks for the Lena image.

In order not to degrade the performance of the original frame-based Canny algorithm when it is applied to a block, the high and low thresholds when computed at the block level should match thresholds that would have been obtained for the entire image. The gradient magnitude CDFs of each block in Fig. 3.4 are shown in Fig. 3.5 , along with the gradient-magnitude CDF of the entire Lena image. According to the CDF of the entire Lena image, the high threshold should be selected as 1.8 corresponding to a P_1 value (percentage of pixels that should be classified as strong edges) of 0.2. However, if P_1 is still set to 0.2 for the medium edge block, the local high threshold for that block would be about 5.2, which is significantly different from the high threshold, 1.8, that is obtained for the entire image. Such a setting will result in a loss of significant edges in the considered strong edge block. On the other hand, the local high thresholds for the uniform and uniform/texture block would be about 0.7, if P_1 is set to 0.2. Such a setting will lead to excessive edges in the uniform and uniform/texture blocks. From this analysis, it can be seen that, in order to keep a similar threshold for all block types, P_1 should be selected differently for different types of blocks by adapting its value to the local block content.

In order to determine, for each block type, the appropriate percentage value P_1 that would result in high and low threshold values similar to the ones that are obtained for the entire image, a training set of 200 images is formed from the Berkeley Segmentation Image Database [68]. For each image in the training database, the high threshold of the entire image is first calculated. Then, the image is divided into blocks and the blocks are classified into six block types as discussed previously. Then,

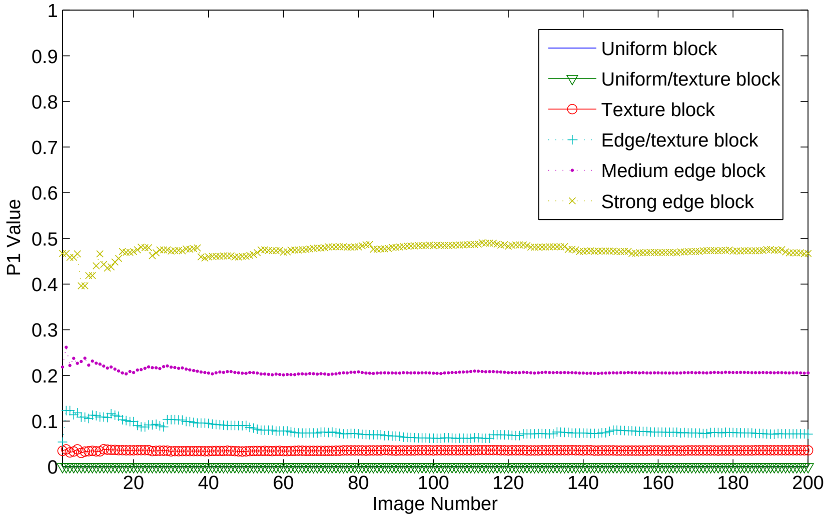


Figure 3.6: P_1 Values for Each Block Type.

Table 3.1: Standard Deviations of P_1 Values for Each Block Type When Block Size Is 64×64.

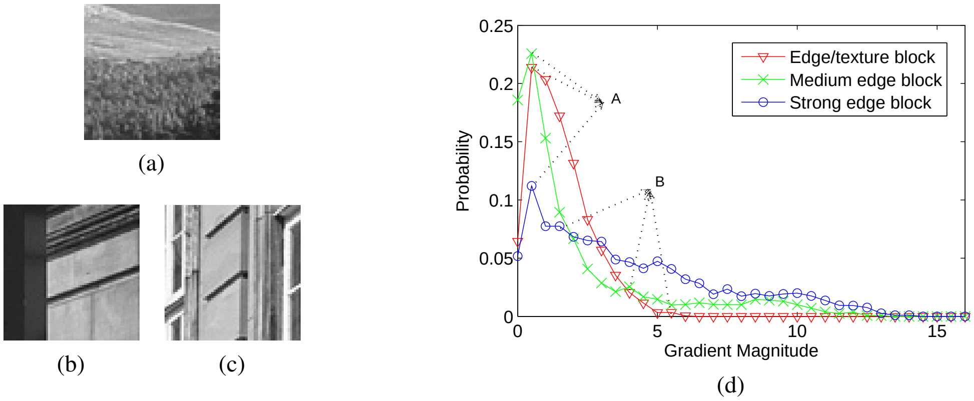
Block type	Uniform	Uniform/ Texture	Texture	Edge/ Texture	Medium Edge	Strong Edge
Variance of P_1 value	0	0	$9.37E^{-4}$	0.0138	0.0071	0.0132

for each block type, the gradient magnitude CDF is computed and the corresponding CDF used to compute the P_1 value such that the local high threshold of the blocks in this class is the same as the one for the entire image. Fig. 3.6 shows the P_1 values that are obtained for different block types, each of size 64×64, for 200 512×512 images with varied content. It illustrates that the P_1 values of each block type are highly different from each other, except for the uniform block and uniform/texture block types. Also, for a given block type, the P_1 values across all 200 images are quite similar for different block sizes. The final P_1 value for a considered block type is computed as the average value of its corresponding set over all images and over all block sizes. For illustration, the standard deviations of the obtained P_1 values for each block type are shown in Table 3.1 for 64×64 blocks.

To evaluate the robustness of the obtained P_1 values with respect to the block size, the 512×512 images are divided into fixed-size blocks, with the block size varying from 8×8 to 256×256. Table

Table 3.2: P_1 Values for Each Block Type with Different Block Sizes.

Block Size	Block Type					
	Uniform	Uniform/ Texture	Texture	Edge/ Texture	Medium Edge	Strong Edge
8×8	0	0	0.0312	0.1022	0.2183	0.4820
16×16	0	0	0.0307	0.1016	0.2616	0.4830
32×32	0	0	0.0305	0.1117	0.2079	0.4852
64×64	0	0	0.0318	0.1060	0.2218	0.4670
128×128	0	0	0.0302	0.0933	0.2375	0.4842
256×256	0	0	0.0299	0.0911	0.2304	0.4893



(a)

(b) (c)

(d)

Figure 3.7: (a)-(c) Different Types of 64×64 Blocks ((a) Edge/Texture, (b) Medium Edge, (c) Strong Edge) And (d) Corresponding Gradient Magnitude Histograms.

3.2 shows the P_1 values that are obtained for each block type and for each block size. It should be noted that the P_1 values for uniform and uniform/texture blocks are equal to 0 for all block sizes, which indicates that the uniform and uniform/texture blocks can be combined into one block type, which we refer to as smooth block type. Also, this implies that there are no pixels that should be classified as edges in a smooth block. Therefore, there is no need to perform edge detection on smooth blocks, and this results in reduced computations, a feature that is exploited in the FPGA implementation.

In order to compute the high and low hysteresis thresholds, a finely quantized gradient magnitude histogram is usually needed. Here, we employ the non-uniform quantizer, which has been proposed by us in [69], to obtain the gradient magnitude histogram for each block such that the high threshold can be precisely calculated. As in [64, 69], for the edge/texture, medium edge and

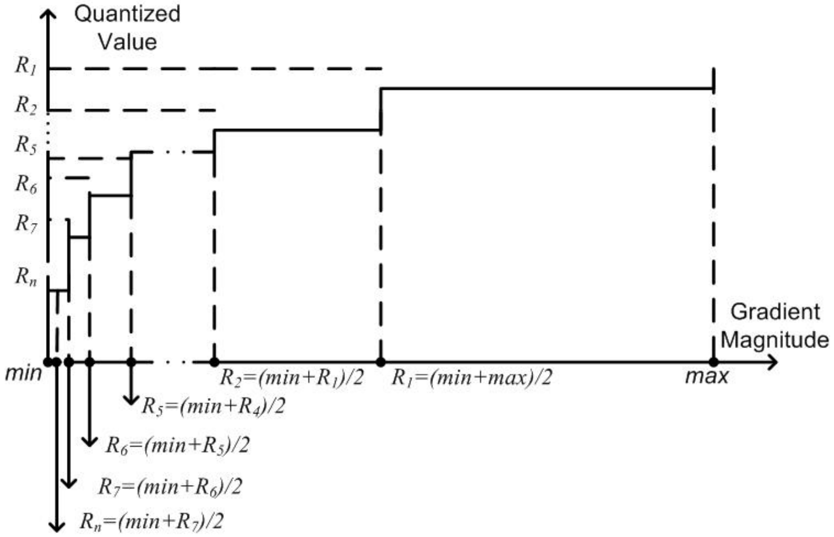
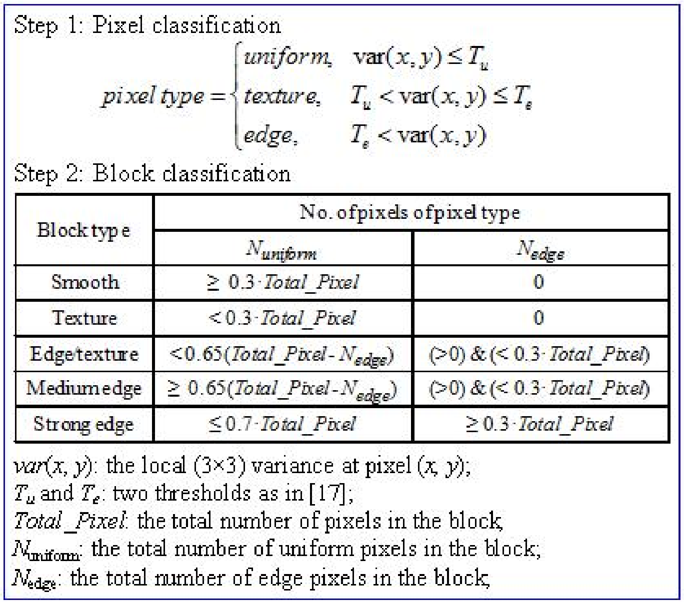


Figure 3.8: Reconstruction Values And Quantization Levels.

strong edge block, it was observed that the largest peak in the gradient magnitude histograms of the Gaussian-smoothed images occurs near the origin and corresponds to low-frequency content, while edge pixels form a series of smaller peaks, where each peak corresponds to a class of edges having similar gradient magnitudes. Consequently, the high threshold should be selected between the largest peak and the second largest edge peak. Examples of gradient magnitude histograms for an edge/texture, medium edge and strong edge block are shown in Fig. 3.7. Based on the above observation, we proposed a non-uniform quantizer to discretize the gradient-magnitude histogram in [69]. Specifically, the quantizer needs to have more quantization levels in the region between the largest peak A and the second largest peak B and few quantization levels in other parts. Fig. 3.8 shows the schematic diagram of the proposed non-uniform quantizer. The first reconstruction level (R_1) is computed as the average of the maximum value and minimum value of the gradient magnitude in the considered block, and the second reconstruction level (R_2) is the average of the minimum value of the gradient magnitude and the first reconstruction level. Accordingly, n reconstruction levels can be computed as shown in [69], $R_1 = (min + max)/2$ and $R_{i+1} = (min + R_i)/2(i = 2, 3, \ldots n)$, where min and max represent the minimum and maximum values of the gradient magnitude, respectively, and R_i is the reconstruction level.



Step 1: Pixel classification

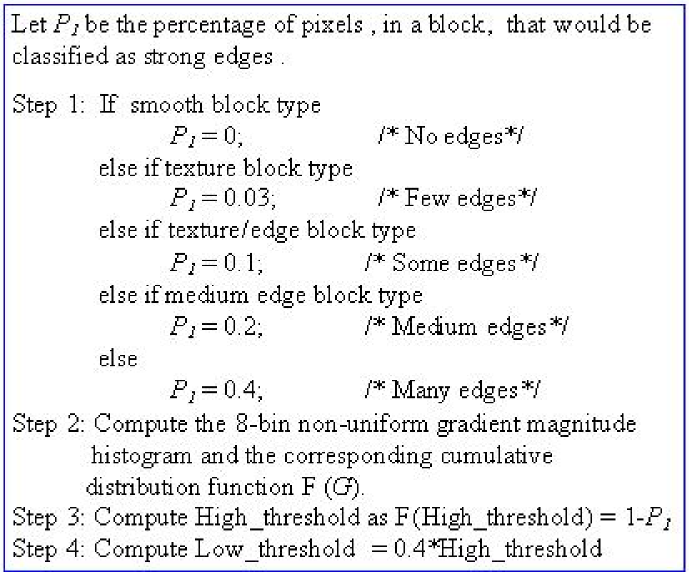
$$pixel\ type = \begin{cases} uniform, & \mathrm{var}(x,y) \le T_u \\ texture, & T_u < \mathrm{var}(x,y) \le T_e \\ edge, & T_e < \mathrm{var}(x,y) \end{cases}$$

Step 2: Block classification

Block type	No. of pixels of pixel type	
	$N_{uniform}$	N_{edge}
Smooth	$\ge 0.3 \cdot Total\_Pixel$	0
Texture	$< 0.3 \cdot Total\_Pixel$	0
Edge/texture	$< 0.65(Total\_Pixel - N_{edge})$	$(>0)\ \&\ (< 0.3 \cdot Total\_Pixel)$
Medium edge	$\ge 0.65(Total\_Pixel - N_{edge})$	$(>0)\ \&\ (< 0.3 \cdot Total\_Pixel)$
Strong edge	$\le 0.7 \cdot Total\_Pixel$	$\ge 0.3 \cdot Total\_Pixel$

$var(x, y)$: the local (3×3) variance at pixel (x, y);
T_u and T_e: two thresholds as in [17];
Total_Pixel: the total number of pixels in the block;
$N_{uniform}$: the total number of uniform pixels in the block;
N_{edge}: the total number of edge pixels in the block;

(a)



```
Let $P_1$ be the percentage of pixels , in a block,  that would be
classified as strong edges .

Step 1:  If  smooth block type
                 $P_1 = 0$;               /* No edges*/
         else if texture block type
                 $P_1 = 0.03$;            /* Few edges*/
         else if texture/edge block type
                 $P_1 = 0.1$;             /* Some edges*/
         else if medium edge block type
                 $P_1 = 0.2$;             /* Medium edges*/
         else
                 $P_1 = 0.4$;             /* Many edges*/
Step 2: Compute the 8-bin non-uniform gradient magnitude
        histogram and the corresponding cumulative
        distribution function F (G).
Step 3: Compute High_threshold as F(High_threshold) = 1-$P_1$
Step 4: Compute Low_threshold  = 0.4*High_threshold
```

(b)

Figure 3.9: Pseudo-codes for the Proposed (a) Block Classification And (b) Adaptive Threshold Selection.

The pseudo-code of the block classification technique in [67] and the proposed adaptive threshold selection algorithm is shown in Fig. 3.9a and 3.9b, respectively.

3.3 Parametrical Analysis

The performance of the proposed algorithm is affected by two parameters, the mask size and the block size. We discuss what is the best choice for the mask size for different types of images, and what is the smallest block size that can be used by our proposed distributed Canny algorithm without sacrificing performance.

3.3.1 The Effect of Mask Size

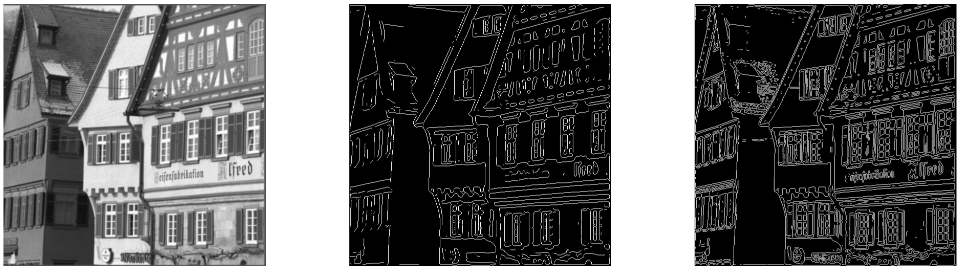
As indicated in Section 2.2.1, the size of the gradient mask is a function of the standard deviation σ of the Gaussian filter, and the best choice of σ is based on the image characteristics. Canny has shown in [39] that the optimal operator for detecting step edges in the presence of noise is the first derivative of the Gaussian operator. As stated in Section 2.2.1, for the original Canny algorithm as well as the proposed algorithm, this standard deviation is a parameter that is typically set by the user based on the knowledge of sensor noise characteristics. It can also be set by a separate application that estimates the noise and/or blur in the image. A large value of σ results in smoothing and improves the edge detector's resilience to noise, but it undermines the detector's ability to detect the location of true edges. In contrast, a smaller mask size (corresponding to a lower σ) is better for detecting detailed textures and fine edges but it decreases the edge detector's resilience to noise.

An $L-$point even-symmetric FIR Gaussian pulse-shaping filter design can be obtained by truncating a sampled version of the continuous-domain Gaussian filter of standard deviation σ. The size L of the FIR Gaussian filter depends on the standard deviation σ and can be determined as follows:

$$L = 2 \cdot L_{side} + 1 \tag{3.1}$$

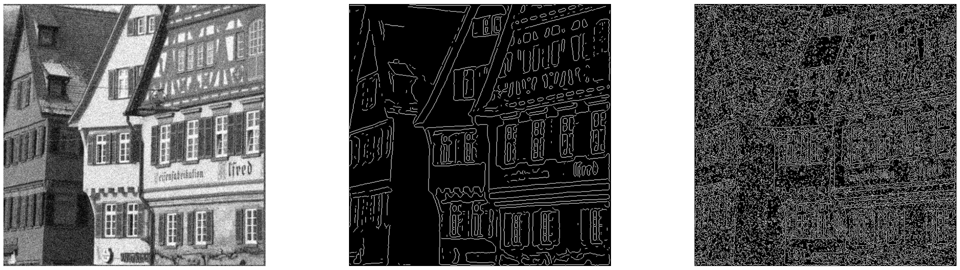
$$L_{side} = \lfloor 2\sigma\sqrt{(-1)log(C_T)} \rfloor \tag{3.2}$$

where C_T represents the cut-off value in the spatial domain of the continuous-domain Gaussian function and determines the cut-off error. A smaller C_T corresponds to a smaller cut-off error. In our implementation, in order to make the cut-off error small for all σ, C_T is chosen to be a small value (e.g., $C_T = 10^{-3}$). Fig. 3.10 shows that the relatively smaller size 3×3 filter (σ from 0.3 to 0.4) can detect the edges of a detailed texture image, while Fig. 3.11 shows that the larger



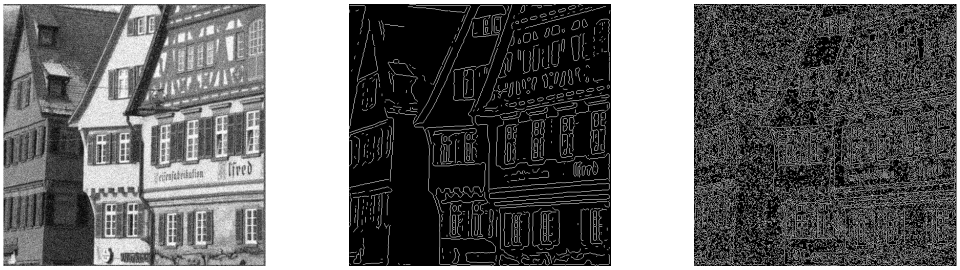
(a) (b) (c)

Figure 3.10: (a) Original ”Houses” Image; Edge Map Using the Original Canny Algorithm (b) with A 9×9 ($\sigma = 1.4$) Gradient Mask And (c) with A 3×3 ($\sigma = 0.4$) Gradient Mask.



(a) (b) (c)

Figure 3.11: (a) ”Houses” Image with Gaussian White Noise ($\sigma_n = 0.01$); Edge Map Using the Original Canny Algorithm (b) with A 9×9 ($\sigma = 1.4$) Gradient Mask And (c) with A 3×3 ($\sigma = 0.4$) Gradient Mask.



(a) (b) (c)

Figure 3.12: (a) Gaussian Blurred ”Houses” Image ($\sigma_{blur} = 2$); Edge Map Using The Original Canny Algorithm (b) with A 9×9 ($\sigma = 1.4$) Gradient Mask And (c) with A 3×3 ($\sigma = 0.4$) Gradient Mask.

9×9 filter (σ from 1.2 to 1.4) suppresses noise more effectively. In addition, Fig. 3.12 shows that the 3×3 filter is exhibits a better performance in generating the edge maps of blurred images as compared to the 9×9 filter.

3.3.2 Block Size

To find out the smallest block size for which the proposed Canny algorithm can detect all the psycho-visually important edges, the perceptual visual quality of the obtained edge maps was assessed using visual quality metrics. More specifically, the quality of edge maps obtained by using different block sizes was analyzed by evaluating their corresponding Pearson's correlation coefficient (PCC), which was used to measure how well state-of-the-art edge-based sharpness metrics commensurate with the actual perceived sharpness. For this purpose, the psycho-visual sharpness metric of [70] is used, which primarily depends on the edge map to estimate the cumulative probability of detecting blur (CPBD) in an image. This metric was shown in [70] to achieve the best performance among existing sharpness/blur metrics. It calculates the probability of detecting blur at each edge in an image and then obtains a final quality score over the entire image by evaluating the cumulative probability of blur detection. Four images with varying image characteristics were chosen. These were blurred using Gaussian masks with six different σ values (0.5, 1, 1.5, 2, 2.5, and 3) to generate a set of 24 images. The images were displayed one after another in a random order and subjects were asked to rate them on a 5-point scale of 1 (poor) to 5 (excellent) to form a Mean Opinion Score (MOS) by averaging the subjective scores for each image. Fig. 3.13 depicts the effect of replacing the frame-based Canny edge detection with the proposed distributed Canny edge detection on the sharpness metric of [70], for masks of different sizes and for different block sizes. Note that a 512×512 block corresponds to the original frame-based Canny edge detector. Fig. 3.13 shows that PCC has a maximal value when the block size is 64×64. It also illustrates that, for the various block sizes, the increase or decrease in the PCC is insignificant. In other words, the PCC is robust to changes in block size. In conclusion, the proposed distributed Canny algorithm can detect all the psycho-visually important edges for images with moderate noise and blur levels, similar to the original frame-based Canny algorithm.

3.4 Distributed Edge Detection Performance Analysis

The edge detection performance of the proposed distributed approach is analyzed by comparing the perceptual significance of its resulting edge map with the one produced by the original frame based

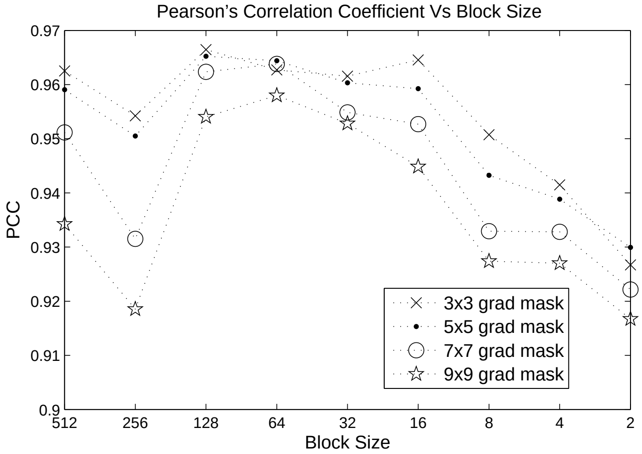


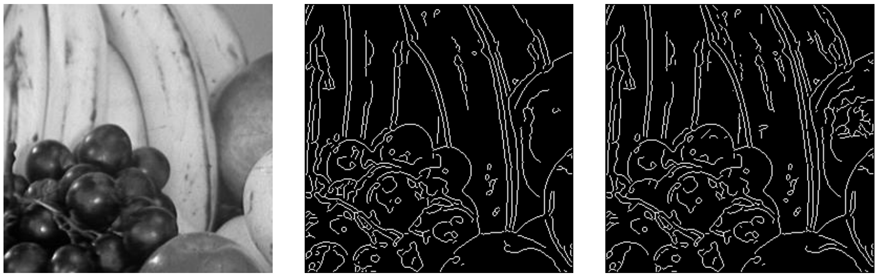
Figure 3.13: Pearson's Correlation Coefficient Reflecting the Correlation between the CPBD Sharpness Metric [70] And MOS as A Function of Block Size And Mask Size.

Canny edge detector. Fig. 3.14−Fig. 3.17 show the edge maps that are obtained by the original frame-based Canny edge detector and the proposed distributed Canny edge detection algorithm with 3×3 gradient masks and a non-overlapping block size of 64×64 $(n = 64; m = 68)$ for the 512×512 "Salesman," "Fruit," "Houses," and "Lena" images, respectively. As shown in Fig. 3.14 and Fig. 3.15, the proposed distributed Canny edge detection algorithm yields comparable edge detection results as compared with the original frame-based Canny edge detector. From Fig. 3.16 and Fig. 3.17, it can be seen that, in contrast to the original frame-based Canny, the proposed distributed Canny algorithm can detect edges in low contrast regions since it is able to adapt to the local statistical distribution of gradient values in a block. Furthermore, Fig. 3.18 shows the edge maps of images with white Gaussian noise generated by the original frame-based Canny edge detector and the proposed distributed Canny edge detection algorithm with 33 gradient masks and a non-overlapping block size of 64×64. Comparing the edge maps in Fig. 3.18 (b) and (c) with the clean edge maps in Fig. 3.14 to Fig. 3.17, it can be seen that, the proposed Canny edge detection algorithm is more robust to noise than the original frame-based Canny.

To further assess the performance of the proposed distributed Canny algorithm, quantitative conformance evaluations and subjective tests are performed. The conformance evaluations aim to evaluate the similarity between edges detected by the original frame-based Canny algorithm and the proposed distributed Canny edge detection algorithm, while the subjective tests aim to validate

(a) (b) (c)

Figure 3.14: (a) 512×512 "Salesman" Image; Edge-Maps of (b) the Original Canny Edge Detector, And (c) the Proposed Algorithm with A Non-Overlapping Block Size of 64×64, Using A 3×3 Gradient Mask.



(a) (b) (c)

Figure 3.15: (a) 512×512 "Fruit" Image; Edge-Maps of (b) the Original Canny Edge Detector, And (c) the Proposed Algorithm with A Non-Overlapping Block Size of 64×64, Using A 3×3 Gradient Mask.

(a) (b) (c)

Figure 3.16: (a) 512×512 "Houses" Image; Edge-Maps of (b) the Original Canny Edge Detector, And (c) the Proposed Algorithm with A Non-Overlapping Block Size of 64×64, Using A 3×3 Gradient Mask.

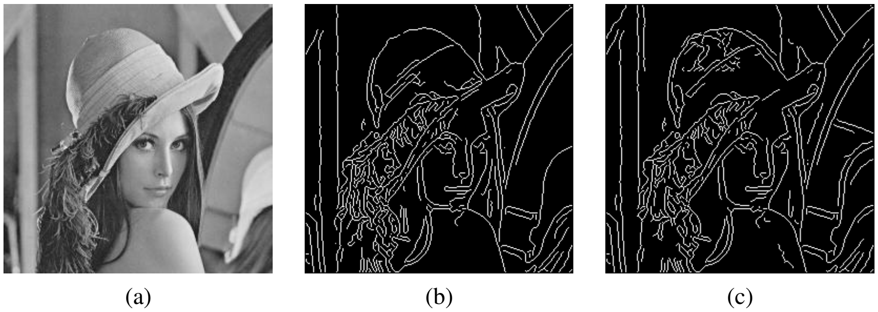


Figure 3.17: (a) 512×512 "Lena" Image; Edge-Maps of (b) the Original Canny Edge Detector, And (c) the Proposed Algorithm with A Non-Overlapping Block Size of 64×64, Using A 3×3 Gradient Mask.

whether the edge detection performance of the proposed distributed Canny is better, worse, or similar to the original frame-based Canny as perceived by subjects.

3.4.1 Conformance Evaluation

In order to quantify the similarity of two edge maps, three metrics, P_{co} (percentage of edge pixels detected by both implementations) P_{nd} (percentage of edge pixels detected by the original Canny edge detection algorithm that were not detected by the proposed distributed Canny edge detection algorithm also referred to as false negatives) and P_{fa} (percentage of edge pixels detected by the proposed Canny edge detection algorithm that were not detected by the original Canny edge detection algorithm, also referred to as false positives), as proposed in [52], are used for the conformance test. Here, the edge map generated by the original frame-based Canny for the clean image is considered as the reference for both the clean and noisy cases. This is compared with the edge map generated by the proposed algorithm for the same clean image, as well as the edge maps generated by these two algorithms (original Canny and proposed) for the noisy version of the same image. Experiments were conducted using the USC SIPI Image Database [71] and the Standard Test Image Database [72]. The results are shown in Table 3.3.

For clean images, the detected edge pixel percentage (P_{co}) was higher than 94.8%. This further validates our claim that the proposed distributed algorithm detects almost all edges that are detected by the original Canny edge detection algorithm.

For noisy images, as shown in Table 3.3, the P_{co} of the proposed distributed Canny algorithm

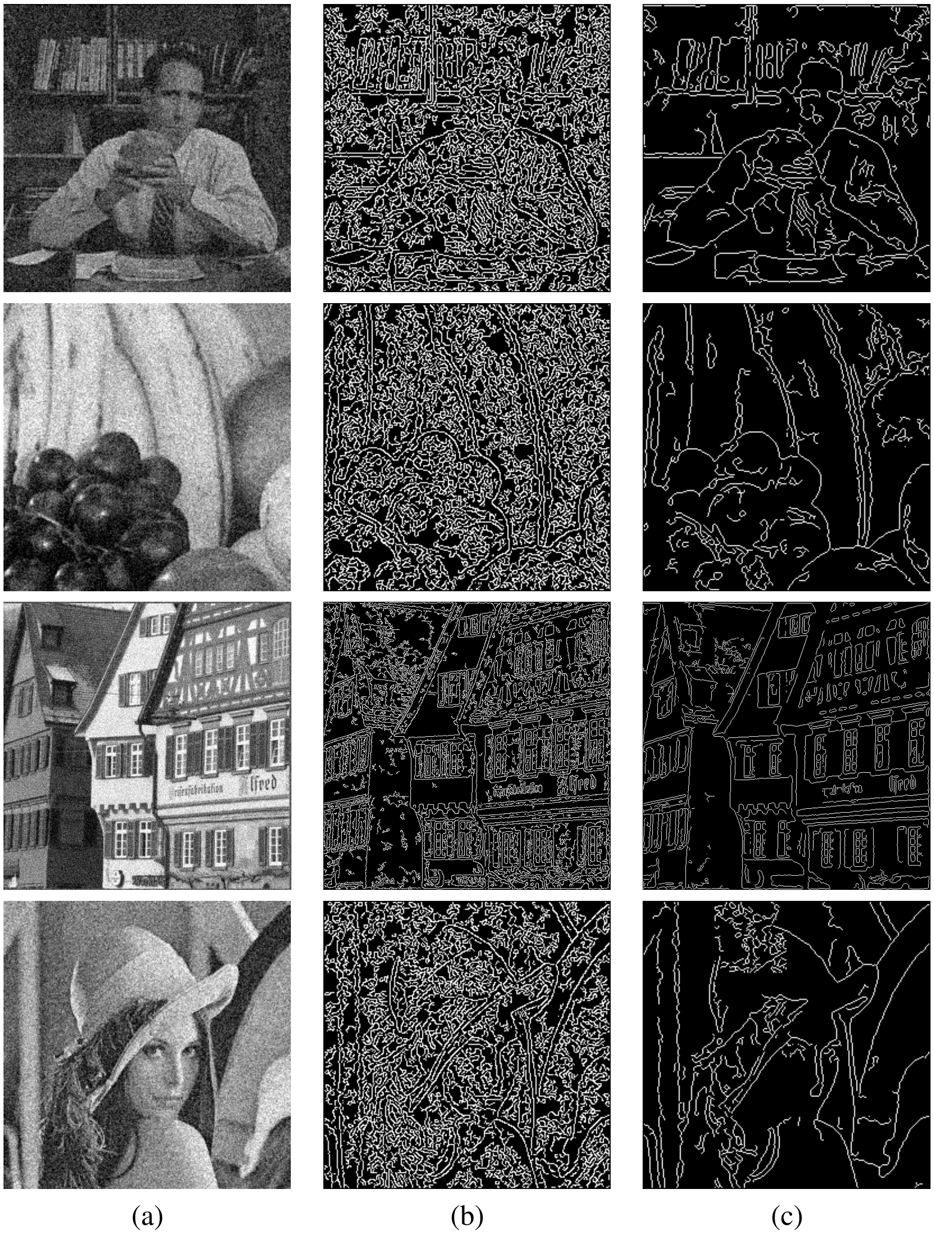

(a) (b) (c)

Figure 3.18: Comparison of the Edge Maps of Noisy Images by Using the Original Canny Edge Detector And the Proposed Method: (a) Images with Gaussian White Noise ($\sigma_n = 0.01$); Edge-Maps of (b) the Original Canny Edge Detector, And (c) the Proposed Algorithm with A Non-Overlapping Block Size of 64×64, Using A 9×9 Gradient Mask.

is higher than that of the original Canny algorithm, and the error (the sum of P_{nd} and P_{fa}) of the proposed distributed Canny algorithm is much lower than the original Canny detector. This validates our earlier observation that, for noisy image, the edge detection performance of the proposed

Table 3.3: Conformance Evaluation.

	Clean Image	Noisy Image	
	Proposed Canny detector	Original Canny detector	Proposed Canny detector
P_{co}	94.8%	56.2%	64.9%
P_{nd}	1.2%	13.7%	16.6%
P_{fa}	4%	30.1%	18.5%


Compare Right Edge Map to Left Edge Map

Counts (out of 60) 1

Output Filename (.mat file will be created with subject ratings)

Figure 3.19: Snapshot of the Performed Subjective Test Comparing the Edge Maps Generated by the Original Canny Algorithm And the Proposed Algorithm.

Figure 3.20: Subjective Scale Where Scores 1, 2, 3, 4, & 5 Correspond to ”Worse,” ”Slightly Worse,” ”Same,” ”Slightly Better,” And ”Better,” with Respect to the Original Canny Algorithm.

distributed Canny algorithm is better than the original frame-based Canny algorithm.

3.4.2 Subjective Testing

Subjective tests were conducted by having human subjects evaluate the quality of the detected edge maps that are generated by the proposed algorithm and the original Canny for both clean and noisy images, without the subjects knowing which algorithm produced which edge maps, using images from the SIPI Database [71] and the Standard Test Image Database [72].

The edge maps of clean images obtained by the original Canny algorithm and the proposed algorithm were displayed simultaneously for comparison together with the original image, which was displayed in the middle. Fig. 3.19 shows a snapshot of the subjective test interface. This was done for both clean and noisy images. Experiments were conducted using a 19-inch DELL LCD monitor having a resolution of 1024×1280. The average room luminance is measured to be $30cd/m_2$. The subjects were asked to rate the quality of the displayed edge maps using a five-point scale with scores 1 to 5 corresponding, respectively, to the rightmost edge map is ”worse,” ”slightly worse,” ”same,” ”slightly better,” and ”better” compared to the edge map shown on the left. These scores were then automatically converted internally by the interface into five-point scale scores from 1 to 5 corresponding, respectively, to the edge map produced by the proposed distributed Canny algorithm is ”worse,” ”slightly worse,” ”same,” ”slightly better,” and ”better” compared to the original Canny algorithm (see Fig. 3.20). Thus, a score higher than 3 indicates that the generated edge map by the proposed algorithm is better than the one obtained using the original Canny algorithm and vice versa. The images are randomly displayed. Each case is randomly repeated four times, with the left and right images swapped in a random fashion, to obtain better subjective response statistics [73]. Ten subjects with normal to corrected-to-normal vision took the test. Two Mean Opinion Scores (MOS) were computed by averaging the subjective scores over the set of clean images and over the set of noisy images, respectively. The obtained MOS values were 3.69 (slightly better than the original Canny) for the clean image set and 4.6 (better than the original Canny) for the noisy image set. Results from the subjective tests show clear preference of the subjects for the edge maps that are generated by the proposed algorithm for both clean and noisy images. These results indicate that the proposed algorithm results in a superior performance as compared to the original frame-based Canny algorithm. This is partly due to the capability of the proposed algorithm to adapt to the local image characteristics since the P_1 value is selected differently for different types of blocks by adapting its value to the local block content.

3.4.3 FPGA-Based Performance Results

In order to demonstrate the parallel efficiency of the proposed distributed Canny edge detection algorithm, we describe an FPGA-based hardware implementation of the proposed algorithm in this chapter. Fig. 3.21 gives a bird’s eye view of the embedded system for implementing the distributed Canny edge detection algorithm based on an FPGA platform. It is composed of several compo-

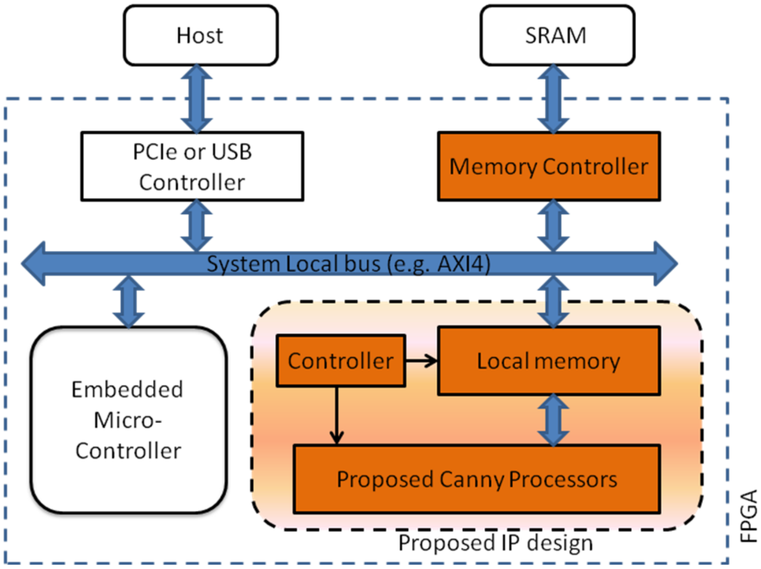


Figure 3.21: Block Diagram of the Embedded System for the Proposed Algorithm.

nents, including an embedded micro-controller, a system bus, peripherals & peripheral controllers, external Static RAMs (SRAM) & memory controllers, and an intellectual property (IP) design for the proposed distributed Canny detection algorithm. The embedded micro-controller coordinates the transfer of the image data from the host computer (through the PCIe (or USB) controller, system local bus, and memory controller) to the SRAM; then from the SRAM to the local memory in the FGPA for processing and finally storing back to the SRAM. Xilinx and Altera offer extensive libraries of intellectual property (IP) in the form of embedded micro-controllers and peripherals controller [74, 75]. Therefore, in our design, we focused only on the implementation of the proposed algorithm on the Xilinx Virtex-5 FPGA and the data communication with external SRAMs. These components have been highlighted in Fig. 3.21.

The proposed architecture, shown in Fig. 3.22, consists of q processing units (PU) and external dual-port Static RAMs (SRAMs) to store image data. As shown in Fig. 3.22, each PU consists of p computing engines (CE), where each CE processes an $m \times m$ overlapping image block and generates the edges of an $n \times n$ block, where $m = n + L + 1$ for an $L \times L$ gradient mask. The dataflow through this architecture is as follows. For each PU, the SRAM controller fetches the input data from SRAM and stores them into the input local memory in the PU. The CEs read this data, process them and store the edges into the output local memory. Finally, the edges are written

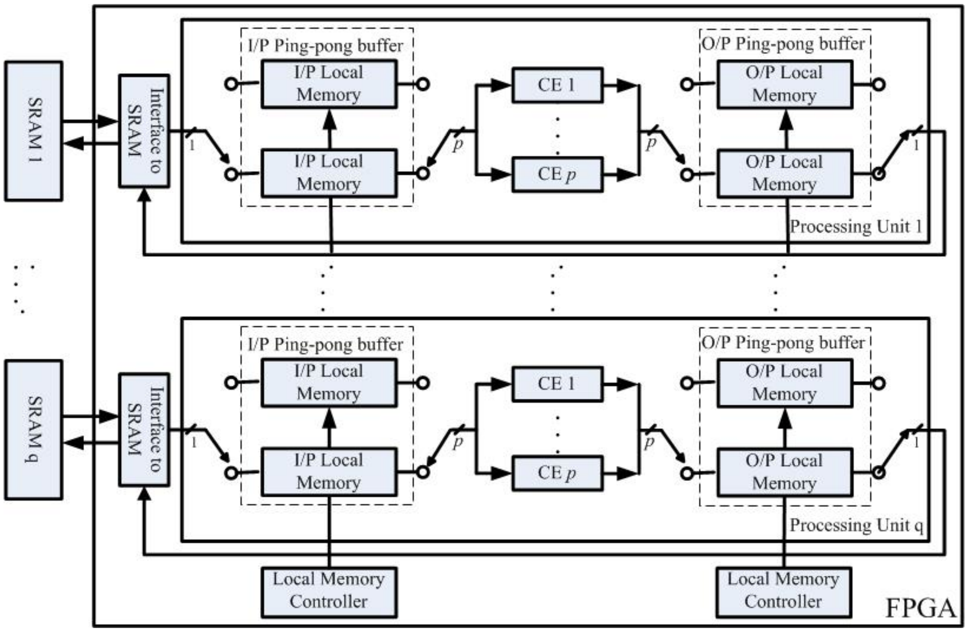


Figure 3.22: The Architecture of the Proposed Distributed Canny Algorithm.

back to the SRAM one output value at a time from the output local memory.

In order to increase the throughput, the SRAM external memory is organized into q memory banks, one bank per PU. Since only one b-bit data, corresponding to one pixel value, can be read from a SRAM at a time, such an organization helps multiple PUs to fetch data at the same time and facilitates parallel processing by the PUs. For an image of size $N \times N$, each SRAM bank stores a tile of size N^2/q image data,where the term ”tile” refers to an image partition containing several non-overlapping blocks. The SRAM bank is dual-port so that the PUs can read and write at the same time.

In order to maximize the overlap between data read/write and data processing, the local memory in each PU is implemented using dual port block RAM (BRAM) based ping-pong buffers. Furthermore, in order that all the p CEs can access data at the same time, the local memory is organized into p banks. In this way, a total of $p \cdot q$ overlapping blocks can be processed by q groups of p CEs at the same time. The processing time for an $N \times N$ image is thus reduced approximately by a factor of $p \cdot q$. If there are enough hardware resources to support more CEs and more PUs, the throughput would increase proportionally.

However, FPGAs are constrained by the size of on-chip BRAM memory, number of slices,

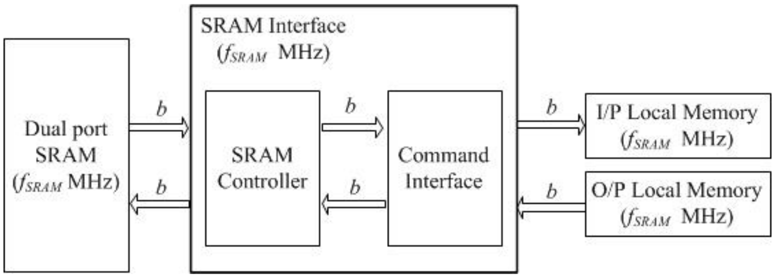


Figure 3.23: High Throughput Memory Interface.

number of I/O pins; and the maximum throughput achievable is a function of these parameters. Since each CE processes an $m \times m$ overlapping image block, for a b-bit implementation, this would require $3 \times m \times m \times b$ bits to store the vertical and horizontal gradient components and the gradient magnitude. To enable parallel access of these blocks, there are three BRAMs of size $m \times m \times b$ in each CE. In addition, $2p \times m \times m \times b$ bits are needed for each of the ping-pong buffers. Therefore, for each PU, $3p \times m \times m \times b$ bits are need for p CEs and $4p \times m \times m \times b$ bits are needed for the input and output ping-pong buffers. This results in a total of $7p \times m \times m \times b \times q = 7pqm^2b$ bits for the FPGA memory. Thus, if there are more CEs and/or larger sized block, more FPGA memory is required. Similarly, if there are more PUs, more I/O pins are required to communicate with the external SRAM memory banks. Thus, the choice of p and q depends on the FPGA memory resources and numbers of I/O pins. We do not consider the numbers of slices as a constraint since the number of available slices is much larger than required by the proposed algorithm.

The customized memory interface, shown in Fig. 3.23, has a $2b$-bit wide internal data-bus. In our application, the dual-port SRAM, the memory interface and the local memories, which connect with the SRAM interface, operate at the same frequency, which is f_{SRAM} MHz.

Each CE processes an $m \times m$ overlapping image block and generates the edges of an $n \times n$ non-overlapping block. The computations that take place in CE can be broken down into the following five units: 1) block classification, 2) vertical and horizontal gradient calculation as well as magnitude calculation, 3) directional non-maximum suppression, 4) high and low threshold calculation, and 5) thresholding with hysteresis. Each of these units is mapped onto a hardware unit as shown in Fig. 3.24 and described in the following subsections. The communication between each component is also illustrated in Fig. 3.24. More details of the architecture for each component can be

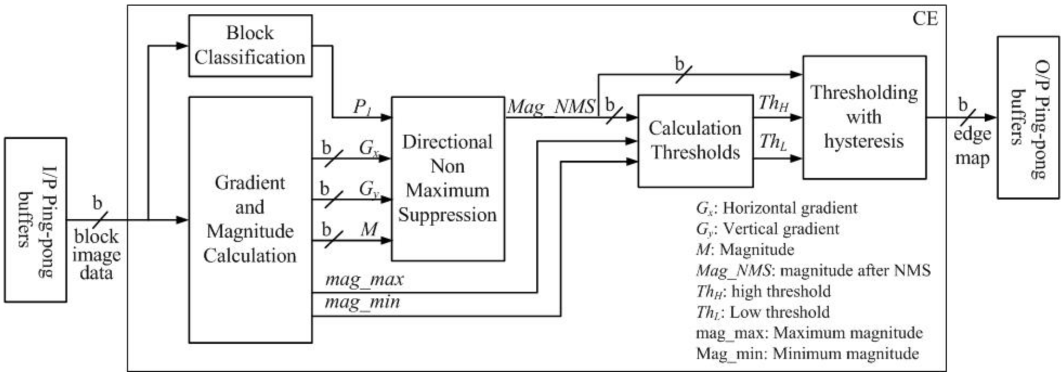


Figure 3.24: Block Diagram of CE (Computing Engine for Edge Detection).

found in [**?**].

The proposed FPGA-based architecture can support multiple image sizes and block sizes. To demonstrate the performance of the proposed system, a Xilinx Virtex-5 FPGA [76] was used to process grayscale images with a block size of 64×64. The data width is 16 bits (Q8.7) with 8 bits to represent the integer part since the maximum gray value of the image data is 255, and 7 bits to represent the fractional part since the Gaussian filter parameters are decimals. Our analysis shows that 7 bits are sufficient to meet the accuracy requirement of the Gaussian filter parameters, which is typically in the order of 0.001.

To store grayscale images, we used the SRAM (CY7C0832BV) [77]. This is a dual ported SRAM with 110 pins. The Xilinx Virtex-5 FPGA (XC5VSX240T) has 960 I/O pins and so, to satisfy the I/O pin constraint, the maximum number of PUs is 8 ($q = 8$). The local memory on the FPGA for a block size of 64×64, which is needed to support 8 PUs, is equal to $7pqm^2b = 4046p$ Kbits, for $q = 8$, $m = 68$ (for a 64×64 block size and a $3\times$ gradient mask size), and $b = 16$. Since the available memory resource on the FPGA is 18,576 Kbits, the p value using the memory constraint is determined to be 4. The p value could have also been constrained by the number of available slices. Since the number of slices for the considered FPGA is very large (37,440) and since each CE only utilizes a small slice percentage, the local memory resource in each PU constrains p, the number of CEs in each PU, and not the numbers of slices. Taking all this into consideration, our design has $q = 8$ PUs and each PU has $p = 4$ CEs. This design is coded in Verilog and synthesized on a Xilinx Virtex-5 device (XC5VSX240T) using the Xilinx's ISE software and

Table 3.4: Resource Utilization On XC5VSX240T for 1 CE.

Block Size	Number of CE	Occupied slices	Slice Reg.	Slice LUTs	DSP48Es	Total used Memory (KB)
64×64	1	747 (2%)	1,270 (1%)	2,578 (1%)	7 (1%)	217 (1%)

Table 3.5: Resource Utilization on XC5VSX240T for 1 PU (4 CEs and 2 Ping-Pong Buffers).

Block Size	Number of CE	Occupied slices	Slice Reg.	Slice LUTs	DSP48Es	Total used Memory (KB)
64×64	4	2,988 (8%)	5,080 (4%)	10,312 (4%)	28 (3%)	2023 (10%)

Table 3.6: Resource Utilization on XC5VSX240T for An 8-PU Architecture.

Block Size	Number of CE	Occupied slices	Slice Reg.	Slice LUTs	DSP48Es	Total used Memory (KB)
64×64	32	23,904 (64%)	40,640 (32%)	82,496 (34%)	224 (25%)	16,184 (87%)

verified using Modelsim. According to the ”Place and Route” synthesis report, our implementation can achieve an operating frequency of 250 MHz. But we choose 100 MHz to support a pipelined implementation of SRAM read/write and CE processing as described later in this section.

The FPGA resource utilization in each CE is listed in Table 3.4, followed by the resource utilization for each PU in Table 3.5. The FIR IP core and the arithmetic functions are implemented using DSP48Es. The on-chip memory is implemented using BRAMs. Table 3.6 summarizes the resource utilization of the 8-PU architecture. It shows that the 8-PU architecture occupies 64% of the slices and 87% of the BRAM memory.

Fig. 3.25 shows the pipeline implementation of SRAM read/write with the CE computation, where each SRAM bank stores a tile of size N^2/q image data and each ping or pong buffer stores a group of blocks (GOB) of size $p \times m \times m$ image data. Since our design has $q = 8$ PUs, one SRAM can hold a tile of size 32,768 ($64 \times 64 \times 8$) image data for a 512×512 image. In addition, for $p = 4$ CEs and for a 64×64 block size ($n = 64$; $m = 68$), the image data stored in the SRAM result in two GOBs. These GOBs can be pipelined. As shown in Fig. 3.25, while GOB 2 is loaded to the ping-pong buffers, the CEs process GOB 1. Also, while GOB 1 is written back into SRAM, the CEs process GOB 2 at the same time. Such a pipelined design can increase throughput.

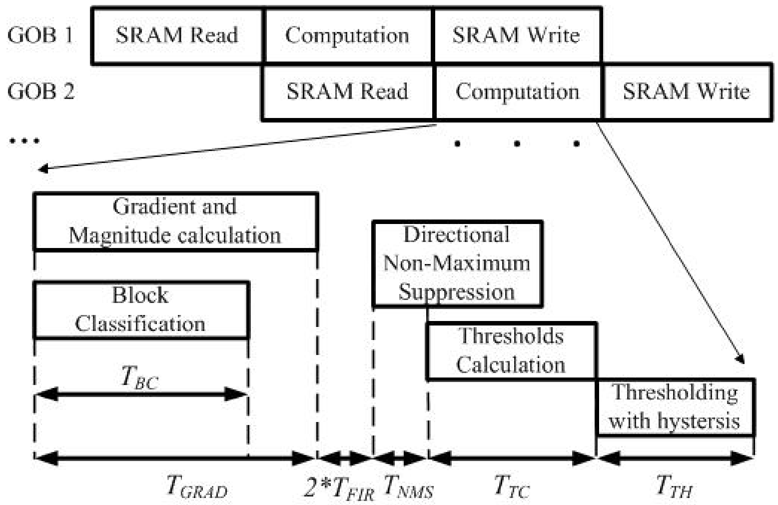


Figure 3.25: Execution Time of the 512×512 Image.

Table 3.7: Clock Cycles for Each Unit.

	T_{GRAD}	T_{FIR}	T_{NMS}	T_{TC}	T_{TH}
Clock Cycles	9248	16	20	4630	4634

Fig. 3.25 also shows the computation time of each stage in a CE during the processing of an $m \times m$ overlapping block ($m = 68$ for a 64×64 block and a 3×3 gradient mask). As shown in Fig. 3.25, T_{BC}, the time to classify the block type, is less than T_{GRAD}, the time for Gradient and Magnitude calculation, which equals to 9248 clock cycles. T_{FIR}, the FIR filter computation latency equals to 8 clock cycles for a 3×3 FIR separable filter. The high and low thresholds calculation unit is pipelined with the directional NMS unit and the latency of the NMS unit is 20 clock cycles. This is referred to as T_{NMS} in Fig. 3.25. T_{TC} represents the latency of the thresholds calculation stage and is equal to 4630 cycles, while T_{TH} represents the latency of the thresholding with hysteresis stage and is equal to 4634 cycles. Table 3.7 shows the latency for each unit. Therefore, one CE takes $T_{CE} = T_{GRAD} + T_{FIR} + T_{NMS} + T_{TC} + T_{TH} = 18,548 cycles$.

Each PU takes 18,496 cycles to load 4 68×68 overlapping blocks from the SRAM into the local memory. It also takes 16,384 (4 64×64 non-overlapping blocks) cycles to write final edge maps into SRAM. If SRAM operates at f_{SRAM}, the SRAM read time is $18,496/f_{SRAM}$. The CE processing time equals to $18,548/f_{CE}$ when CE is clocked at f_{CE}. In order to completely overlap communication with computation and avoid any loss of performance due to communication with

SRAM, given a fixed f_{SRAM}, f_{CE} should be selected such that the processing time is approximately equal to the SRAM read time (since the SRAM write time is less than the read time). Thus, $18,496/f_{SRAM} = 18,548/f_{CE}$, and the f_{CE} can be set to be 1.003 times higher than f_{SRAM}.

The maximum speed of the employed SRAM device (CY7C0832BV) is 133MHz. However, we choose the SRAM clock rate as $f_{SRAM} = 100MHz$ to allow for sufficient design margin. Thus, $f_{CE} \approx 100MHz$, which is lower than the maximum operating frequency (250 MHz) of the used FPGA according to the synthesis report. The total computation period for one CE is $T_{CE} = 18,548/105 \approx 0.186ms$ when clocked at 100 MHz. Thus, for a 512×512 image, the total computation period is $T_{com} = 0.186 \times 2 = 0.372ms$; while the total execution time, including the SRAM read/write time and the computation time, is $T_{total} = (18,496+16,384)/10^5 + 0.186 \times 2 = 0.721ms$. The simulation results also show that, at a clock frequency of 100MHz, the execution time for processing 512×512 images is 0.721ms for the images in the USC SIPI database.

In order to validate the FPGA generated results, two conformance tests are performed. One aims to evaluate the similarity between edges detected by the fixed-point FPGA and Matlab implementation of the distributed Canny edge detection algorithm. The other is to measure the similarity between edges detected by the fixed-point FPGA and the 64-bit floating-point Matlab implementation of the distributed Canny edge detection algorithm. Our results for both tests were performed on the USC SIPI image database.

For the first test, the difference between the edges detected by the fixed-point FPGA and Matlab implementations is calculated. Fig. 3.26 shows an example of the obtained fixed-point Matlab and FPGA results for the Houses image using the proposed algorithm with a 64×64 block size and 3×3 gradient masks. The FPGA simulation result is obtained using Modelsim and assumes the original image data has been stored in SRAMs. It can be seen that the two edge maps are the same. Furthermore, the quantitative difference between the FPGA simulation result and the fixed-point Matlab simulation result is zero. The same results were obtained for all 26 images in dataset.

In order to verify similarity of the edges generated by the fixed-point FPGA and 64-bit floating-point Matlab implementation of our proposed algorithm, the P_{co}, P_{nd} and P_{fa}, metrics as described in Section 3.4.1, were used. For all images, the correctly detected edge pixel percentage (P_{co}) was higher than 98.2%. Furthermore, less than 1.8% of all edge pixels were detected by only one of

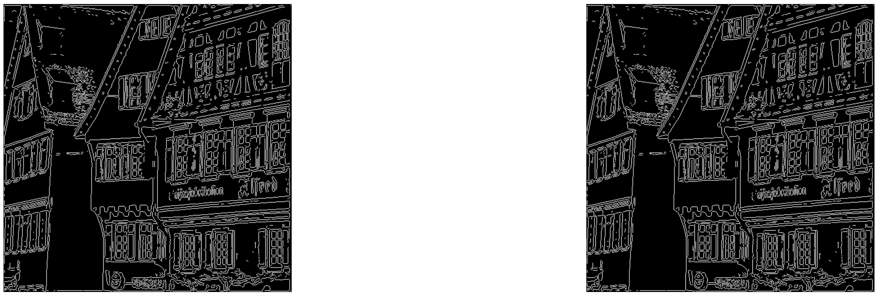

(a) (b)

Figure 3.26: Comparison between the Fixed-Point Matlab Software Simulation Result And FPGA Hardware Simulation Result For Detecting Edges of the 512×512 Houses Image Using A Block Size of 64×64 And A 3×3 Gradient Mask: (a) Edge Map Generated by the Fixed-Point MATLAB Implementation; (b) Edge Map Generated by the FPGA Implementation.

the detectors (P_{nd} and P_{fa}). This leads to the conclusion that the hardware implementation of our proposed algorithm can successfully detect significant edges.

Furthermore, to show the efficiency of the FPGA-based system, we compared the FPGA implementation of the original frame-based Canny edge detector with an assembly optimized CPU implementation of the original Canny edge detector available in Intel's OpenCV library, which makes good use of thread parallelization and SSE instructions. Also, to show the efficiency of the FPGA implementation of our proposed distributed Canny algorithm, the proposed FPGA distributed Canny implementation is compared with the FPGA and CPU implementations of the original frame-based Canny edge detector. The two candidate hardware platforms were:

- CPU: Intel Core i7-975, four 3.33G cores, 8192KB cache and 12GB RAM;
- FPGA: Xilinx SXT Virtex-5, 37,440 slices, 1,056 DSP48E slices and 18,576 Kbits BRAM

Runtime results are shown in Table 3.8 for the standard "Lena" image. The absolute runtimes of the two algorithms were recorded and averaged over 100 times. Since the FPGA runtime does not include the image loading time from the host to the external SRAMs, for fair comparison, the loading time from the HDD to the RAM for the CPU implementation is also not included in the reported runtime. In general, the CPU loading time is shorter than the FPGA loading time since the speed of the internal CPU bus is faster than when using an external bus such as PCI or PCI external.

Table 3.8: Runtime Comparison On Various Platforms for the "Lena" Image at Different Sizes.

Image Size	Frame-based Canny		Proposed distributed Canny (64×64 non-overlapping block)
	OpenCV (ms) @3.33 GHz	FPGA (ms) @100 MHz	FPGA (ms) @ 100 MHz
256×256	2.206	0.67	0.535
512×512	5.97	2.74	0.721
1024×1024	20.852	12.02	1.837
2048×2048	76.102	56.30	6.301

From Table 3.8, it can be seen that the speed of the FPGA implementation for the original frame-based Canny edge detection is faster than the frame-based CPU implementation even though, for the CPU implementation, the CPU is clocked 33 times faster. Furthermore, the FPGA implementation for the proposed distributed Canny edge detection is much faster than the frame-based FPGA and CPU implementations. This illustrates the efficiency of the proposed distributed Canny edge detector.

A long standing challenge for the use of the Canny edge detection algorithm for real-time applications is its high computational complexity. Several FPGA and GPU implementations have been proposed to meet the requirements of real-time applications. Comparisons between our proposed FPGA implementation and other existing FPGA and GPU implementations are discussed below.

As far as we know, the existing FPGA-based approaches implemented the frame-based Canny algorithm and only the computation time for detecting edges are presented in papers [47–49]. The computation time and resource utilization results of these frame-based Canny FPGA implementations and our distributed Canny FPGA implementation are shown in Table 3.9. Since these approaches are implemented for different image sizes and operated at different clock frequencies, their results have to be normalized for a fair comparison. The rightmost column of Table 3.9 gives the normalized computing time, where the normalization is done with respect to an image of size 512×512 and an FPGA clock frequency of 100 MHz. From Table 3.9, it can be seen that the computation time of the proposed distributed implementation is much faster than the existing implementations. Furthermore, the implementations [48, 49, 64] result in degraded performance compared to our implementation since the thresholds are fixed to predetermined values, while our

Table 3.9: Computation Time of the Proposed Distributed Canny FPGA Implementation Using A 64 × 64 Non-Overlaping Block Size And A 3 × 3 Gradient Mask And of Existing Frame-Based Canny FPGA Implementations.

	Image Size	FPGA Device	Occupied Slices	Total used Memory (KB)	Highest Freq. (MHz)	Time (ms)	Norm. Time (ms)
[47]	256×256	Xilinx Vertex-E	-	-	16	4.2	2.688
[48]	256×256	Altera Stratix II	1,530 /48,352	1,116 /2,044	264	0.25	2.64
[64]	360×280	Altera Cyclone	-	-	27	2.5	0.72
[49]	512×512	Xilinx Virtex-5	4,553 /71,680	192 /5,328	292.8	0.57	1.669
Our	512×512	Xilinx Virtex-5	23,904 /37,440	16,184 /18,576	250	0.15	0.372

Table 3.10: Runtimes of the GPGPU Implementations And Our 32-CE FPGA Implementation.

	GTX 80 [50]	GT 80 [52]	GT 200 [52]	Fermi [52]	32-CE FPGA
Image Size	512×512	321×481	321×481	321×481	512×512
Freq (MHz)	768	1500	1300	1150	100
Time (ms)	3.4	5.47	2.95	2.3	0.721
Norm. Time (ms)	26.112	139.31	65.11	44.91	0.721

implementation computes the hysteresis thresholds adaptively based on the input image characteristics. In addition, compared to other frame-based FPGA implementations, the proposed implementation enables a higher throughput due to its distributed nature, at the expense of more memory and slice usage.

Also, the GPU-based Canny implementations in [50] and [52] are compared to our 32-CE FPGA Canny implementation. One NVidia GPGPU was used in [50] and three NVidia GPGPUs were used in the recent GPU Canny implementation [52]. The GPU configuration details are given below:

- GTX 80 [50]: NVidia GeForce 8800 GTX with 128 575-MHz cores and 768MB RAM;
- GT 80 [52]: NVidia GeForce 8800 GT with 112 1.5-GHz cores and 512MB RAM;

- GT 200 [52]: NVidia Tesla C1060 with 240 1.3-GHz cores and 4GB RAM;
- Fermi [52]: NVidia Tesla C2050 with 448 1.15-GHz cores and 3GB RAM.

Since the FPGA runtime does not include the image loading time from the host to the external SRAMs, the loading time for GPU implementations was removed from the total runtime for a fair comparison. Table 3.10 shows the runtimes of the GPGPU implementations and our 32-CE FPGA implementation (with a 64×64 block size and a 3×3 gradient mask) for processing a 512×512 image. Even though all these existing GPGPU implementations do not have an adaptive threshold calculation stage and the FPGA is operated at a much slower frequency, our 32-CE FPGA implementation is faster than the GPGPU implementations.

3.5 Summary

The original Canny algorithm relies on frame-level statistics to predict the high and low thresholds and thus has latency proportional to the frame size. In order to reduce the large latency and meet real-time requirements, we presented a novel distributed Canny edge detection algorithm which has the ability to compute edges of multiple blocks at the same time. To support this, an adaptive threshold selection method is proposed that predicts the high and low thresholds of the entire image while only processing the pixels of an individual block. This results in three benefits: 1) a significant reduction in the latency; 2) better edge detection performance; 3) the possibility of pipelining the Canny edge detector with other block-based image codecs; 4) no memory resources constraint for large-size images. In addition, a low complexity non-uniform quantized histogram calculation method is proposed to compute the block hysteresis thresholds. The proposed algorithm is scalable and has very high detection performance. We show that our algorithm can detect all psycho-visually important edges in the image for various block sizes. Finally, the algorithm is mapped onto a Xilinx Virtex-5 FPGA platform and tested using ModelSim. The synthesized results show 64% slice utilization and 87% BRAM memory utilization. The proposed FPGA implementation takes only 0.721ms (including the SRAM read/write time and the computation time) to detect edges of 512 × 512 images in the USC SIPI database when clocked at 100 MHz. Thus the proposed implementation is capable of supporting fast real-time edge detection of images and videos including those with full-HD content.

Chapter 4

OVERVIEW OF CHANGE DETECTION IN MULTI-TEMPORAL SAR IMAGES

4.1 Introduction

Detecting temporal changes occurring on the earth surface by observing them at different times constitutes one of the most important use of remote sensing technology. Especially, due to the all-weather operating ability of synthetic aperture radar (SAR) imagery, multi-temporal SAR image change detection has many applications, such as environmental monitoring, agricultural surveys, urban studies, and forest monitoring [78]. One critical issue in multi-temporal SAR images is the detection of changes occurring after a natural or anthropic disaster.

This change-detection procedure is made difficult due to three reasons. The first one is the time constraints imposed by the emergency context. Indeed, the first available acquisition after the event has to be used, whatever is the modality, which is more likely to be a radar image, due to weather and daylight constraints. Secondly, it is hard to model the type of changes produced by the event of interest. For example, an earthquake can have different signature depending on when or where it happens,(e.g., high-density built-up areas, agricultural area, etc.). Finally, the changes of interest are all mixed up with normal changes if the time gap between the two acquisitions is too long [54]. All these issues present us with a tough problem: detecting abrupt unmoulded changes in a temporal series.

Over the years, a variety of methods have been proposed to solve this problem in the literature. According to the data sources, the existing methods fall into two categories: bi-temporal change detection and image time series change detection [79, 80]. Furthermore, most bi-temporal change detection techniques can be classified into supervised and unsupervised change detection methods. In supervised change detection, reliable training samples based on the prior knowledge of the research scenario are selected to train a classifier, which will be used after training to classify each pixel as changed or unchanged pixel. Recently, support vector machines have been widely applied for supervised change detection [81–83]. On the contrary, in unsupervised change detection, the first step is to compare directly image features or even the original two images by some similarity metrics resulting in a similarity image, and then threshold or label the similarity image to derive a binary change map consisting of two classes associated with changed and unchanged pixels. As a result, unsupervised approaches are preferable to supervised approaches in practice as training samples are not always available and since training is expensive in terms of both time and cost in order to obtain reliable training data.

Supervised change detection can be performed as a binary classification including post classification comparison [84], [85], direct multi-data classification [84], [86] and compound classification [87–89]. Post classification comparison performs change detection by comparing the change map derived from the independent classification of the two images, while the direct multi-data classification achieves the change map by classifying the concatenated features of the two images. Compound classification performs the change detection by maximizing the posterior joint probabilities of classes. Recently, support vector machines have been widely applied for supervised change detection. In [81], inductive SVM was initially used for classifying features characterizing changes at object-level and then the classification was refined by an iterative transductive SVM. SVM using a stochastic kernel [90] was applied for change detection based on similarity measures of the local statistics. An approach for unsupervised change detection was proposed in [82] using a selective Bayesian thresholding for deriving a training set that is used for initializing a binary semi-supervised support vector machine (S3VM) classifier. In [83], change detection was formulated as a minimum enclosing ball (MEB) problem which was approached by a Support Vector Domain Description (SVDD) classifier. One advantage of these methods is their ability to capture various changes denoted by a transition map. However, in practice, training samples are not always available for forming a reliable set of training data.

In general, most unsupervised change detection methods include three steps as shown in Fig. 4.1: 1) preprocessing, 2) image comparison to generate a similarity image, and 3) thresholding the similarity image to compute the final binary change detection map associated with changed and unchanged classes. The overall performance of the detection system will depend on both the quality of the similarity image and the quality of the thresholding method. Usually, there are two aims that the preprocessing step attempt to achieve. One aim is to make two images as similar as possible. Typically radiometric corrections and image registration techniques are involved. The other aim of preprocessing is to increase the SNR of the considered images (by reducing noisy speckle components), while preserving sufficient spatial details. Many adaptive filters for speckle reduc-

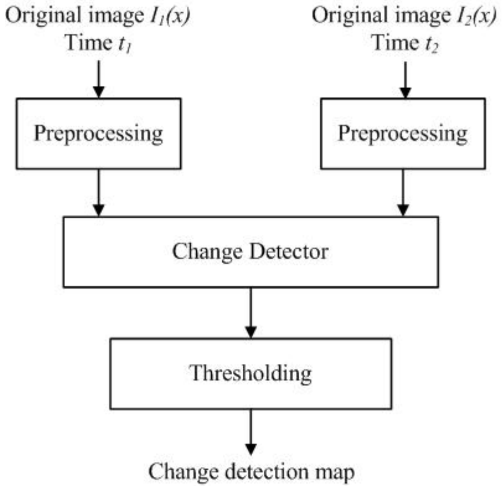


Figure 4.1: General Block Diagram of the Change Detection.

tion have been proposed, e.g., the Frost [91], Lee [92], Kuan [93], Gamma Map [94, 95], and the Gamma WMAP [96] (i.e., the Gamma MAP filter applied in the wavelet domain) filters. Note that the requirements on the pre-processing may vary among different change detection methods. For image comparison, several detectors have been proposed. The classical detectors include differencing and ratioing techniques [78], which are carried out through pixel-by-pixel comparison. The image comparison method in [97] is based on analyzing the higher order statistics of SAR images. It is concluded that mean ratio detector was useful for step changes and the second- and third-order log-cumulant were usful for progressive changes appearing in consecutive images. Since higher order statistics seem to be helpful, some recent image comparison methods [34, 54] compare the local probability density functions (pdfs) of the neighborhood of the homologous pixels of the pair of SAR images for change detection. In the field of unsupervised change image thresholding, many approaches [22, 33, 98–101] have been proposed to analyze the similarity image in both the spatial and wavelet domains. Most approaches in the literature concentrate on the third step (thresholding), the focus of this work is on the second step (image comparison), where the objective is to find a good detector to measure the degree of similarly between two image data. The existing image comparison methods will be described in detail in the following subsections.

4.2 Existing Multi-temporal SAR Image Comparison Methods in the Spatial Domain

The existing image comparison methodologies for change detection on in bi-temporal SAR images can be classified into two categories: pixel-based comparison methods, and local statistics comparison methods based on information similarity measures. The pixel-based image comparison techniques usually compute the similarity by performing some specific arithmetic operation on the intensity values of two SAR images, pixel by pixel, while the local statistics image comparison techniques are based on the idea that the images acquired over the same area at two different times are two different measurements of the same information source if that area did not significantly change. The experiment results in [34, 54, 97] concluded that the local-statistics based methods exhibit a better performance in terms of the classification accuracy than the pixel-based methods.

4.2.1 Pixel-Based Image Comparison Techniques

Differencing and ratioing are well-known techniques for change detection. In differencing, changes in radar backscatter are measured by subtracting the intensity values, pixel by pixel, between two dates; while in ratioing, the changes are measured by dividing the intensity values, pixel by pixel. In [78], it was concluded that the ratioing method is superior to the differencing method due to the following reasons. Firstly, in contrast to the difference method, the ratio method only depends on the relative change in average intensity between two dates, and detecting changes in the SAR imagery does not depend on the intensity level of the pixels. Another important reason is that the ratio method is very robust to calibration errors whereas the difference method is not [78]. Both errors are multiplicative factors to the total radar intensity. These factors are eliminated when computing the ratio image because calibration errors are exactly reproduced in repeat-pass imagery. In contrast, in the difference method, radiometric errors directly modulate the difference of the radar intensities, typically yielding higher changes in very hilly terrain (e.g., slopes facing the radar) than in rather flat areas for the same type of surface, which is not a desired effect. The ratioing techniques is also called Mean Ratio Detector (MRD) and is usually computed as the following normalized quantity:

$$D_{MRD} = 1 - \min\{\frac{\mu_X}{\mu_Y}, \frac{\mu_Y}{\mu_X}\}. \tag{4.1}$$

where μ_X and μ_Y stand for the local mean values of the images before and after the event of interest, respectively. The logarithm of Equation 4.1 is conveniently used.

4.2.2 Local Statistics Comparison Techniques Based on Information Similarity Measures

Recently, many change detection algorithms were proposed based on the evolution of the local statistics of the image between two dates. More specifically, the modification of the statistics of each pixel's neighborhood between the two acquisition dates is analyzed, and a pixel is considered as having changed if its statistical distribution changes from one image to the other. In order to quantify the distance between probability densities of the two acquisition dates, a measure, which maps the two estimated statistical distributions (one for each date at a colocated area) into a scalar change index is required. Several information theoretical similarity measures were proposed in the literature including the cluster reward algorithm [102], Woods criterion [102], the correlation ratio [102], mutual information [103], mixed information [104], and the Kullback-Leibler (KL) divergence [34, 54]. Among all these similarity measurement, the KL divergence is widely used as a measure to quantify the change since the KL divergence appears to be an appropriate tool to detect changes when we consider that changes on the ground induce different shapes on the local pdf [54].

Let f_X and g_Y be two probability density functions (pdf) of the random variables X and Y, respectively. The KL-divergence from Y to X, also known as the relative entropy, is given by [105]:

$$KL(f_X \| g_Y) = \int f_X(x) \log \frac{f_X(x)}{g_Y(y)} dx.$$

The measure $\log(f_X(x)/g_Y(y))$ can be thought of as the information contained in x for the discrimination between the hypothesis $\mathscr{H}_X$ and $\mathscr{H}_Y$, if hypothesis $\mathscr{H}_X$ is associated with the pdf $f_X(x)$ and $\mathscr{H}_Y$ with $g_Y(y)$. Therefore, the KL divergence can be understood as the mean information for the discrimination between $\mathscr{H}_X$ and $\mathscr{H}_Y$ per observation. This divergence appears to be an appropriate tool to detect changes when we consider that changes on the ground induce different shapes on the local pdf. It can be easily proved that $KL(f_x \| g_y) \neq KL(g_y \| f_x)$, but a symmetric version called the KL distance (KLD) [54] is wildly used and is defined as:

$$D(f_x, g_y) = D(g_y, f_x) = KL(f_x \| g_y) + KL(g_y \| f_x).$$

In order to estimate the KL distance, the pdfs of the two variables to be compared have to be known. We will introduce below several popular approaches to estimate the pdfs and compute the corresponding KLD for SAR imagery.

- Gaussian KL Detector (GKLD)

In this case, the local statistics of the local random variables X and Y are assumed to be normally distributed and the pdf f_X can be written as

$$f_X(x) = \mathscr{G}(x; \mu_X, \sigma_X) = \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}}. \tag{4.2}$$

An analogous expression holds for g_Y. This Gaussian model yields the GKLD [54]

$$D_{GKLD} = \frac{\sigma_X^4 + \sigma_Y^4 + (\mu_X - \mu_Y)^2(\sigma_X^2 + \sigma_Y^2)}{2\sigma_X^2\sigma_Y^2} - 1. \tag{4.3}$$

It can be seen that, even in the case of identical mean values, the GKLD detector is able to underline the shading of textures, which is linked to the local variance evolution. Nevertheless, SAR-intensity values are not normally distributed, and the use of a bad model can induce bad performance of the detector. However, this example has been given as a simple case of a parametric model, which takes into account second-order statistics. Since some Gaussianity may be introduced into the data when resampling and filtering the images during the preprocessing step, the Gaussian model may nevertheless be justified.

- KLD Using the Pearson System (PKLD)

In [106],the statistics of SAR images are modeled by the family of probability distributes known as the Pearson system [107]. The Pearson system is composed of eight types of distributions, among which the Gaussian and the Gamma distributions may be found. The Pearson system is very easy to use since the type of distribution can be inferred from the following parameters:

$$\beta_{X;1} = \frac{\mu_{X;3}^2}{\mu_{X;2}^3} \qquad \beta_{X;2} = \frac{\mu_{X;4}}{\mu_{X;2}^2}$$

where $\mu_{X;i}$ is the centered moment of the order i of variable X. That means that any distribution from the Pearson system can be assessed from a given set of samples by computing the first four statistical moments. Any distribution, therefore, can be reprosented by a point on the $(\beta_{X;1}, \beta_{X;2})$ plane. For instance, the Gaussian distribution is located at $(\beta_{X;1}, \beta_{X;2}) = (0,3)$, and the Gamma distribution lie on the $\beta_{X;2} = (3/2)\beta_{X;1} + 3$ line. Details about the theory of the Pearson system can be found in [108].

The Person-based KLD (PKLD) dose not have a unique analytic expression since eight different types of distributions may be held [106]. Therefore, 64 different possibilities for the pairs of pdf exist. Once a pair of pdfs is identified, the corresponding KLD can be computed by numerical intergration

$$D_{PKLD} = \int \Bigg[f_X(x;\beta_{X;1},\beta_{X;2}) log \frac{f_X(x;\beta_{X;1},\beta_{X;2})}{f_Y(y;\beta_{Y;1},\beta_{Y;2})} dx \\ + f_Y(y;\beta_{Y;1},\beta_{Y;2}) log \frac{f_Y(y;\beta_{Y;1},\beta_{Y;2})}{f_X(x;\beta_{X;1},\beta_{X;2})} dy \Bigg] . \tag{4.4}$$

The correct way in proceeding to use the Pearson system is to choose a pdf using the estimated moments and then estimate the parameters of the distribution by maximum likelihood [106]. The readers should notice that although this procedure can improve the results of the pdf estimation, the effect is not noticeable in terms of the estimation of the change indicator; and the computation cost is very expensive [54].

- Cumulant-Based KL Approximation (CKLD)

Instead of considering a parameterizations of a given density or set of densities, the authors of [54] proposed the use of the Edgeworth series expansion to describe the local statistics. The basic idea is to describe the shape of the distribution and assume that the quantitative terms of the shape can approximate the pdf itself. More specifically, the cumulants are used to describe the pdf's shape; and then the density is estimated through the Edgeworth series expansion [54]. For the case of the four first-order cumulants, the following expressions hold [54]:

$$\begin{aligned} \kappa_{X;1} &= \mu_{X;1} \\ \kappa_{X;2} &= \mu_{X;2} - \mu_{X;1}^2 \\ \kappa_{X;3} &= \mu_{X;3} - 3\mu_{X;2}\mu_{X;1} + 2\mu_{X;1}^3 \\ \kappa_{X;4} &= \mu_{X;4} - 4\mu_{X;3}\mu_{X;1} - 3\mu_{X;2}^2 + 12\mu_{X;2}\mu_{X;1}^2 - 6\mu_{X;1}^4 . \end{aligned} \tag{4.5}$$

Let us assume that the density to be approximated (denoted as $f_X(x)$) is not too far from a Guassian pdf (the Gaussian pdf is denoted as $\mathscr{G}_X$ to underline the fact that the $\mathscr{G}_X$ has the same mean and variance as $f_X(x)$), that is, with a shape similar to the Gaussian distribution. When choosing a Gaussian law (the density to be approximated is with a shape similar to the Gaussian distribution), the Edgeworth series expansion is obtained. Its expression, when truncated to an order of six, is the following [54]:

$$\begin{aligned} f_X(x) = \Big(1 &+ \frac{\kappa_{X';3}}{6} H_3(x) + \frac{\kappa_{X';4}}{24} H_4(x) + \frac{\kappa_{X';5}}{120} H_5(x) \\ &+ \frac{\kappa_{X';6} + 10\kappa_{X';3}^2}{720} H_3(x)\Big) \mathscr{G}_X(x). \end{aligned} \tag{4.6}$$

where $H_r(x)$ is known as the Chebyshev-Hermite polynomial of order r [109]. This equation can be thought of as a model of the form $X = X_{\mathscr{G}} + X'$ where $X_{\mathscr{G}}$ is a random variable with Gaussian density with the same mean and variance as X, and X' is a standardized version of X [110] with $X' = (X - \kappa_{X;1})\kappa_{X;1}^{-1/2}$.

The approximation of the KL divergence by the Edgeworth series, truncated at the order of four, is given by [111]:

$$\begin{aligned} KL_{Edgeworth}(X,Y) = {}& \frac{1}{12}\frac{\kappa_{X';3}^2}{\kappa_{X;2}^2} + \frac{1}{2}\left(\log\frac{\kappa_{Y;2}}{\kappa_{X;2}} - 1 + \frac{1}{\kappa_{Y;2}}(\kappa_{X;1} - \kappa_{Y;1} + \kappa_{X;2}^{1/2})^2\right) \\ & - \left(\kappa_{Y';3}\frac{a_1}{6} + \kappa_{Y';4}\frac{a_2}{24} + \kappa_{Y';3}^2\frac{a_3}{72}\right) - \frac{1}{2}\frac{\kappa_{Y';3}^2}{36}\left(c_6 - 6\frac{c_4}{\kappa_{X;2}} + 9\frac{c_2}{\kappa_{Y;2}^2}\right) \\ & - 10\frac{\kappa_{X';3}\kappa_{Y';3}(\kappa_{X;1} - \kappa_{Y;1})(\kappa_{X;2} - \kappa_{Y;2})}{\kappa_{Y;2}^6}. \end{aligned} \tag{4.7}$$

where

$$\begin{aligned} & a_1 = c_3 - 3\frac{\alpha}{\kappa_{Y;2}} \qquad a_2 = c_4 - 6\frac{c_2}{\kappa_{Y;2}} + \frac{3}{\kappa_{Y;2}^2} \\ & a_3 = c_6 - 15\frac{c_4}{\kappa_{Y;2}} + 45\frac{c_2}{\kappa_{Y;2}^2} - \frac{15}{\kappa_{Y;2}^3} \\ & c_2 = \alpha^2 + \beta^2; \quad c_3 = \alpha^3 + 3\alpha\beta^2 \\ & c_4 = \alpha^4 + 6\alpha^2\beta^2 + 3\beta^4 \\ & c_6 = \alpha^6 + 15\alpha^4\beta^2 + 45\alpha^2\beta^4 + 15\beta^6 \\ & \alpha = \frac{\kappa_{X;1} - \kappa_{Y;1}}{\kappa_{Y;2}} \qquad \beta = \frac{\kappa_{X;2}^{1/2}}{\kappa_{Y;2}}. \end{aligned} \tag{4.8}$$

Finally, the cumulant-based KLD (CKLD) between two oberservations X and Y is writen as

$$D_{CKLD} = KL_{Edgeworth}(X,Y) + KL_{Edgeworth}(Y,X). \tag{4.9}$$

4.3 Existing Multi-temporal SAR Image Comparison Methods in the Wavelet Domain

In [34], an unsupervised change detection method was proposed in the wavelet domain based on statistical wavelet subband modeling. The motivation is to capture textures efficiently in the wavelet domain. The wavelet transform is applied to decompose the image into multiple scales and the probability density function of the coefficient magnitudes of each subband is assumed to be a Generalized Gaussian Distribution (GGD) or a Generalized Gamma Distribution (GΓD) and is estimated by parameter estimation through fitting. A closed-form expression of the Kullback-Leibler divergence between two corresponding subbands (belonging to the two images being compared) of the same scale is computed and used to generate the change map.

- Generalized Gaussian KL Detector (GGKLD)

It is well-known that the distribution of wavelet subbands can be modeled using the Generalized Gaussian distribution (GGD) [112, 113], which is given as follows:

$$f_X(x,\alpha,\beta) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} \exp\left(-\left|\frac{x}{\alpha}\right|^{\beta}\right). \tag{4.10}$$

where $\alpha > 0$ is a scale parameter, $\beta > 0$ is a shape parameter, and Γ detnotes the Gamma function. Compared to the Gaussian distribution, the GGD distribution model is more flexible and general while the estimation of its parameters is more complex and difficult. A common method for parameter estimation is maximum likelihood estimation [113]. Recently, an alternative and fast method for estimating the shape parameter based on the convex shape equation was proposed in [114]. Fortunately, an analytic expression of the KL divergence of the GGD is available for the fast evaluation of the the similarity between two corresponding windows from the two images being compared. The closed-form symmetric KL divergence between two GGDs ($f_X(x:\alpha_1,\beta_1)$ and $f_Y(y:\alpha_2,\beta_2)$) is given as follows [34]:

$$D_{GGKLD}(f_X \| f_Y) = \log\left(\frac{\beta_1\alpha_2\Gamma\left(\frac{1}{\beta_2}\right)}{\beta_2\alpha_1\Gamma\left(\frac{1}{\beta_1}\right)}\right) + \left(\frac{\alpha_1}{\alpha_2}\right)^{\beta_2} \frac{\Gamma\left(\frac{\beta_2+1}{\beta_2}\right)}{\Gamma\left(\frac{1}{\beta_1}\right)} - \frac{1}{\beta_1}. \tag{4.11}$$

To exploit the multiscale property of the wavelet transform, the authors of [34] assume that the subbands are independent such that the total similarity of two blocks are defined as the sum of similarity measures of each subband. The final similarity map $SMAP_{GGKLD}$ is give by:

$$SMAP_{GGKLD} = \sum_{i=1}^{3S} D_{GGKLD}(f_{X;i} \| f_{Y;i}) \tag{4.12}$$

where S is the number of scales; $f_{X;i}$ and $f_{Y;i}$ are the estimated distributions of the wavelet coefficients in subband i.

- Generalized Gamma KL Detector (GΓKLD)

Recently, it has been observed that the GGD can not characterize fully the coefficients' distribution of complex scenes with high variations [115–117]. A promising distribution model, called the Generalized Gamma distribution (GΓD), has been proposed for modeling the distribution of the wavelet coefficients. The one-side GΓD version ($x > 0$) is defined as

$$f_X(x, \alpha, \beta) = \frac{\beta x^{\beta\lambda - 1}}{\alpha^{\beta\lambda}\Gamma(\lambda)} \exp\left(-\left(\frac{x}{\alpha}\right)^{\beta}\right) \qquad x > 0. \tag{4.13}$$

where α is the scale parameter, β is the shape parameter and λ is the index shape parameter. The two-side GΓD is given as $f'_X(x) = f_X(|x|)/2$ $(x \in R)$. The advantage of GΓD over GGD is that it introduces another index shape parameter such that the model is more general and flexible. Usually, the parameters of a one-sided GΓD are estimated first. The one-sided GΓD can be easily extended to a two-sided version by the symmetric property. Two promising estimation methods have been discussed in [34]. One is the Method of Log-Cumulants (MoLC) [118] based on second kind statistics, which was demonstrated to be efficient for the parameter estimation of a positive random variable. The other method is based on scale-independent shape estimation (SISE) and allows the efficient estimation of the parameters [116], and for which the uniqueness and consistency of the solution were proved mathematically [116].

The close-form expression of the KL divergence between two GΓDs is available. The closed-form symmetric KL divergence between two GΓDs ($f_X(x:\ \alpha_1,\ \beta_1,\ \lambda_1)$ and $f_Y(y:\ \alpha_2,\ \beta_2,\ \lambda_2)$) is given as follows [34]:

$$\begin{aligned} D_{G\Gamma KLD}(f_X \| f_Y) = & -\lambda_1 - \lambda_2 + (\beta_1\lambda_1 - \beta_2\lambda_2)\left(\log\frac{\alpha_1}{\alpha_2} + \frac{\Psi(0,\lambda_1)}{\beta_1} - \frac{\Psi(0,\lambda_2)}{\beta_2}\right) \\ & \left(\frac{\alpha_1}{\alpha_2}\right)^{\beta_2}\frac{\Gamma(\lambda_1+\beta_2/\beta_1)}{\Gamma\lambda_1} + \left(\frac{\alpha_2}{\alpha_1}\right)^{\beta_1}\frac{\Gamma(\lambda_2+\beta_1/\beta_2)}{\Gamma\lambda_1}. \end{aligned} \tag{4.14}$$

where $\Gamma()$ is the Gamma function and $\Psi()$ is the polygamma function.

Similar to $SMAP_{GGKLD}$, the final similarity map $SMAP_{G\Gamma KLD}$ is obtained by summing all similarity measures across subbands as follows:

$$SMAP_{G\Gamma KLD} = \sum_{i=1}^{3S} D_{G\Gamma KLD}(f_{X;i} \| f_{Y;i}) \tag{4.15}$$

Chapter 5

CHANGE DETECTION ON SAR IMAGES USING LOCAL STATISTICS AND GAUSSIAN MIXTURE MODELS

5.1 Introduction

For image comparison, several change detectors have been proposed. The classical detectors include differencing and ratioing techniques [78], which are carried out through pixel-by-pixel comparison. Compared to the difference operator, the ratio operator is more robust to illumination variations, speckle noise and calibration errors. However, the ratio operator, also known as the mean ratio detector [119], assumes that the texture is a zero-mean multiplicative contribution. As a result, it cannot detect changes taking place at the texture level.

In recent years, promising methods based on information theory have been developed for change detection of multi-temporal SAR images, where the basic idea is that the images acquired over the same area at two different times are two different measurements of the same information source. Several information similarity measures were proposed to assess the similarity of the two SAR images of the same source. Generally, most of those measures utilize the local probability density functions (PDFs) of the neighborhood of pixels to perform the comparison, instead of a pixel-by-pixel comparison. Consequently, the quality of the resulting difference image depends on how well the statistical models fit the local statistics of the images. In [91], the Gaussian model was used to approximate the local image statistics. However, SAR-intensity statistics are not typically normally distributed. In [54], the Pearson system, which is composed of eight types of distributions, and one-dimensional Edgeworth series expansion techniques were proposed to model the local distributions in the neighborhood of each pixel. Then, the Kullback-Leibler divergence was used to measure the degree of evolution of the local statistics. However, the computational complexity of these two methods proposed in [54] is very high since they have to use fourth order statistics to estimate the parameters of the local PDF. The method proposed in [54] was extended to object-based change detection by computing the Kullback-Leibler divergence of two corresponding objects resulting from image segmentation [120]. The method proposed in [34] extends the information similarity measure-based method to the wavelet domain by using generalized Gaussian and Gamma distributions to model the magnitude of the subband coefficients, and the Kullback-Leibler divergence is used to compute the similarity between the statistical models corresponding to the same subbands at the same scale.

However, all these image comparison methodologies were developed under the assumption that the local neighborhood of each pixel corresponds to a land-cover typology whose local statistics can be approximated using a single parametric mathematical distribution model. However, the local neighborhoods typically contain different regions with different characteristics, especially for lager window sizes. Typically, the window size ranges from 3×3 pixels to 51×51 pixels [54]. In addition, a SAR image generally contains a scene with several distinct land-cover typologies. Therefore, a locally adaptive model is needed to approximate the distribution of each window block. Gaussian Mixture Models (GMM) have been widely used to model natural image statistics [121] since they can approximate a variety of distributions, and only second-order statistical parameters are needed to be estimated to obtain the GMM.

In this work, the GMM is used to adaptively approximate the local statistical distribution of the SAR image both in the spatial and wavelet domains. The proposed adaptive GMM estimation method is capable of selecting the optimal number of components and parameters of each Gaussian component according to the characteristics of each local distribution. The degree of similarity between the local statistics of two SAR images is measured using the Kullback-Leibler (KL) divergence. Results from experiments that are conducted with real SAR data are shown to illustrate that the proposed change detection method outperforms other change detection algorithms both in the spatial and wavelet domains. One analytical expression for approximating the KL divergence between GMMs is given and is compared with the Monte Carlo sampling method.

This chapter is organized as follows. Section 5.2 begins with a review of existing statistical models for SAR images and then presents the proposed GMM model and its adaptive parameter estimation method to approximate the local statistics of SAR images. The proposed GMM model is also compared with two popular statistical distributions. Section 5.3 presents the proposed spatial-domain and wavelet-domain change detection methods based on the KL-divergence of the GMM. Section 5.4 describes the data sets used for evaluations and the experimental setup, and shows the experimental results. Finally, conclusions and remarks are given in Section 5.5.

5.2 Statistical Modeling for SAR Image

This section begins with the review of existing popular statistical distributions that are used for modeling the statistics of SAR images. Then, the proposed Gaussian mixture model and corresponding adaptive parameter estimation method are presented. Finally, the proposed GMM model is compared with existing statistical distributions in order to evaluate their ability to approximate the PDF of actual SAR images.

5.2.1 Problem Formulation and Related Work

The precise modeling of local statistics is crucial in the context of SAR image processing and applications, especially for change detection. Many techniques have been developed to characterize the statistics of SAR images. Both non-parametric and parametric approaches can be employed for this task [122]. Non-parametric estimation approaches do not assume any specific analytical model for the unknown PDF but directly estimate the PDF using training data. These include, for example, the Parzen window estimator [123], artificial neural networks [124], or support vector machines [125]. Although non-parametric approaches can achieve more flexibility and adaptability, they require a large amount of training samples and usually present internal architecture parameters to be set by the user [126].

In contrast, the parametric estimation approaches postulate a mathematical model for the statistical modeling of SAR images and they do not require a large number of training data. Therefore, the parametric statistical methods have received more attention. Many mathematical distribution models have been proposed to characterize SAR amplitude images or intensity images. These parametric estimation approaches are generally based on three SAR image formation models, namely, the multiplicative, empirical and SAR scattering model.

The multiplicative model states that the backscattered return is the product of the speckle noise and the terrain backscatter. More specifically, the multiplicative model combines an underlying radar cross-section (RCS) component σ with an uncorrelated multiplicative speckle component n_{noise}. Thus, in this model, the observed intensity I in a SAR image is expressed as the product $I = \sigma \cdot n_{noise}$. The $\mathscr{K}$ distribution in [127] and $\mathscr{G}$ distribution in [128] are such examples. Their difference lies in the adopted distribution of RCS. For the $\mathscr{K}$ distribution case, the Gamma distribution is used to approximate the distribution of RCS, while the generalized inverse Gaussian law is used for the $\mathscr{G}$ distribution case.

In the empirical model case, the corresponding models are obtained through experimental analysis of actual SAR images rather than SAR physical image formation mechanisms [129]. Examples reported in the literature include the log-normal [130], Weibull [130], and Fisher [131] distributions. Recently, more flexible and generalized models were proposed to model the SAR images with different characteristics. In [54], the Pearson system and one-dimensional Edgeworth series expansion were proposed to model the local statistics of SAR images. The generalized Gamma distribution (GΓD) was proposed in [132] for the empirical modeling of SAR images. The GΓD supports a large variety of distributions, including Weibull, log-normal, Rayleigh, exponential, Nakagami, and Gamma distributions. The experimental results in [132] demonstrate that the GΓD can achieve a better goodness of fit than existing state-of-the-art PDFs (e.g., Fisher and $\mathscr{K}$ distributions).

The SAR scattering model is based on the following assumptions in relation to the SAR image formation process: 1) the scatterers are statistically independent; 2) the number of scatterers is large; 3) the scattering amplitude and the instantaneous phase are independent random variables; 4) the phase is uniformly distributed over the range [0, 2π]; 5) no individual scatterer dominates the whole scene; and 6) the reflection surface is large when compared to the size of individual reflections. The SAR scattering model describes the complex signal of backscattered fields for a given ground area under illumination by a single-look SAR sensor [122]. Let Z refers to the complex signal received by the SAR sensor from the ground area corresponding to a given pixel. Z is given by:

$$\begin{aligned} Z =&X + jY = r\exp\left(j\theta\right) = \sqrt{I}\exp\left(j\theta\right) \\ &= \sum_{i=1}^{N_{waves}} r_i e^{j\theta_i} = \sum_{i=1}^{N_{waves}} \left(r_i\cos(\theta_i) + jr_i\sin(\theta_i)\right) \end{aligned} \tag{5.1}$$

where X, Y, r, I, and θ are the real part, imaginary part, amplitude, intensity, and phase of the complex SAR backscattered signal Z, respectively. In addition, r_i and θ_i are, respectively, the amplitude and phase of the ith scattered complex wave. N_{waves} is the number of reflected waves at the receiver. The problem here is to achieve parametric statistical modeling of the amplitude SAR image (r) from the PDFs of X and Y. Several well-known statistical distributions for the SAR amplitude image (r) were derived based on distributions for the real part X and imaginary part Y. In [133], invoking the central limit theorem and under the assumptions described above, the real and imaginary parts (X and Y) of the complex backscattered signal were modeled as i.i.d. Gaussian random variables. In this case, the signal amplitude Z can be approximated by the Rayleigh PDF and the intensity I can be modeled by the exponential distribution. Based on a generalized version of the central limit theorem, which states that the sum of a large number of i.i.d. processes approach the α-stable law, the authors of [134] adopt the zero-mean symmetric α-stable distribution for the real and imaginary parts of the complex backscattered signal, and they proposed the generalized heavy-tailed Rayleigh amplitude distribution (denoted by SαSGR). The authors of [122] extended the Rayleigh-distributed amplitude model by assuming that the real and imaginary parts of the complex backscattered signal are distributed following a generalized Gaussian PDF. Based on this assumption, they showed that the statistics of the amplitude follow the generalized Gaussian Rayleigh (GGR) distribution [122]. More recently, by assuming a two-sided generalized Gamma distribution for the real and imaginary parts of the complex SAR backscattered signal, the generalized Gamma Rayleigh distribution (GΓR) was proposed in [129] to model the amplitude SAR images. The experimental results in [129] showed that the GΓR distribution is a very efficient and flexible model since the GΓR distribution is the general form of many other models, including the heavy-tailed Rayleigh distribution and the GGR model.

The GΓD and GΓR based models were found to outperform the majority of other aforementioned models which correspond to special cases of the GΓD or GΓR models. Therefore, in this paper, these two models are adopted as the competitive models for comparison in order to evaluate the performance of the proposed model. In addition, several strategies were proposed to estimate the parameters of GΓD and GΓR models. The three most frequently used methods include the maximum-likelihood (ML) method [135, 136], the method of moments (MoM) [137], and the method of log-cumulants (MoLC) [129, 132]. Table 5.1 shows the analytic expression of the PDFs and the MoLC nonlinear equations for the parametric estimation corresponding to the GΓD and GΓR models.

Table 5.1: PDFs And MoLC Equations for GΓD And GΓR Models. Here $k_r(\cdot)$ Is the rth-order Modified Bessel Function of the Second Kind; $\psi(\cdot)$ Is the Digamma Function; And $\psi(r,\cdot)$ Is the rth-order Polygamma Function [138]. $G_r(\cdot,\cdot)$ Is the Specific Integral Functions for GΓR [129].

Model	Distribution Function	MoLC Equation
GΓD	$p(r)=\frac{\lvert\nu\rvert\kappa^{\kappa}}{\sigma\Gamma(\kappa)}(\frac{r}{\sigma})^{\kappa\nu-1}exp\{-\kappa(\frac{r}{\sigma})^{\nu}\}$	$k_1=\log(\sigma)+\frac{\psi(\kappa)-\log(\kappa)}{\nu}$ $k_i=\frac{\psi(i-1,\kappa)}{\nu_i}, i=2,3$
GΓR	$p(r)=[\frac{\upsilon}{\eta^{\kappa\nu}\Gamma(\kappa)}]^2 r^{2\kappa\upsilon-1}\int\lvert\cos\theta\sin\theta\rvert^{\kappa\nu-1}$ $\cdot exp\{-[\frac{r}{\eta}]^{\nu}(\lvert\cos\theta\rvert^{\nu}+\lvert\sin\theta\rvert^{\nu})\}d\theta,$ with $\upsilon=1/\varpi$	$k_1=\log\eta+\varpi\psi(2\kappa)-\varpi\frac{G_1(\kappa,\omega)}{G_0(\kappa,\omega)}$ $k_2=\varpi^2[\psi(1,2\kappa)+\frac{G_2(\kappa,\omega)}{G_0(\kappa,\omega)}-\frac{G_1^2(\kappa,\omega)}{G_0^2(\kappa,\omega)}]$ $k_3=\varpi^3[\psi(2,2\kappa)-\frac{G_3(\kappa,\omega)}{G_0(\kappa,\omega)}$ $+3\frac{G_2(\kappa,\omega)G_1(\kappa,\omega)}{G_0^2(\kappa,\omega)}-2\frac{G_1^3(\kappa,\omega)}{G_0^3(\kappa,\omega)}]$

5.2.2 Proposed Gaussian Mixture Model(GMM) for PDF Estimation of SAR Image in the Spatial and Wavelet Domains

Note that all of the aforementioned distributions are mainly suitable for SAR images containing homogenous regions, even though each homogenous region shows different land-cover typologies. However, in general, actual SAR images correspond to a varied scene presenting several different regions with different characteristics. In this paper, the Gaussian mixture model is proposed to address this problem by postulating that the local neighborhood around each pixel in the SAR image can be modeled as a combination of parametric Gaussian components, each one corresponding to a specific land-cover typology. The PDF of a SAR image patch is therefore viewed as an instance of a generative mixture of Gaussian model. Consequently, at each pixel, we adaptively model the surrounding local neighborhood, consisting of N pixels p_j, $j=1\dots N$, according to a Gaussian mixture model (GMM). The GMM is a weighted sum of K component Gaussian distributions as given below:

$$p(R)=\sum_{i=1}^{K} w_i\mathcal{N}(R;\mu_i,\Sigma_i), \tag{5.2}$$

where R is the measurement or feature vector for the considered local neighborhood, $w_i, i=1\dots K$, are called mixing coefficients which must fulfill $\sum_{i=1}^{K} w_i=1$, and $\mathcal{N}(R;\mu_i,\Sigma_i)$ is a Gaussian density with mean vector μ_i and covariance matrix Σ_i. As a result, a GMM is parameterized by $\{K, w_i, \mu_i, \Sigma_i\}$. Estimating the GMM parameters requires: 1) optimizing the number K of mixture components; and 2) estimating the parameters of each Gaussian component.

Several methods have been proposed to estimate K, the number of components of a mixture [139–141]. Most of these methods are model-class selection criteria: they select a model-class based on its ”best” representative. The main idea is to first obtain a set of candidate models for a range of values of K, e.g., from K_{min} to K_{max}. Then, find the optimal $\hat{K}$ according to

$$\hat{K} = \underset{K}{\operatorname{argmin}}\{\mathscr{C}(\hat{\theta}(K),K), K = K_{min},...,K_{max}\}, \tag{5.3}$$

where θ is the set of parameters needed to be estimated. For Gaussian model, $\theta = \{\mu, \Sigma\}$. $\mathscr{C}(\hat{\theta}(K),K)$ is the cost function and it has the form

$$\mathscr{C}(\hat{\theta}(K),K) = -\log p(R|\hat{\theta}(K)) + \mathscr{P}(K), \tag{5.4}$$

where R is a set of measurements or feature vectors and $\mathscr{P}(K)$ is an increasing function with K. Traditionally, the expectation maximization (EM) algorithm is used to obtain the candidate models for certain number of components (fixed K). However, in mixture models, the distinction between model-class selection and model estimation is unclear, e.g. a 3-component mixture in which one of the mixing probabilities is zero is undistinguishable form a 2-component mixture. Also, the traditional EM algorithm surfers from two main drawbacks: it is sensitive to initialization and it may converge to the boundary of the parameter space [142, 143].

To address these problems, the authors in [144] proposed an unsupervised method for learning a finite mixture model, which seamlessly integrates estimation and model selection in a single algorithm and utilized a variant of EM algorithm (component-wise EM for mixture algorithm (CEM^2) [145]) to directly implement the the minimum message length (MML) criterion. More specifically, Let k be some arbitrary large value and infer the structure of the mixture probabilities by letting the estimatesof some of the mixing probabilism be zero. This approach coincides with the MML philosophy, which dose not adopt the ”model-class/model” hierarchy, but directly aims at finding the ”best” overall model in the entire set of available models. This MML based CEM^2 algorithm is less sensitive to initialization and avoid the boundary of the parameters space.

The estimation function of the MML based CEM^2 algorithm is shown as follow [144]

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\mathscr{L}(\theta,R), \tag{5.5}$$

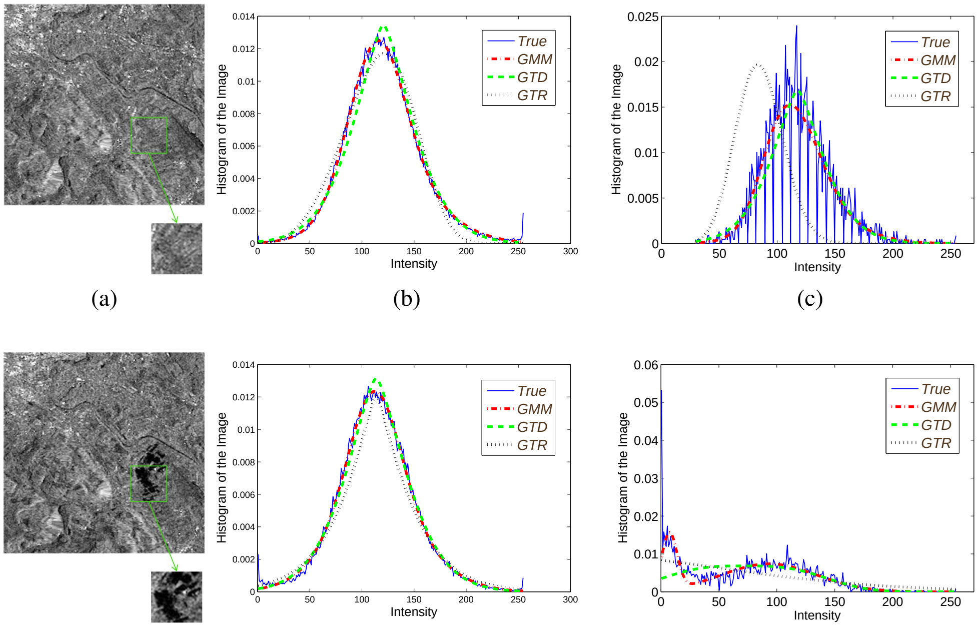
Algorithm 1 Minimum Message Length (MML) Algorithm [144]

```
Inputs: $K_{max}$, initial parameters $\hat{\theta}(0) = \{\hat{\theta}_1, ..., \hat{\theta}_{K_{max}}, \hat{w}_1, ..., \hat{w}_{K_{max}}\}$
Outputs: Optimal parameter $\hat{\theta}_{best}$
Set $t = 0$, $K_{nz} = K_{max}$, $\mathscr{L}_{min} = +\infty$, $u_m^{(i)} = \phi(r^{(i)}|\hat{\theta}_m)$,for $m = 1, ..., K_{max}$, and $i = 1, ..., N$
while $K_{nz} \geq 1$ do
  $t = t + 1$
  for $m = 1 : K_{max}$ do
    $\phi_m^{(i)} = \hat{w}_m u_m^{(i)} (\sum_{i=1}^{K_{max}} \hat{w}_j u_j^{(i)})^{-1}$,for $i = 1, ..., N$
    $\hat{w}_m = max\{0, (\sum_{i=1}^{n} \phi_m^{(i)}) - \frac{H}{2}\}(\sum_{j=1}^{K} max\{0, (\sum_{i=1}^{n} \phi_j^{(i)}) - \frac{H}{2}\})^{-1}$
    $\{\hat{w}_1, ..., \hat{w}_{K_{max}}\} = \{\hat{w}_1, ..., \hat{w}_{K_{max}}\}(\sum_{m=1}^{K_{max}} \hat{w}_m)^{-1}$
    if $\hat{w} > 0$ then
      $\hat{\theta}_m = \underset{\theta_m}{\text{argmax}}\ log p(r, \mathscr{W}|\theta)$
      $u_m^{(i)} = p(r^{(i)}|\hat{\theta}_m)$, for $i = 1, ..., N$
    else
      $K_{nz} = K_{nz} - 1$
    end if
  end for
  $\hat{\theta}(t) = \{\hat{\theta}_1, ..., \hat{\theta}_{K_{max}}, \hat{w}_1, ..., \hat{w}_{K_{max}}\}$,
  $\mathscr{L}[\hat{\theta}(t), r] = \frac{H}{2} \sum_{m:w_m>0} log \frac{n\hat{w}_m}{12} + \frac{K_{nz}}{2} log \frac{n}{12} + \frac{K_{nz}H + K_{nz}}{2} - \sum_{i=1}^{n} log \sum_{m=1}^{K} \hat{w}_m u_m^{(i)}$
  until $\mathscr{L}[\hat{\theta}(t-1), r] - \mathscr{L}[\hat{\theta}(t), r] < \varepsilon|\mathscr{L}[\hat{\theta}(t-1), r]|$
  if $\mathscr{L}[\hat{\theta}(t), r] \leq \mathscr{L}_{min}$ then
    $\mathscr{L}_{min} = \mathscr{L}[\hat{\theta}(t), r]$
    $\hat{\theta}_{best} = \hat{\theta}(t)$
  end if
  $m^* = argmin_m\{\hat{w}_m > 0\}, \hat{w}_{m^*} = 0, K_{nz} = K_{nz} - 1$
end while
```

where θ is the set of parameters that need to be estimated and $\mathscr{L}(\theta, R)$ is a cost function given by:

$$\begin{aligned}\mathscr{L}(\theta, R) =& \frac{H}{2} \sum_{m:w_m>0} \log\left(\frac{Nw_m}{12}\right) + \frac{K_{nz}}{2} \log\frac{N}{12} \\ &+ \frac{K_{nz}(H+1)}{2} - \log p(R|\theta).\end{aligned} \tag{5.6}$$

In (5.6), H is the number of parameters in each component, (i.e., for Gaussian components $H = 2$), N is the number of independent and identically distributed (iid) samples, w_m is the probability of the mth component, and K_{nz} denote the number of non-zero-probability components. Equation (5.6) is the final cost function under the MML criterion, whose minimization with respect to θ will constitute mixture estimate. This equation has the following intuitively interpretation: 1) as usual, $-\log p(R|\theta)$ is the code-length of the data; 2) the expected number of data points generated by the mth component of the mixture is Nw_m; this can be seen as an effective sample size from which w_m



(a) (b) (c)

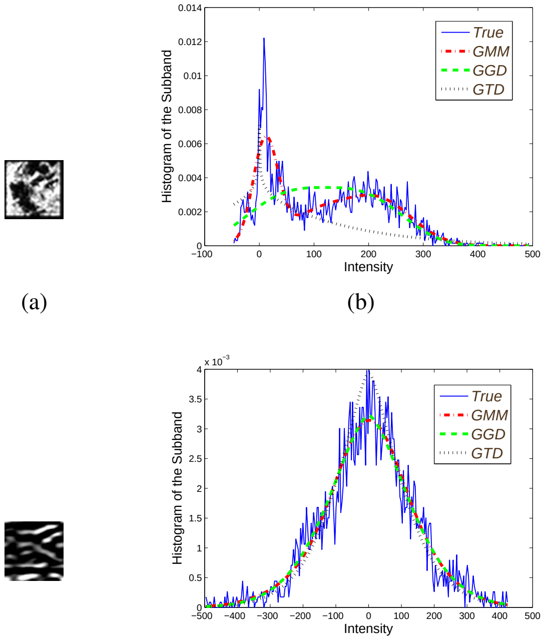
(d) (e) (f)

Figure 5.1: Approximation of the Histograms of SAR Images And of Local Extracted 51 × 51 Windows Using the GΓD (Green, Dashed), GΓR (Purple, Dotted) And Proposed GMM (Red, Dash-Dotted) Models:(a) Original SAR Image And Extracted Window before Abrupt Change; (b) PDF Fitting of the Histogram of the SAR Image (a); (c) PDF Fitting of the Histogram of the Extracted Window from SAR Image (a); (d) Original SAR Image And Extracted Window after Abrupt Change; (e) PDF Fitting of the Histogram of the SAR Image (d); (f) PDF Fitting of the Histogram of the Extracted Window from SAR Image (d).

is estimated; this the "optimal" (in the minimum description length sense) code length for each θ_m is $(H/2)\log(Nw_m)$; 3) the w_ms are estimated from all the N observations, giving rise to the $\frac{K_{nz}}{2} log \frac{N}{12}$ term. Rather than using EM algorithm, the recent component-wise EM for mixtures (CEM^2) [145] algorithm was employed to minimize the cost function in Equation (5.6) in [144].

The implementation of Equations (5.5 and 5.6) can be achieved using Algorithm (1) [144]. Here $\phi_m^{(i)}$ is the a posteriori probability and $\mathscr{W}$ is the conditional expectation. From Algorithm (1), we can see that it is an iterative algorithm. The computation is repeated until $K_{nz} = 1$. After several iterations, if the relative decrease in $\mathscr{L}(\hat{\theta}(t), R)$ falls below a threshold ε (e.g., $\varepsilon = 10^{-5}$), the algorithm converges. In the end, we choose the number of components $\hat{K}$ and the estimated parameters $\hat{\theta}$ which led to the minimum value of $\mathscr{L}(\theta, R)$.

In this work, we propose the use of the GMM to model the SAR image in a locally adaptive



(a) (b)

(c) (d)

Figure 5.2: Approximation of the Histograms of Subband Coefficients of Extracted SAR Window Image (Shown in Fig. 5.1 (d)) with the GGD (Green, Dashed), GΓD (Purple, Dotted) And Proposed GMM (Red, Dash-Dotted) Models: (a)LL Subband at Level 1; (b) PDF Fitting of the Histogram of Subband Coefficients (a); (c)LH Subband at Level 3; (d) PDF Fitting of the Histogram of Subband Coefficients (c).

manner, and adopt the unsupervised learning algorithm of [144] to estimate the number of mixture components and the parameters of each Gaussian component. In Fig. 5.1, we compare the fitting performance of the GΓD, GΓR and the proposed GGM models in approximating the local pdfs of SAR images. The top three figures (Figs. 5.1 (a), (b), (c)), respectively, show the SAR image as well as the extracted window, the fitting results for the image and the fitting results for the extracted window before abrupt change. In this case, although the GΓD and GΓR approximations give fairly good fitting performance, the proposed GMM approximation fits the data much better. The bottom three figures (Figs. 5.1 (d), (e), (f)) show the results for the SAR image and the extracted window after abrupt change. In this case, there are more than one peak in the histogram, and the GΓD and GΓR approximations fail to fit the data. However, the proposed GMM approximation still provides a good fitting performance as shown in Figure 5.1 (f). A similar conclusion is obtained for other SAR images and extracted windows.

Since textures can be easily represented and discriminated in the wavelet domain [34], we also investigate the suitability of the GMM model to approximate the local distribution of wavelet coefficients of SAR images. The generalized Gaussian distribution (GGD) and generalized Gamma distribution (GΓD) were used previously to model the distributions of the subbands coefficient magnitude [34]. In this paper, these two models are adopted as the competitive models for comparison when evaluating the performance of the proposed GMM model in the wavelet domain. Fig. 5.2 shows the fitting performance of the GGD, GΓD, and proposed GGM models in approximating the distributions of wavelet coefficients at different levels. From Figs. 5.2 (c) and (d), it can be clearly seen that the GMM model is superior to the GGD and GΓD models.

5.3 Proposed Change Detection Algorithm Applied to the Spatial and Wavelet Domains

Let us consider two co-registered SAR intensity images I_X and I_Y acquired over the same geographical area at two different times t_X and t_Y, respectively. Our aim is to generate a change detection map that represents changes that occurred on the ground between the acquisition dates. This change detection problem can be modeled as a binary classification problem where 1 represents changed pixels and 0 represents unchanged pixels.

The proposed change detection algorithm analyzes the difference of the local statistics of each pixel's neighborhood between the two acquired image data in both the spatial and wavelet domains. A pixel will be considered as a changed pixel if its local statistical distribution changes from one acquired image to the other. In order to quantify this change, the Kullback-Leiber (KL) divergence [105] between two probability density functions is used. In this paper, as discussed before, the local PDFs are approximated by the Gaussian Mixture Model. More details about the proposed algorithm are given below.

5.3.1 Change Detection in the Spatial Domain

The block diagram of the proposed change detection method in the spatial domain is shown in Fig. 5.3. For two pixels at the same location in the two acquired images, the local statistics of their respective neighborhoods are modeled using the GMM distribution. Then, for each location, the KL divergence is computed using the estimated GMM distributions. This process is performed for each pair of pixels from the two acquired image resulting in a similarity map. The final binary

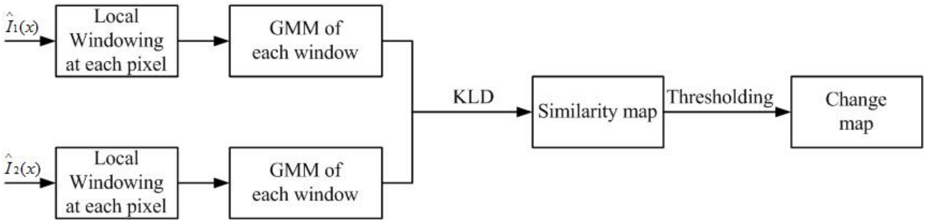


Figure 5.3: Block Diagram of the Proposed Spatial-Domain Change Detection Method.

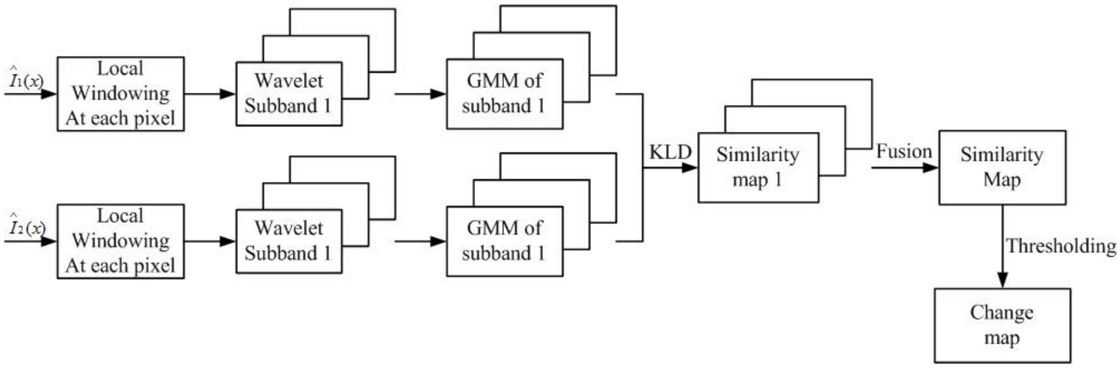


Figure 5.4: Block Diagram of the Proposed Wavelet-Domain Change Detection Method.

change detection map is obtained by thresholding the similarity map.

5.3.2 Change Detection in the Wavelet Domain

Since textures can be easily represented and discriminated in the wavelet domain, the proposed change detection algorithm is also extended to the wavelet domain. The framework of the proposed method in the wavelet domain is shown in Fig. 5.4. The first step is to decompose a sliding window at each pixel into multiple subbands using the wavelet transform. The second step is to estimate, for each subband, the GMM parameters governing the distribution of subband coefficients. Then, the KL divergence between the estimated GMM PDFs of two subbands, one from each of the two acquired images, at the same level and orientation is computed; thus the similarity map for each pair of subbands at the same level and orientation is obtained. Finally, all the subband-specific similarity maps are combined to obtain a final change map $SMAP(n_1, n_2)$ by summing the similarity maps over all subbands as follows:

$$SMAP(n_1,n_2) = \sum_{i=1}^{S}\sum_{j=1}^{M} D(p_{x;i,j} \| p_{y;i,j}) \tag{5.7}$$

where (n_1, n_2) is the location of the pixel where the sliding window is centered, S and M are the number of the scales and orientations, respectively, and $p_{x;i,j}$ and $p_{y;i,j}$ are the estimated distributions of wavelet coefficients at scale i and orientation j for the considered local windows being compared. D is the symmetric KL distance and will be discussed in the following section. The final binary change detection map is obtained by thresholding the similarity map $SMAP(n_1, n_2)$.

5.3.3 Kullback-Leibler Divergence of GMM

As described in the previous section, the GMM is used to characterize the local statistics of SAR images and thus the normalized histogram of SAR images can be represented as follows: $f_x(x) = \sum_i \alpha_i \mathscr{N}(x; \mu_i, \Sigma_i)$; $g_y(y) = \sum_j \beta_j \mathscr{N}(y; \mu_j, \Sigma_j)$, where $f_x(x)$ is the normalized histogram before abrupt change and $g_y(y)$ is the normalized histogram after abrupt change. As a result, our objective is to find the similarity of these two GMM densities. For two Gaussian densities $\hat{f}_x$ and $\hat{g}_y$, the KL divergence has a closed form expression [146] given by:

$$\begin{aligned} KL(f_x \| g_y) &= \frac{1}{2}\left[\log\frac{|\Sigma_{\hat{g}}|}{|\Sigma_{\hat{f}}|} + Tr[\Sigma_{\hat{g}}^{-1}\Sigma_{\hat{f}}]\right. \\ &\quad \left. + (\mu_{\hat{f}} - \mu_{\hat{g}})^T \Sigma_{\hat{g}}^{-1} (\mu_{\hat{f}} - \mu_{\hat{g}})\right] \end{aligned} \tag{5.8}$$

However, there is no closed form expression for the KL divergence between two GMMs. Monte Carlo simulation can estimate the KL divergence with arbitrary accuracy [146]. Using Monte Carlo simulation, the KL divergence of two GMM distributions f and g can be approximated as

$$KL_{MC}(f_x \| g_y) = \frac{1}{N}\sum_{i=1}^{N} \log f_x(x_i)/g_y(x_i), \tag{5.9}$$

$$KL_{MC}(g_y \| f_x) = \frac{1}{N}\sum_{i=1}^{N} \log g_y(y_i)/f_x(y_i), \tag{5.10}$$

where $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ are i.i.d samples drawn from the GMM $f_x(x)$ and $g_y(y)$, respectively [146]. As the number of samples $N \to \infty$, the KL divergence obtained by Monte Carlo simulation tends to the true KL divergence. The Monte Carlo KL distance is then given by

$$D_{MC}(f_x, g_y) = KL_{MC}(f_x \| g_y) + KL_{MC}(g_y \| f_x). \tag{5.11}$$

The Monte Carlo (MC) method is a convergent method. However, the number of samples required for high accuracy approximation is very large which may cause a significant increase in

Table 5.2: Comparison of KL Approximation Methods.

Algorithms	MC(100)	MC(1K)	MC(10K)	MKL
RMSE	0.21	0.13	0.06	0.25

computational complexity. The matching based approximation [121] can be used to reduce the computational complexity. The matching KL divergence (MKL) approximation algorithm can be briefly described as follows:

1. Components matching: Define the matching function π which matches each components of $f_x(x)$ to the components of $g_y(x)$ as:

   $$\pi(i) = argmin_j \left(KL(f_i \| g_j) - \log \beta_j\right). \tag{5.12}$$

   where $KL(f_i \| g_j)$ is the closed form KL-divergence between the ith Gaussian component of f_x and jth Gaussian component of g_y which can be calculated using (5.8). In (5.12), β_j is the weight of the jth component of g_y.

2. GMM KL Approximation: Based on π, approximate the KL divergence between two GMMs f_x and g_y as:

   $$KL_{mat}(f_x \| g_y) = \sum_i \alpha_i \left(KL(f_i \| g_{\pi(i)}) + \log \frac{\alpha_i}{\beta_{\pi(i)}}\right). \tag{5.13}$$

As a result, the closed form expression of the KL distance for the matching based method is given by

$$D_{mat}(f_x, g_y) = KL_{mat}(f_x \| g_y) + KL_{mat}(g_y \| f_x). \tag{5.14}$$

In our experiments, both the MC and MKL methods are applied to approximate the KL divergence between two SAR subimages. In order to compare the approximation performance, we use the Monte Carlo method with one million samples as the "ground truth". Table 5.2 shows the approximation results in terms of root mean square error (RMSE) which can be calculated as: $RMSE = \sqrt{\sum_{n=1}^{N_{total}} (\hat{KL} - KL_{true})^2}$, where N_{total} is the total number of pixels in the SAR image.

From Table 5.2, it can be seen that, as the number of samples increases, the approximation performance of MC improves. However, this will also increase the computational complexity. The MKL approximation can reduce the computation complexity with minor performance decrease.

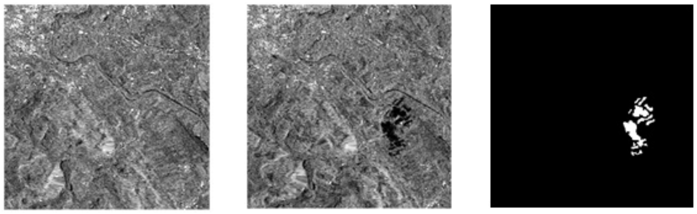
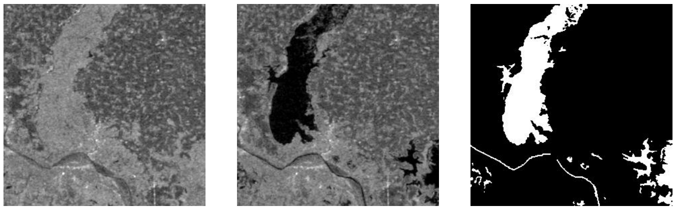


Figure 5.5: Multi-Temporal SAR Images of Switzerland: (a) Image Acquired before the Flooding; (b) Image Acquired after the Flooding; (c) the Ground Truth Change Map Used as Reference in the Experiments.



(a) (b) (c)

Figure 5.6: Multi-Temporal SAR Images of Bangladesh: (a) Image Acquired Before the Flooding; (b) Image Acquired after the Flooding; (c) the Ground Truth Change Map Used as Reference in the Experiments.

5.4 Experimental Results

5.4.1 Dataset Description and Experiment Design

In order to assess the effectiveness of the proposed approach, two datasets with different characteristics were considered in the experiments. The first dataset represents a section (301 × 301 pixels) of two SAR images acquired by the European Remote Sensing 2 (ERS2) satellite SAR sensor over an area near the city of Bern, Switzerland, in April and May 1999, respectively [33]. Between the two acquisition dates, the river flooded parts of the cities of Thun and Bern. Therefore, the valley between Bern and Thun was selected as a test site for detecting flooded areas. The Bern data set images and the available ground truth change between these two images are shown in Fig. 5.5. The second dataset used in the experiments consists of a section (300 × 300 pixels) of two SAR images acquired over a region in Bangladesh. From the two images of Bangladesh, it is possible to analyze which parts of the area were affected by the flooding that occurred just before the first acquisition data. The images and the ground truth are shown in Fig. 5.6. As shown in Figs. 5.5 and 5.6, the change area in the Bern dataset is relatively small while the change area in the Bangladesh dataset is large. Accordingly, these are proper datasets for assessing the robustness of the proposed algorithm to the variability in the change area and the scalability of the proposed algorithm.

For performing the evaluation, both qualitative and quantitative measurements are conducted. For the qualitative measurement, the final binary change detection map generated by using the automatic change detection method was visually compared with the ground truth change detection map. The quantitative measures are computed for comparing the computed change detection map against the ground truth change detection map as follows:

1) False Alarm (FA): This measure is given by the number of ground-truth unchanged pixels that were incorrectly classified as changed ones. The false alarm rate P_{FA} is computed in percentage as $P_{FA} = N_{FA}/N_{unchange} \times 100\%$, where N_{FA} is the total number of detected false alarm pixels and $N_{unchange}$ is the total number of unchanged pixels in the ground truth change detection map.

2) Miss Detection (MD): This measure is given by the number of ground-truth changed pixels that were mistakenly determined as unchanged ones. The missed detection rate $P_{MD} = N_{MD}/N_{change} \times 100\%$, where N_{MD} is the total number of miss-detected pixels and N_{change} is the total number of change pixels in the ground truth change detection map.

3) Total Error (TE): This measure is based on the total number of incorrect detections made, which is the sum of the false alarms and the missed detections. Thus, the total error rate $P_{TE} = (N_{FA} + N_{MD})/(N_{unchange} + N_{change}) \times 100\%$.

The similarity map can also be useful by itself [54]. Usually, the users of a similarity map investigate not only the final binary map but also use the similarity map as an indicator of the intensity of the change. In order to evaluate the quality of a similarity map independently of the choice of the thresholding algorithm, the receiver operating characteristic (ROC) is used. The ROC curve can be considered as the evolution of the true positive rate (TPR) as a function of the false alarm rate (FPR), where the TPR is defined as the fraction of correctly detected changes. The area under the ROC curve is a good change performance measure. The larger the area under the ROC curve, the better the performance is. Note that the focus of this work is to generate an accurate similarity map rather than to obtain a binary change map. Therefore, in order to generate a binary change map from the computed similarity map, an optimal threshold corresponding to the point nearest to (0.0, 1.0) and lying on the ROC curve, is selected to obtain the best tradeoff between detection and false alarm.

In order to assess the effectiveness of the proposed change detection approach, three main experiments have been carried out aimed at analyzing: 1) the effects of the size of the local neighborhood support (window size) in the spatial domain; 2) the effects of the scales and window sizes in the wavelet domain; 3) the change detection performance as compared to other existing change detection algorithms.

The first experiment is intended to assess the effect of the size of the local neighborhood support on the performance of our proposed algorithm in the spatial domain. To this end, a set of similarity maps and corresponding binary change maps is generated by using our proposed method with different window sizes ranging from 3×3 to 33×33. Finally, the FA, MD, and TE are computed for each binary change map corresponding to each window size.

In the second experiment, the aim is to evaluate the effect of the wavelet scales and window sizes on the performance of our proposed algorithm in the wavelet domain. For this purpose, different scales and window sizes are used to compute the final binary change map. In particular, each sliding window, with size ranging form 3×3 to 33×33, is decomposed into L = (1, 2, 3) scales using an undecimated wavelet transform with a Daubechies filter bank (db2). Also, the FA, MD, and TE measures are used to quantitatively access the effect of scales and window sizes on the resulting change detection map.

In the third experiment, the effectiveness of the proposed change detection algorithm is assessed by comparing it with other methods in terms of the resulting similarity maps and the binary change detection maps. In the spatial domain, the classical mean ratio detector (MRD) [78], and the recently proposed Pearson-based KL detector (PKLD) and cumulant-based KL detector (CKLD) of [54] are selected for comparison. It is stated in [54] that the CKLD outperforms other model-based methods such as PKLD. In the wavelet domain, the recent methods proposed in [34] are selected for comparison. In [34], the generalized Gaussian and Gamma distributions are used to model the subbands coefficient magnitudes in the wavelet domain and the Kullback-Leibler divergence is used to compute the similarity maps. These two wavelet-based methods are abbreviated as GGDKLD and GΓDKLD, respectively. The qualities of the similarity maps obtained by these methods and our proposed algorithm are assessed in terms of the ROC curves. Finally, the FA, MD, TE measures are computed for the computed change detection maps.

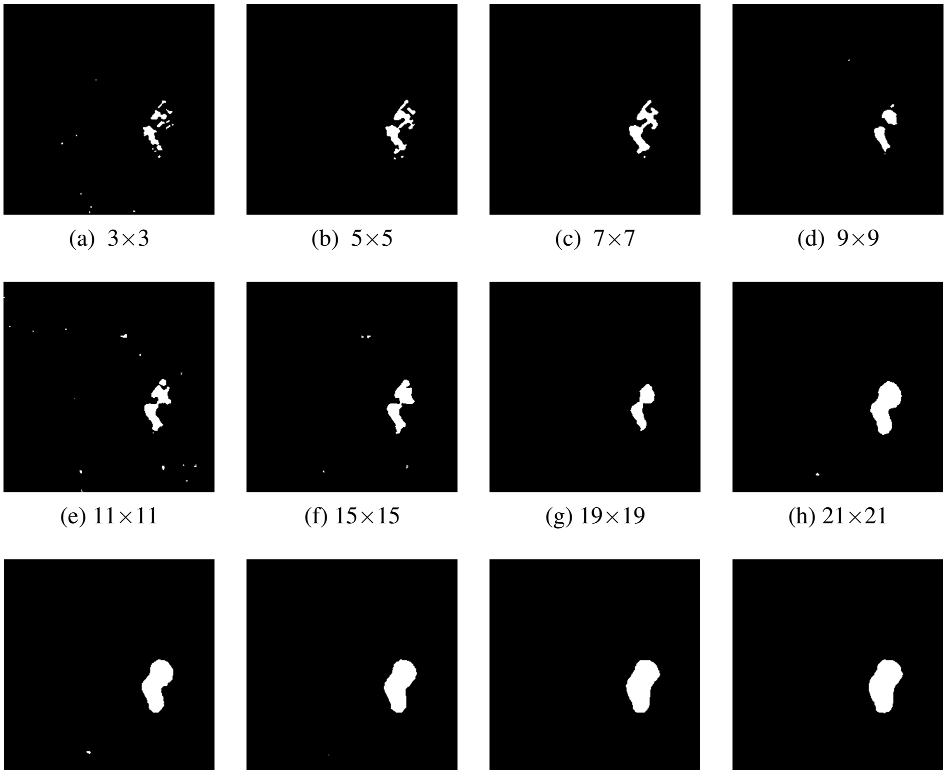
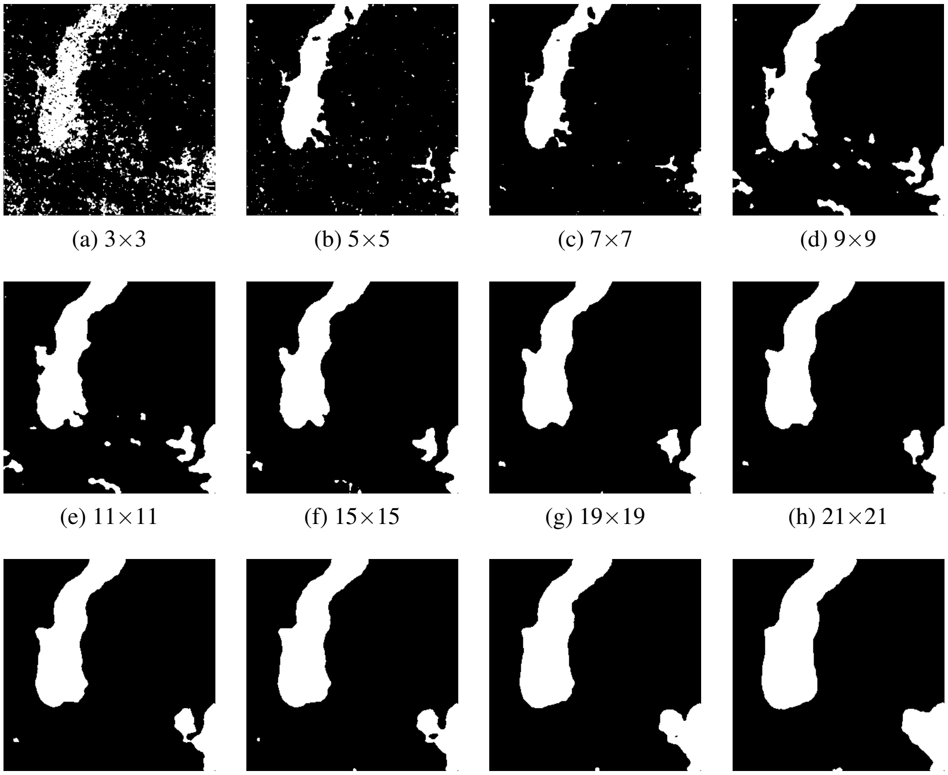

(a) 3×3 (b) 5×5 (c) 7×7 (d) 9×9
(e) 11×11 (f) 15×15 (g) 19×19 (h) 21×21
(i) 23×23 (j) 25×25 (k) 29×29 (l) 33×33

Figure 5.7: Final Change Detection Maps Obtained by the Proposed Algorithm on the Bern SAR Image Dataset in the Spatial Domain with Different Window Sizes.

5.4.2 The Effects of Window Size in the Spatial Domain

The proposed GMM change detection algorithm in the spatial domain utilizes a local neighborhood support in calculating the similarity of the two images. In order to show the effect of the size of the local neighborhood support on the performance of the proposed spatial-domain GMM change detection algorithm, change detection experiments on two different SAR image datasets are conducted for different local neighborhood sizes. The final binary change detection maps that are obtained using the proposed spatial-domain GGM algorithm for the Bern and Bangladesh SAR image datasets with different window sizes, are shown in Fig. 5.7 and Fig. 5.8, respectively. Note that the optimal threshold is selected by using the ROC curve of the similarity map to obtain the final binary map. As shown in Figs. 5.7 and 5.8, the change detection results obtained for different window sizes illustrate the trade-off between spatial detail preservation and noise reduction.



(a) 3×3 (b) 5×5 (c) 7×7 (d) 9×9

(e) 11×11 (f) 15×15 (g) 19×19 (h) 21×21

(i) 23×23 (j) 25×25 (k) 29×29 (l) 33×33

Figure 5.8: Final Change Detection Maps Obtained by the Proposed Algorithm on the Bangladesh SAR Image Dataset in the Spatial Domain with Different Window Sizes.

Table 5.3: Performance Results for the Obtained Change Detection Maps Using the Proposed Change Detection Algorithm on the Bern SAR Image Dataset in the Spatial Domain with Different Window Sizes (WS).

	FA	MD	TE
3×3	0.06%	42%	0.6%
5×5	0.14%	21.04%	0.41%
7×7	0.26%	20.26%	0.51%
9×9	0.43%	47.19%	1.02%
11×11	0.45%	30.57%	0.83%
15×15	0.20%	48.83%	0.81%
19×19	0.35%	41.21%	0.86%
21×21	0.34%	44.76%	0.91%
23×23	0.17%	61.21%	0.95%
25×25	0.24%	56.88%	0.96%
29×29	0.32%	56.53%	1.04%
33×33	0.6%	41.99%	1.17%

Table 5.4: Performance Results for the Obtained Change Detection Maps Using the Proposed Change Detection Algorithm on the Bangladesh SAR Image Dataset in the Spatial Domain With Different Window Sizes (WS).

	FA	MD	TE
3×3	5.34%	17.71%	6.85%
5×5	0.22%	13.11%	1.79%
7×7	0.32%	21.85%	2.94%
9×9	2.55%	5.54%	2.92%
11×11	2.18%	6.42%	2.7%
15×15	1.6%	9.26%	2.53%
19×19	1.67%	10.42%	2.74%
21×21	1.87%	10.53%	2.93%
23×23	2.24%	10.2%	3.21%
25×25	2.67%	9.91%	3.55%
29×29	3.61%	8.86%	4.25%
33×33	4.72%	7.54%	5.06%

In particular, change detection results with a smaller window size are more sensitive to noise interference while preserving more details of image content. In contrast, change detection results with a larger window size are less subject to noise interference, but show loss of image details. Tables 5.3 and 5.4 show the performance results of the proposed spatial-domain method in terms of FA, MD, and TE for the Bern and Bangladesh datasets, respectively. From these tables, it can be seen that the lowest total error rate (TE) is obtained for a window size of 5×5.

5.4.3 The Effects of Scale and Window Size in the Wavelet Domain

Various scales and window sizes are used for evaluating the performance of the proposed GMM change detection algorithm in the wavelet domain. Figs. 5.9 to 5.12 show the obtained change detection maps using the Bern (Figs. 5.9 & 5.10) and the Bangladesh (Figs. 5.11 & 5.12) datasets for different scales and window sizes. From Figs. 5.9 and 5.11, it can be seen that, for the fixed window size (5×5), the change detection maps obtained at scale 3 are more similar to the ground-truth change detection maps (Figs. 5.5 (c) & 5.6 (c)) than those obtained at scale 1 and 2. Figs. 5.10 and 5.12 show that, at the same scale (scale $=$ 3), the proposed wavelet-domain change detector with window size 5×5 yields better change detection results than with other window sizes (15×15 and 33×33).

Tables 5.5 and 5.6 show FA, MD, and TE values of the change detection results obtained by our proposed wavelet-domain change detection algorithm for different scales (1 to 3) and window sizes (ranging from 3×3 to 33×33) using the Bern and Bangladesh SAR image datasets, respectively. As shown in Tables 5.5 and 5.6, for the fixed window size, the total error rate (TE) decreases as the number of scales increases. The wavelet subband at a higher scale includes more geometrical details and less speckle noise than the subbands at lower scales, and thus yields better performance. In addition, from Tables 5.5 and 5.6, it can be seen that, at each scale, the lowest TE is obtained for a window size of 5×5.

5.4.4 Comparison with Other Existing Change Detectors in the Spatial Domain

As indicated before, the similarity map can also be useful by itself and the ROC can be used to assess the quality of the similarity map. Fig. 5.13 shows the ROC plots corresponding to the similarity maps obtained using the proposed spatial-domain GMM-based KL detector (SGMMKLD) method

Table 5.5: Performance Results of the Proposed Change Detection Algorithm Using the Bern SAR Image Dataset in the Wavelet Domain for Different Scales and Window Sizes.

	Scale 1			Scale 2			Scale 3		
	FA	MD	TA	FA	MD	TA	FA	MD	TA
3×3	0.11%	95.56%	1.7%	0.12%	92.42%	1.3%	0.06%	92.98%	1.25%
5×5	0.31%	51.08%	0.68%	0.2%	38.44%	0.56%	0.15%	25.02%	0.49%
7×7	0.33%	47.22%	0.93%	0.31%	27.12%	0.65%	0.17%	30.91%	0.56%
9×9	0.34%	52.51%	1.01%	0.21%	42.05%	0.74%	0.27%	30.48%	0.65%
11×11	0.17%	68.37%	1.04%	0.37%	37.2%	0.84%	0.26%	33.68%	0.69%
15×15	0.51%	49.5%	1.13%	0.31%	45.42%	0.89%	0.18%	47.53%	0.79%
19×19	0.49%	51.3%	1.14%	0.39%	43.24%	0.94%	0.3%	41.3%	0.82%
21×21	0.52%	52.63%	1.18%	0.32%	49.77%	0.95%	0.26%	45.97%	0.84%
23×23	0.4%	66.8%	1.25%	0.35%	51.85%	1.01%	0.4%	38.53%	0.89%
25×25	0.54%	58.03%	1.28%	0.36%	53.14%	1.03%	0.21%	57.66%	0.95%
29×29	0.45%	71.79%	1.36%	0.42%	52.94%	1.09%	0.41%	47.88%	1.02%
33×33	0.67%	66.32%	1.51%	0.55%	54.22%	1.23%	0.57%	42.86%	1.11%


(a) scale=1 (b) scale=2 (c) scale=3

Figure 5.9: Change Detection Results Obtained by the Proposed Algorithm on the Bern SAR Image Dataset in the Wavelet Domain at Different Scales with A Window Size of 5×5.


(a) 5×5 (b) 15×15 (c) 33×33

Figure 5.10: Change Detection Results Obtained by the Proposed Algorithm on the Bern SAR Image Dataset in the Wavelet Domain at the Same Scale (Scale = 3) But with Different Window Sizes.

using the Bern dataset for different window sizes. For comparison, Fig. 5.13 also shows the ROC plots corresponding to similarity maps obtained using the classical pixel-based mean ratio detector

Table 5.6: Performance Results of the Proposed Change Detection Algorithm Using the Bangladesh SAR Image Dataset in the Wavelet Domain for Different Scales and Window Sizes.

	Scale 1			Scale 2			Scale 3		
	FA	MD	TA	FA	MD	TA	FA	MD	TA
3×3	0.9%	19.1%	3.12%	0.73%	14.6%	2.42%	0.34%	17%	2.37%
5×5	0.73%	13.3%	2.26%	0.44%	10.66%	1.69%	0.5%	8.61%	1.49%
7×7	0.93%	12.62%	2.35%	0.57%	10.81%	1.82%	0.51%	9.37%	1.59%
9×9	0.91%	14.12%	2.52%	0.94%	9.68%	2.0%	0.64%	9.58%	1.73%
11×11	1.14%	14.21%	2.74%	1.04%	10.55%	2.2%	0.84%	9.76%	1.93%
15×15	1.25%	16.9%	3.15%	1.12%	13.45%	2.63%	0.94%	11.92%	2.28%
19×19	1.64%	18.12%	3.65%	1.12%	16.34%	2.98%	1.24%	12.36%	2.6%
21×21	1.22%	22.58%	3.82%	1.32%	16.25%	3.14%	1.18%	13.99%	2.74%
23×23	1.4%	22.44%	3.97%	1.11%	18.71%	3.25%	1.05%	15.85%	2.85%
25×25	1.48%	22.83%	4.08%	1.46%	16.93%	3.34%	1.06%	16.5%	2.94%
29×29	1.66%	23.54%	4.33%	1.24%	20.42%	3.58%	1.25%	16.63%	3.12%
33×33	1.68%	26.14%	4.67%	1.49%	20.93%	3.87%	1.31%	18.16%	3.37%

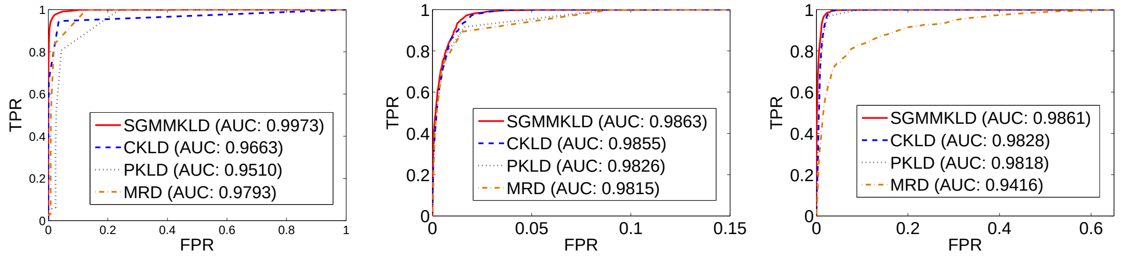
(a) scale=1 (b) scale=2 (c) scale=3

Figure 5.11: Change Detection Results Obtained by the Proposed Algorithm on the Bangladesh SAR Image Dataset in the Wavelet Domain at Different Scales with A Window Size of 5×5.

(a) 5×5 (b) 15×15 (c) 25×25

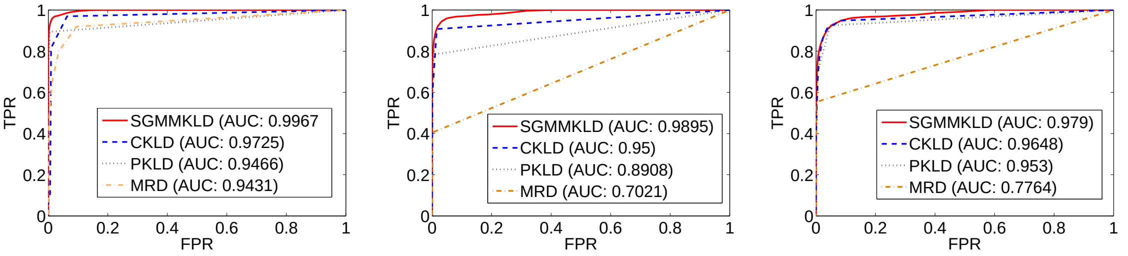
Figure 5.12: Change Detection Results Obtained by the Proposed Algorithm on the Bangladesh SAR Image Dataset in the Wavelet Domain at the Same Scale (Scale = 3) But with Different Window Sizes.

(MRD) [78], the Pearson-based KL detector (PKLD) [54], and the cumulant-based KL detector (CKLD) [54] for the Bern dataset. The corresponding area under the ROC curve (AUC) for each



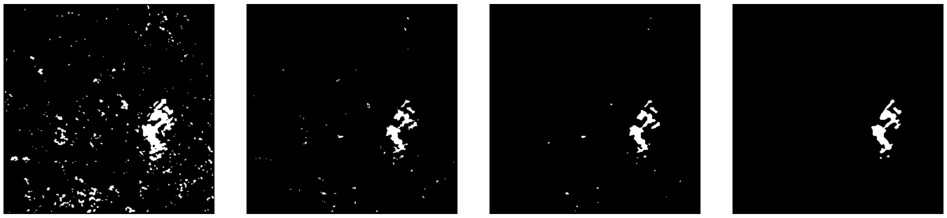
(a) 5×5 (b) 15×15 (c) 33×33

Figure 5.13: ROC Plots Comparison between MRD [78], PKLD [54], CKLD [54], And the Proposed SGMMKLD with Different Window Sizes Using the Bern SAR Image Dataset.



(a) 5×5 (b) 15×15 (c) 33×33

Figure 5.14: ROC Plots Comparison between MRD [78], PKLD [54], CKLD [54], And the Proposed SGMMKLD with Different Window Sizes Using the Bangladesh SAR Image Dataset.



(a) MRD [78] (b) PKLD [54] (c) CKLD [54] (d) SGMMKLD

Figure 5.15: Change Detection Results from Different Spatial-Domain Algorithms for the Bern SAR Image Dataset with A Window Size of 5×5.

Table 5.7: FD, MD, And TE of The Binary Change Maps (as Shown in Fig. 5.15) Resulting From Different Spatial-Domain Methods for the Bern SAR Image Dataset.

Detector	False detections	Missed detections	Total errors
MRD [78]	3%	2.68%	3%
PKLD [54]	0.36%	31.69%	0.76%
CKLD [54]	0.21%	33.25%	0.63%
SGMMKLD	0.14%	21.04%	0.41%

(a) MRD [78] (b) PKLD [54] (c) CKLD [54] (d) SGMMKLD

Figure 5.16: Change Detection Results from Different Spatial-Domain Algorithms for the Bangladesh SAR Image Dataset with A Window Size of 5×5.

Table 5.8: FD, MD, And TE of the Binary Change Maps (as Shown in Fig. 5.16) Resulting from Different Spatial-Domain Methods for the Bangladesh SAR Image Dataset.

Detector	False detections	Missed detections	Total errors
MRD [78]	3.55%	4.32%	3.64%
PKLD [54]	0.01%	29.85%	3.65%
CKLD [54]	0.32%	22.56%	2.95%
SGMMKLD	0.22%	13.11%	1.79%

change detector is also computed and shown in Fig. 5.13. The results in Fig. 5.13 clearly shows that the proposed SGMMKLD outperforms existing spatial-domain change detectors for all window sizes from small to large. This is due to the fact that the locally adaptive mixture model (GMM) is more suitable and flexible than the single parametric mathematical distribution models (e.g., CKLD and PKLD) to approximate the local distribution of the actual SAR image. It is interesting to also note that the ratio criterion is not always worse than the pdf-based detectors. As shown in Figs. 5.13 (a) and (b), it can be observed that, even though the performance of the pixel-based MRD is not so good as our proposed SGMMKLD, the MRD exhibits a competitive performance as compared to the existing pdf-based detectors (PKLD and CKLD) for small and medium window sizes. To further demonstrate the detection capability of the proposed SGMMKLD as compared to other change detectors when the change area is relatively large, the detectors were applied to the Bangladesh SAR image dataset. From Fig. 5.14, it can be clearly seen that the change detection performance of our proposed SGMMKLD is superior to other existing spatial-domain change detectors for all window sizes.

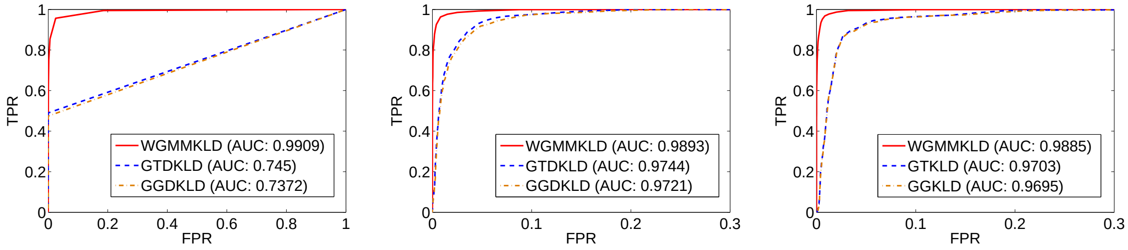
From Figs. 5.13 and 5.14, it can be seen that the small window size 5×5 achieves better performance as compared to larger window sizes while achieving lower computational complexity.

For this purpose, the change similarity maps that were obtained using a window size of 5×5, are used to compute the final binary maps. The change detection maps obtained using different change detectors are shown in Figs.5.15 and 5.16 for the Bern and Bangladesh datasets, respectively. Tables 5.7 and 5.8 show the corresponding FA, MD and TE performance measures for the Bern and Bangladesh datasets, respectively. From Tables 5.7 and 5.8, it can be seen that the proposed SGMMKLD detector produces the lowest total error rate of 0.41% and 1.79% for the Bern and Bangladesh datasets, respectively.

5.4.5 Comparison with Other Existing Change Detectors in the Wavelet Domain

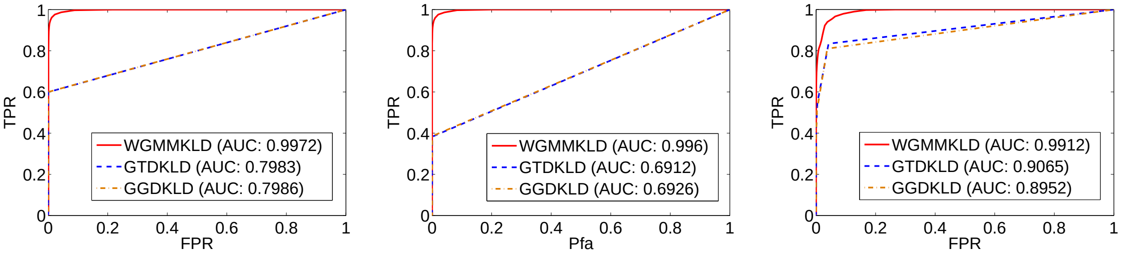
The performance of the proposed wavelet-domain GMM-based KL detector (WGMMKLD) is compared with the existing wavelet-based GGDKLD and GΓDKLD change detectors of [34]. Figs. 5.17 and 5.18 show, for the Bern and Bangladesh datasets, respectively, the ROC plots corresponding to the similarity maps obtained using the GGDKLD, the GΓDKLD and our proposed WGMMKLD for different window sizes at scale 3. As shown in Fig. 5.17, the performance of the proposed WGMMKLD is superior as compared to existing wavelet-domain change detectors for all window sizes since it is able to adapt to the local statistical distribution of the wavelet coefficients of the SAR images. The results shown in Fig. 5.18 further validate that the proposed WGMMKLD outperforms existing wavelet-domain change detectors for all window sizes when the change area is relatively large.

From Figs. 5.17 and 5.18, it can be seen that the small window size 5×5 achieves better performance and lower computational complexity as compared to larger window sizes. Consequently, the change similarity maps that were obtained using a window size of 5×5 are used to compute the final binary maps. Figs. 5.19 and 5.20 show the change detection maps obtained using different wavelet-domain change detectors for the Bern and Bangladesh datasets, respectively. As shown in Figs. 5.19 and 5.20, the proposed WGMMKLD yields better change detection results as compared to the GGDKLD and GΓDKLD. Tables 5.9 and 5.10 show the corresponding FA, MD and TE performance measures for the Bern and Bangladesh datasets, respectively. It can be seen from Tables 5.9 and 5.10 that the total error rates corresponding to the change maps that are obtained using the proposed WGMMKLD are the lowest among all wavelet-domain change detectors for both the Bern (0.49%) and Bangladesh (1.60%) image datasets, respectively.



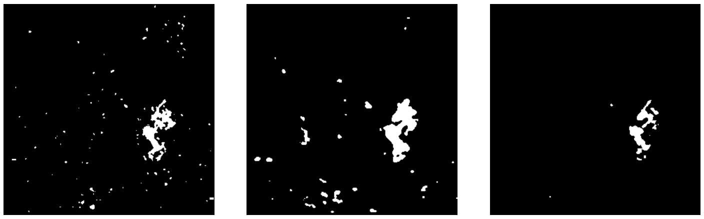
(a) 5×5 (b) 15×15 (c) 33×33

Figure 5.17: ROC Plots Comparison between GGDKLD [34], GΓDKLD [34], And the Proposed WGMMKLD with Different Window Sizes And Scale=3 for the Bern Sar Image Dataset.



(a) 5×5 (b) 15×15 (c) 33×33

Figure 5.18: ROC Plots Comparison between GGDKLD [34], GΓDKLD [34], And the Proposed WGMMKLD with Different Window Sizes And Scale=3 for the Bangladesh Sar Image Dataset.



(a) GGDKLD [34] (b) GΓDKLD [34] (c) WGMMKLD

Figure 5.19: Change Detection Results from Different Wavelet-Domain Algorithms on the Bern SAR Image Dataset With A Window Size Of 5×5 And Scale=3.

Table 5.9: FD, MD, And TE of the Binary Change Maps (as Shown in Fig. 5.19) Resulting from Different Wavelet-Domain Methods on the Bern SAR Image Dataset.

Detector	False detections	Missed detections	Total errors
GGDKLD [34]	2.01%	3.2%	2.03%
GΓDKLD [34]	1.14%	23.72%	1.43%
WGMMKLD	0.15%	25.02%	0.49%

(a) GGDKLD [34] (b) GΓDKLD [34] (c) WGMMKLD

Figure 5.20: Change Detection Results from Different Wavelet-Domain Algorithms on the Bangladesh SAR Image Dataset With A Window Size Of 5×5 And Scale=3.

Table 5.10: FD, MD, And TE of the Binary Change Maps (as Shown in Fig. 5.20) Resulting from Wavelet-Domain Different Methods on the Bangladesh SAR Image Dataset.

Detector	False detections	Missed detections	Total errors
GGDKLD [34]	0%	50.35%	6.14%
GΓDKLD [34]	0%	49.98%	6.09%
WGMMKLD	0.5%	8.61%	1.49%

5.5 Summary

In this chapter, a novel change detector for multitemporal SAR images is proposed and applied in both the spatial and wavelet domains. This detector is based on an analysis of the local distribution of SAR images using a Gaussian Mixture Model (GMM) and the KL distance. The locally adaptive Gaussian Mixture Model is more suitable and flexible than the single parametric mathematical distribution models to approximate the distribution of the actual SAR image, which shows a varied scene presenting several distinct land-cover typologies. Compared to existing detectors with higher order statistics both in the spatial and wavelet domains, the proposed GMM-based change detector exhibits a superior change detection performance.

Chapter 6

CHANGE DETECTION BASED ON THE DIVISIVE NORMALIZATION TRANSFORM

6.1 Introduction

Although the change detection algorithms introduced in Chapter 5 achieved notable success, they have some important limitations. In the spatial domain, the change detection algorithms usually require preprocessing to despeckle the multi-temporal images, these preprocessing techniques can achieve different tradeoffs between detail preservation and noise reduction. But these are contrasting properties; in other words, high accuracy in homogeneous areas usually requires an intensive despeckling phase, which in turn can degrade the details in the SAR image [147]. The proposed wavelet-domain GMM-based change detector can relieve this problem since the wavelet image representation can achieve different scales (levels) of representation of the change signal. Each scale is characterized by a different tradeoff between speckle reduction and preservation of details [147]. However, there exists strong high-order statistical dependencies between neighboring wavelet coefficients, which are not accounted for by the proposed wavelet-domain GMM-based method. Furthermore, the wavelet decomposition is restrictive due to its linear nature and cannot represent possible nonlinear effects. Finally, the computation cost of the proposed GMM algorithm is expensive as compared to a single-parameter Guassian model.

In this work, we propose a new image comparison method that is inspired by the recent success of the divisive normalization transform (DNT) as a statistically and perceptually motivated image representation [148, 149]. This local gain-control divisive normalization model is well-matched to the statistical prosperities of optical images, as well as the perceptual sensitivity of the human visual system [150, 151]. The DNT is built upon linear transform models, where each coefficient (or neuronal response) is normalized (divided) by the energy of a cluster of neighboring coefficients (neighboring neuronal response) [152]. This procedure can explain nonlinearities in the responses of mammalian cortical neurons, and nonlinear masking phenoma in human visual perception, and was also empirically shown to produce approximately Gaussian marginal distributions and to reduce the statistical dependencies of the original linear representation [153]. Therefore, this thesis proposes a change detection method for SAR images in the DNT domain rather than in the wavelet domain due to the superior properties of the DNT as mentioned above.

This chapter is organized as follows. The divisive normalization transform for SAR images is described in Section 6.2. Section 6.3 presents the proposed image comparison algorithm based on the DNT image representation. Section 6.4 presents the results on real data using the proposed detector. Conclusions are drawn in Section 6.5.

6.2 Divisive Normalization-based Image Representation

6.2.1 Computation of the Divisive Normalization Transformation

The computation of the DNT involves two stages [151]:

$$x \xrightarrow{T} w \xrightarrow{R} y \tag{6.1}$$

where the image x is first analyzed by a linear image decomposition T and then followed by a non-linear transform (the divisive normalization) R. Here, the wavelet image decomposition is employed as the linear image decomposition model since it provides a convenient framework of localized representation of images simultaneously in space, frequency (scale) and orientation [154]. Let w and y represent the wavelet and DNT coefficients, respectively, and $y = w/z$, where z is the positive divisive normalization factor that is calculated as the energy of a cluster of coefficients that are neighbors of the coefficient y in space, scale, and orientation.

In the literature, several approaches [149,153,155] were proposed to compute the normalization factor z. Among these approaches, the convenient estimation method in [156] is adopted. This method derives the factor z through the Gaussian scale mixtures (GSM) model. A random vector W is a GSM if it can be expressed as the product of two independent components: $W = zU$, where the mixing multiplier z is a positive scalar random variable; $U \sim \mathrm{N}(0, Q)$ is a zero-mean Gaussian random vector with covariance matrix Q, and z and U are independent. As a consequence, any GSM variable has a density given by an integral [156]:

$$p_W(W) = \int \frac{1}{[2\pi]^{N/2}|z^2 Q|^{1/2}} \exp\left(-\frac{W^T Q^{-1} W}{2z^2}\right) \phi_z(z) dz \tag{6.2}$$

where N is the length of the GSM random vector W, and $\phi_z(z)$ is the probability density of the mixing variable z. The GSM model expresses the density of a random vector as a mixture of Gaussians with the same covariance structure Q but scaled differently by z [152]. A special case of a GSM is a finite mixture of Gaussians, where z is a discrete random variable. This GSM model was shown to represent well the statistics of the wavelet coefficients of images, where the vector W is formed by clustering a set of neighboring wavelet coefficients within a subband, or across neighboring subbands in scale and orientation [156]. The GSM model has also found successful applications in image coding [155], image denoising [157], image restoration [158], and image quality assessment [154].

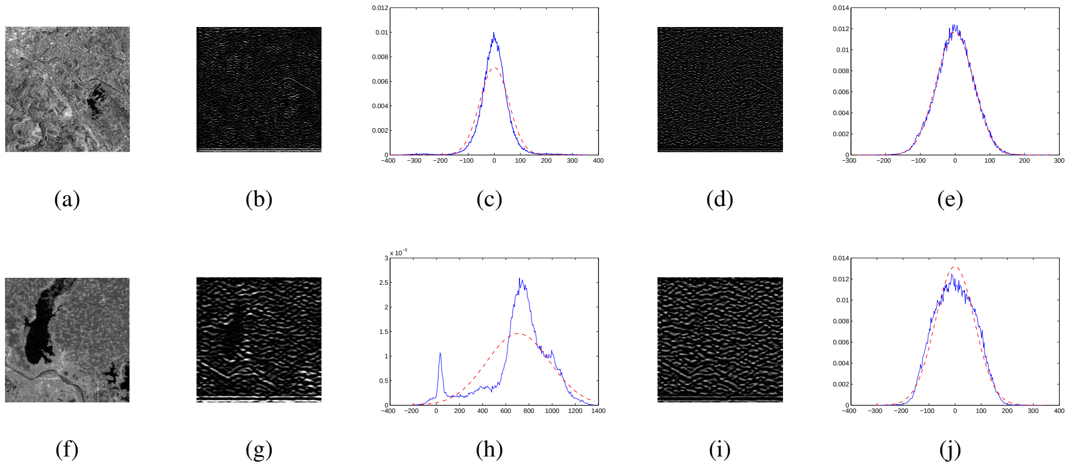
For our application, the wavelet coefficients of SAR images are modeled as a GSM random vector W that is formed by clustering a set of neighboring wavelet coefficients within a subband and that is normalized by the mixing multiplier. The general form of the GSM model allows for the mixing multiplier z to be a continuous random variable at each location of the wavelet subbands. To simplify the model, we assume that z only takes a fixed value at each location (but varies over space and subbands). The benefit of this simplification is that when z is fixed, W_{DNT} is simply a zero-mean Gaussian vector with covariance z^2Q. Thus, it becomes simple to estimate the normalization factor z in the DNT representation from the neighboring coefficients. The coefficient cluster W_{DNT} moves step by step as a sliding window across a wavelet subband, resulting in a spatially varying normalization factor z [152]. In our implementation, the normalization factor computed at each step is only applied to the center coefficient w_c of the vector W_{DNT}, and the normalized coefficient becomes $y_c = w_c/\hat{z}$, where $\hat{z}$ is the estimation of z. An efficient method to obtain $\hat{z}$ is by a maximum-likelihood estimation [156] given by

$$\begin{aligned} \hat{z} &= \arg\max_{z}\{\log p(z|W_{DNT})\} \\ &= \sqrt{W_{DNT}^T Q^{-1} W_{DNT}/N_{DNT}} \end{aligned} \tag{6.3}$$

where $Q = E[UU^T]$ is the positive definite covariance matrix of the underlying Gaussian vector U and is estimated from the entire wavelet subband before estimating local z, and N_{DNT} is the length of vector W_{DNT}, or the size of the sliding window of the neighboring wavelet coefficients [152]. In our implementation, Q is an $N_{DNT} \times N_{DNT}$ diagonal matrix, where the main diagonal is identical to the variance of the entire wavelet subband.

6.2.2 Image Statistics in Divisive Normalization Transform Domain

As will shown in the next section, the proposed DNT-based change detection algorithm is essentially based on the statistics of the transform coefficients in the DNT domain. Before the development



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

(f) (g) (h) (i) (j)

Figure 6.1: (a),(f): Original SAR Images; (b),(g): Wavelet Coefficients; (c),(h): PDFs of Wavelet Coefficients (Solid Curves) Fitted with A Gaussian Model (Dashed Curves); (d),(i): DNT Coefficients; (e),(j): PDFs of DNT Coefficients (Solid Curves) Fitted with A Gaussian Model (Dashed Curves).

of the specific change detection algorithm, it is useful to observe variations of image statistics before and after the DNT. In Fig. 6.1, we compare the distributions of the original wavelet subbands and the same subbands after DNT, for a pair of SAR images. In Figs. 6.1 (c) & (h), the original wavelet coefficient distributions (solid curves) of the SAR images are fitted using a Gaussian model (dashed curve). The noticeable difference between the two curves (the actual pdf and the fitted Gaussian model) shows that the original wavelet coefficients are highly non-Gaussian. In contrast, as shown in Figs.6.1 (e) & (j), the distribution of the coefficients after DNT can be well fitted with a Gaussian. A similar conclusion is obtained for other SAR images.

6.3 Change Detection in the DNT Domain

6.3.1 Proposed Change Detector Algorithm in the DNT Domain

Let us consider two co-registered SAR intensity images I_{X1} and I_{X2} acquired over the same geographical area at two different times t_{X1} and t_{X2}, respectively [54]. Our aim is to generate a change detection map that represents changes that occurred on the ground between the acquisition dates [54]. This change detection problem can be modeled as a binary classification problem where 1 represents changed pixels and 0 represents unchanged pixels.

We propose a change detection algorithm by analyzing the difference in the local statistics of

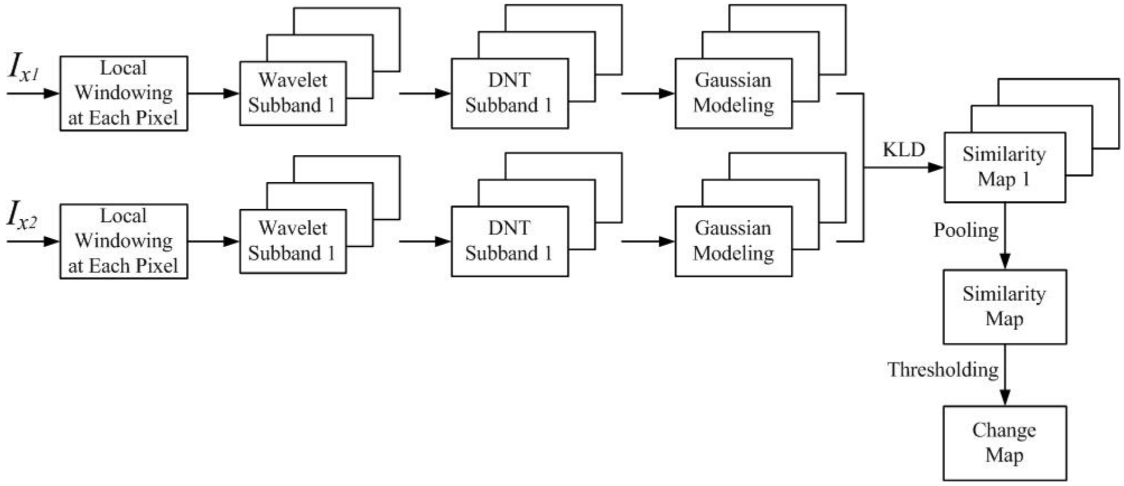


Figure 6.2: The Block Diagram Of the Proposed DNT-Based Change-Detection Algorithm.

the DNT coefficients of two acquired SAR images. A pixel will be considered as a changed pixel if the local statistical distribution of the DNT coefficients significantly changes from one image to the other. In order to quantify this change, the Kullback-Leibler (KL) divergence [105] between two probability density functions is used. The framework of the proposed DNT-based method is shown in Fig. 6.2. The first step is to decompose a $n_{WS} \times n_{WS}$ sliding window at each pixel into multiple subbands by using the wavelet transform. Then, the DNT is performed for each subband. The third step is to estimate for each subband the parameters (mean and variance of the Gaussian distribution) governing the distribution of DNT coefficients. The fourth step is to compute the symmetric KL divergence between the estimated Gausssian PDFs of two subbands at the same level and orientation. Thus, the similarity map for each pair of subbands at the same level and orientation is obtained. Finally, all the subband-specific similarity maps are combined to obtain a final similarity map $SMAP(n_1,n_2)$ by summing the similarity maps over all subbands as follows:

$$SMAP(n_1,n_2) = \sum_{i=1}^{L} \sum_{j=1}^{M} D(p_{y_1;i,j}(n_1,n_2) \| p_{y_2;i,j}(n_1,n_2)) \tag{6.4}$$

where (n_1,n_2) is the location of the pixel where the sliding window is centered; L and M are the numbers of the scales and orientations, respectively; and $p_{y_1;i,j}(n_1,n_2)$ and $p_{y_2;i,j}(n_1,n_2)$ are the estimated distributions of DNT coefficients at scale i and orientation j for the considered local windows being compared. In (6.4), D is the symmetric KL distance as described in Chapter 4. As described earlier in Section 6.2.2, since the divisive normalization transform produces approximately Gaussian distributions, by using the Gaussian model, the symmetric KL divergence can be computed as:

$$D(p_{y_1}, p_{y_2}) = \frac{\sigma_{y_1}^4 + \sigma_{y_2}^4 + (\mu_{y_1} - \mu_{y_2})^2(\sigma_{y_1}^2 + \sigma_{y_2}^2)}{2\sigma_{y_1}^2 \sigma_{y_2}^2} - 1 \tag{6.5}$$

where μ_{y_1} and μ_{y_2} are, respectively, the mean and standard deviation of y. In our application, μ_y and σ_y are always equal to zero because of the DNT produces approximately zero-mean Gaussian distributions. Consequently, Equation (6.5) reduced to:

$$D(p_{y_1}, p_{y_2}) = \frac{\sigma_{y_1}^4 + \sigma_{y_2}^4}{2\sigma_{y_1}^2 \sigma_{y_2}^2} - 1 \tag{6.6}$$

The final binary change detection map is obtained by thresholding the similarity map $SMAP(n_1, n_2)$.

6.3.2 Computational Complexity Analysis

The proposed DNT-domain change detection algorithm saves a significant number of computations with minimal overhead of operations per pixel for the divisive normalization transform. In this section, the computational complexity of the classical mean ratio detector (MRD) [78] and the more recent cumulant-based KL detector (CKLD) [54], and generalized Gamma KL detectors (GΓKLD) [34] are compared with our proposed spatial- and wavelet- domain GMM-based KL detectors (SGMMKLD and WGMMKLD) described in Chapter 5, and our proposed DNT-domain KL detector (DNTKLD). The computational complexity is quantified in terms of Operations Per Pixel (OPP). These operations include the operations for for-loop counters, arithmetic, conditional, logical, relational and bitwise operations. More specifically, the arithmetic operations consist of the addition, multiplication, division, square root, exponent and logarithm operation. Here, the computational complexity analysis mainly focuses on comparing the implementation speed of different algorithms rather than the power consumption. The processing time of the aforementioned operations are the same in terms of CPU clock cycle [159]. Thus, all those operations are treated equally in our analysis. Note that the focus of this work is to generate the similarity map and thus the computational complexity of the thresholding step is not considered.

- Computational complexity analysis of the proposed DNT-domain change detector

As shown in Fig. 6.2, the proposed DNT-domain change detector involves five steps not including the thresholding step. First, in our implementation, each $n_{WS}\times n_{WS}$ sliding window at each pixel is decomposed into L scales and 4 orientations using an undecimated wavelet transform with a Daubechies filter bank (db2), where both the 1D low-pass and high-pass filters have 4 coefficients. As a result, the 1D convolution requires 7 OPP including 4 multiplications and 3 additions. The wavelet decomposition at each scale employs 3 1D high-pass filters and 3 1D low pass filters in order to obtain 4 subbands corresponding to 4 orientations. Consequently, the wavelet transform requires $L\cdot 6\cdot 7\cdot n_{WS}^2 = 42L\cdot n_{WS}^2$ OPP for calculating all subbands. Secondly, the divisive transform is performed on each computed $n_{WS}\times n_{WS}$ wavelet subband (W). In our implementation, the normalization factor computed at each step is only applied to the center coefficient w_c of the vector W_{DNT}, where the coefficient cluster W_{DNT} moves step by step as a sliding window across a wavelet subband W. The normalized coefficient is computed as $y_c = w_c/\hat{z}$, where $\hat{z}$ is the normalization factor and can be estimated by using Equation (6.3). From Equation (6.3), it can be seen that the number of arithmetic operations that are required for divisive normalization for one coefficient within a subband is $2N_{DNT}+2$ OPP, where N_{DNT} is the size of the sliding window of the neighboring wavelet coefficients to compute the DNT. Thus, the number of arithmetic operations that are needed to compute the DNT for an entire $n_{WS}\times n_{WS}$ wavelet subband W is $n_{WS}^2\cdot(2N_{DNT}+2)$ OPP, which results in $4L\cdot n_{WS}^2\cdot(2N_{DNT}+2)$ OPP to perform the DNT for all coefficients over all subbands. The next step is to compute the similarity between two pixels for each $n_{WS}\times n_{WS}$ subband at the same level and orientation according to Equation (6.6). From Equation (6.6), it can be seen that this calculation requires 7 OPP. In addition, the estimation of σ_{y1} (or σ_{y2}) in the $n_{WS}\times n_{WS}$ DNT subband requires $3\cdot n_{WS}^2$ OPP. Therefore, the total OPP of similarity calculation between two pixels for each subband is $7+2\cdot 3\cdot n_{WS}^2$, which results in $4L\cdot(7+6n_{WS}^2)$ for computing similarity maps for all subbands. The final step is to sum up the similarity maps over all subbands to obtain a final similarity map according to Equation (6.4), which requires $4L$ OPP. Consequently, the total complexity of the proposed DNTKLD for a pixel equals to $2\cdot[42L\cdot n_{WS}^2+4L\cdot n_{WS}^2\cdot(2N_{DNT}+2)]+4L\cdot(7+6n_{WS}^2)+4L = 124L\cdot n_{WS}^2+16L\cdot N_{DNT}\cdot n_{WS}^2+32L$ OPP.

- Computational complexity analysis of proposed SGMMKLD and WGMMKLD

The spatial-domain GMMKLD (SGMMKLD) change detector mainly involves two steps as shown in Fig 5.3, including the GMM estimation for each local window of two SAR images and the computation of KL divergence between two GMMs for each location. Algorithm 1 as shown in Chapter 5 illustrates the pseudo-code for estimating the parameters of GMM. From Algorithm 1, it can be seen that the estimation of one Gaussian mixture model requires $N_{loop} \cdot [(K_{max}^2 + 4K_{max}) \cdot n_{WS}^2 + 3K_{max}^2 + 10K_{max} + 15]$OPP, where N_{loop} is the number of iterations to obtain an accurate convergence and K_{max} is the initial value of the number of the Gaussian components. Section 5.3.3 describes the procedure to compute the KL divergence between two GMMs. According to Equations (5.12), (5.13), and (5.14), it can be seen that the calculation KL divergence for two GMMs requires $31 \cdot K_{y1} + 31 \cdot K_{y2} - 4$ OPP, where K_{y1} and K_{y2} are the number of components for each GMM, respectively. In our implementation, usually, K_{y1} and K_{y2} are equal to 2 for most of time ($31 \cdot K_{y1} + 31 \cdot K_{y2} - 4 = 120$). Therefore, the total computational complexity of SGMMKLD is equal to $2N_{loop} \cdot [(K_{max}^2 + 4K_{max}) \cdot n_{WS}^2 + 3K_{max}^2 + 10K_{max} + 15] + 120$ OPP.

The wavelet-domain GMMKLD (WGMMMKLD) change detector mainly consists of four steps as shown in Fig 5.4. The first step is the wavelet transform and, as discussed before for the proposed DNTKLD, the computation of the wavelet transform for one sliding window requires $42L \cdot n_{WS}^2$ OPP. Next, each subband is modeled using a GMM, which requires $N_{loop} \cdot [(K_{max}^2 + 4K_{max}) \cdot n_{WS}^2 + 3K_{max}^2 + 10K_{max} + 15]$OPP as described for the SGMMKLD. Thirdly, the similarity of two subbands, one from each of the two acquired images, at the same level and orientation is computed. As described earlier, the similarity calculation of two GMMs requires 120 OPP; and thus the total computation for calculating all similarity measures for all subbands at each level and orientation is equal to $120 \cdot 4 \cdot L = 480L$ OPP. The final step is to sum up the subband-specific similarity maps over all subbands to obtain a final similarity map according to Equation (5.7), which requires $4L$ OPP. Consequently, the total complexity of the proposed WGMMKLD per pixel is $2 \cdot \{42L \cdot n_{WS}^2 + 4L \cdot N_{loop} \cdot [(K_{max}^2 + 4K_{max}) \cdot n_{WS}^2 + 3K_{max}^2 + 10K_{max} + 15]\} + 480L + 4L = 84L \cdot n_{WS}^2 + 8L \cdot N_{loop} \cdot [(K_{max}^2 + 4K_{max}) \cdot n_{WS}^2 + 3K_{max}^2 + 10K_{max} + 15] + 484L$ OPP.

Table 6.1: Operations Per Pixel (OPP) of the Proposed DNTKLD, SGMMKLD, and WGMMKLD Change Detectors and Comparison with Other Existing Change Detectors for an $n_{W_s} \times n_{W_s}$ Window.

		OPP
Spatial-domain	MRD [78]	$2n_{WS}^2 + 4$
	CKLD [54]	$16n_{WS}^2 + 259$
	SGMMKLD	$2N_{loop} \cdot [(K_{max}^2 + 4K_{max}) \cdot n_{WS}^2 +$ $3K_{max}^2 + 10K_{max} + 15] + 120$
Wavelet-domain	GΓDKLD [34]	$244L \cdot n_{WS}^2 + 648L$
	WGMMKLD	$84L \cdot n_{WS}^2 + 8L \cdot N_{loop} \cdot [(K_{max}^2 + 4K_{max}) \cdot n_{WS}^2$ $+3K_{max}^2 + 10K_{max} + 15] + 484L$
DNT-domain	DNTKLD	$124L \cdot n_{WS}^2 + 16L \cdot N_{DNT} \cdot n_{WS}^2 + 32L$

- Computational complexity analysis of existing change detectors

The computational complexity of existing change detectors is also roughly calculated, including the pixel-based MRD, spatial-domain CKLD, and wavelet-domain $G\Gamma KLD$.

As shown in Section 4.2.1, the MRD is computed according to Equation (4.1). From this equation, it can be seen that, for each pixel, the calculation of MRD requires 4 OPP and the estimation for the local mean values of the images before and after the event of interest requires $2 \cdot n_{WS}^2$ OPP, where $n_{WS} \times n_{WS}$ is the size of the local window. Consequently, the total computation complexity of MRD equals to $2n_{WS}^2 + 4$ OPP.

The framework to calculate the spatial-domain CKLD change detector was described in Section 4.2.2. First, the method of log-cumulants (MoLC), is used to estimate the parameters of two Edgeworth series expansions corresponding to two input $n_{WS} \times n_{WS}$ sliding windows, one from each of the two acquired images. The OPP for calculating the cumulants (up to the order of four) based on Equation (4.5) for one sliding window is $8n_{WS}^2 + 21$ OPP, which results in $2 \times (8n_{WS}^2 + 21) = 16n_{WS}^2 + 22$ OPP for estimating parameters for two sliding windows. Then, the one-side KL divergence for the Edgeworth series is calculated by using Equations (4.7) and (4.8), which requires $48 + 70 = 118$ OPP; it follows that computation of the symmetric KL requires $118 \times 2 + 1 = 237$ OPP. As a result, the total OPP for CKLD is equal to $16n_{WS}^2 + 22 + 237 = 16n_{WS}^2 + 259$ OPP.

The procedure to compute the wavelet-domain $G\Gamma KLD$ change detector was discussed in Section 4.3. First, each $n_{WS} \times n_{WS}$ sliding window at each pixel is decomposed in L scales and 4 orientations using an undecimated wavelet transform with a Daubechies filter bank (db2). As discussed earlier for the proposed DNTKLD, the computation of the wavelet transform for one sliding window requires $42L \cdot n_{WS}^2$ OPP. Secondly, for each subband, the parameters of $G\Gamma KLD$ are estimated using the Method of Log-Cumulants (MoLC), which requires $20n_{WS}^2 + 65$ OPP. Thirdly, the KL divergence of two subbands, one from each of the two acquired images, at the same level and orientation is computed based on Equation (4.14). From Equation (4.14), it can be seen that the KL calculation for two subbands requires 31 OPP; thus, the total computation for calculating all similarity measures for all subbands at each level and orientation is equal to $31 \cdot 4 \cdot L = 124L$ OPP. The final step is to sum up the subband-specific similarity maps over all subbands to obtain a final similarity map accoridng to Equation (5.7), which requires $4L$ OPP. Consequently, the total complexity of $G\Gamma KLD$ for a pixel is equal to $2 \cdot [42L \cdot n_{WS}^2 + 4L \cdot (20n_{WS}^2 + 65)] + 124L + 4L = 244L \cdot n_{WS}^2 + 648L$ OPP.

In summary, Table 6.1 tabulates the OPP for all the aforementioned change detectors.

6.4 Experimental Results

In order to assess the effectiveness of the proposed approach based on the DNT image representation, experiments were performed on the real multi-temporal SAR datasets described in Section 5.4.1. Two main experiments were carried out aimed at analyzing: 1) the effects of the sliding window sizes and the decomposition scales in the DNT domain; 2) the performance comparison with other existing change detectors. More details about experimental results are given below.

For performing the evaluation, both qualitative and quantitative measurements are used. For the qualitative measurement, the final binary change map generated was subjectively compared with the ground-truth image. For the quantitative measurement, the quality of the similarity map is evaluated using the receiver operating characteristic (ROC), while the quality of the final binary change detection map is assessed using the false alarm (FA), miss detection (MD), and total error (TE) as defined in Section 5.4.1 for comparing the computed change detection map against the ground-truth change detection. In addition, the computational complexity of the proposed algorithm is also analyzed. Note that the focus of this work is to generate accurate similarity maps

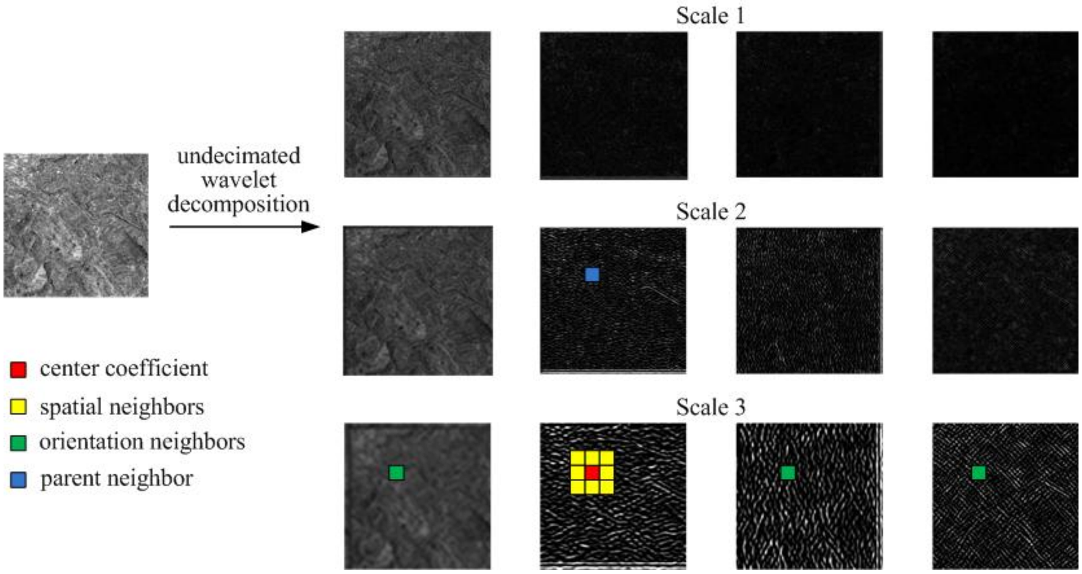


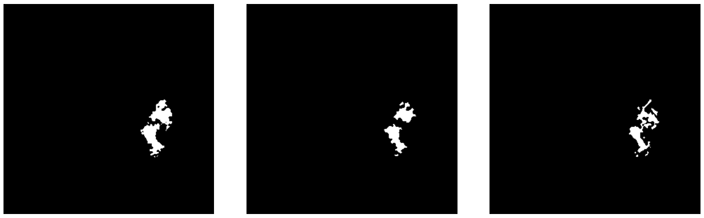
Figure 6.3: Illustration of Undecimated Decomposition And the Selection of DNT Neighbors. The Neighboring Coefficients Include the 3×3 Spatial Neighbors within the Same Subband, One Parent Neighboring Coefficient and Three Orientation Neighboring Coefficients.

rather than obtain binary change maps. Therefore, an optimal threshold corresponding to the point nearest to (0.0, 1.0) and lying on the ROC curve, is selected in order to obtain the best tradeoff between correct detection and false alarm.

In our implementation, to compute the DNT representation, each sliding window is decomposed into three scales ($L = (1, 2, 3)$) and four orientations using undecimated wavelet transform with a Daubechies filter bank (db2), as shown in Fig. 6.3. For each center coefficient of each subband, we define a DNT neighboring vector W that contains 13 coefficients, including nine from the same subband (including the center coefficient itself), one from the parent band, and three from the same spatial location in the other orientation bands at the same scale [152]. An illustration for center coefficient at scale 3 is given in Fig. 6.3. Similarly, for each center coefficient of each subband at scale 1 and 2, the DNT neighboring vector also contains 13 coefficients. Note that the parent band for subands at scale 1 is the original sliding window in the spatial image domain.

6.4.1 The Effects of Scale and Window Size in the DNT Domain

The proposed DNT-domain change detection algorithm decomposes a local neighborhood at different scales to calculate the similarity of the two images. In order to show the effect of the size of the



(a) scale=1 (b) scale=2 (c) scale=3

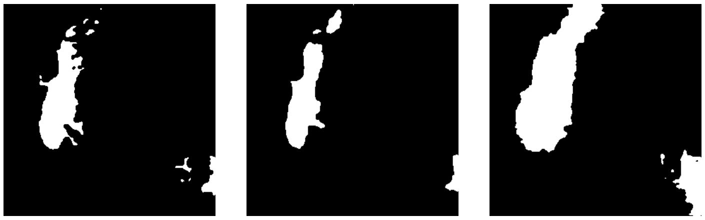
Figure 6.4: Change Detection Results Obtained by the Proposed DNT-Domain Algorithm on the Bern SAR Image Dataset at Different Scales with A Window Size of 5×5.

(a) 5×5 (b) 15×15 (c) 33×33

Figure 6.5: Change Detection Results Obtained by the Proposed DNT-Domain Algorithm on the Bern SAR Image Dataset at the Same Scale (Scale = 3) But with Different Window Sizes.

(a) scale=1 (b) scale=2 (c) scale=3

Figure 6.6: Change Detection Results Obtained by the Proposed DNT-Domain Algorithm on the Bangladesh SAR Image Dataset at Different Scales with A Window Size of 5×5.



(a) 5×5 (b) 15×15 (c) 33×33

Figure 6.7: Change Detection Results Obtained by the Proposed DNT-Domain Algorithm on the Bangladesh SAR Image Dataset at the Same Scale (Scale = 3) But with Different Window Sizes.

local neighborhood and the decomposition scales on the performance of the proposed DNT-domain change detection algorithm, different scales ($L = (1,2,3)$) and window sizes (ranging from 3×3

to 33 × 33) were used to compute the similarity map and the final binary change map. To obtain a binary change map, the optimal threshold is selected by using the ROC curve of the similarity map.

Figs. 6.4 to 6.7 show the obtained change detection maps using the Bern (Figs. 6.4 & 6.5) and the Bangladesh (Figs. 6.6 & 6.7) datasets for different scales and window sizes. From Figs. 6.4 and 6.6, it can be seen that, for the same window size (5×5), the change detection maps obtained at scale 3 has slightly better performance than those obtained at scale 1 and 2. Besides, Figs. 6.5 and 6.7 show that, at the same scale (scale = 3), the proposed wavelet-domain change detector with window size 5×5 yields better change detection results than with other window sizes (15×15 and 33×33).

Tables 6.2 and 6.3 show the obtained false alarm rate (FA), miss detection rate (MD), and total error (TE) values of the change detection results at different scales and for different window sizes, for the Bern and Bangladesh SAR image datasets, respectively. As illustrated in Tables 6.2 and 6.3, for a given window size, the performance in terms of FA, MD, and TE, dose not vary significantly with a change of scale. This is in contrast to the previously proposed wavelet-domain GMM-based change detection algorithm (Chapter 5), for which it is observed that the total error rate decreases as the number of scales increases for a given window size. In addition, from Tables 6.2 and 6.3, it can be seen that, at each scale, the lowest TE is obtained at a window size of 5×5.

6.4.2 Comparison with Other Existing Change Detectors in the Spatial and Wavelet Domains

In this experiment, the effectiveness of the proposed change detection algorithm is assessed by comparing its performance with other methods in terms of the similarity map, the binary change detection map and the computational complexity. Existing methods in both the spatial and wavelet domains are selected for comparison. In the spatial domain, the classical mean ratio detector in (MRD) [78], the recently proposed cumulant-based KL detector (CKLD) in [54] and the proposed spatial-domain GMM-based KL detectors (SGMMKLD) in Chapter 5 are selected for comparison. In the wavelet domain, the recent generalized Gamma KL detectors (GΓKLD) in [34] and the proposed wavelet-domain GMM-based KL detectors (WGMMKLD) in Chapter 5 are selected for comparison.

The qualities of the similarity maps obtained by those existing methods and the proposed DNT-domain algorithm (abbreviated as DNTKLD) are accessed in terms of the ROC curves. The corresponding area under the ROC curve (AUC) for each change detector is alos comput-

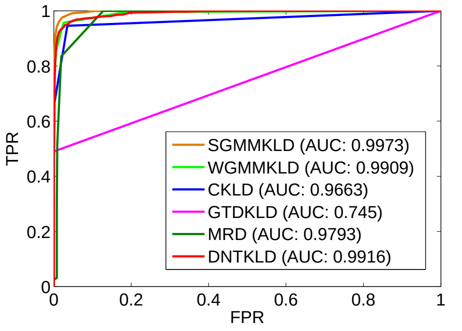
Table 6.2: Performance Results of the Proposed Change Detection Algorithm on Bern SAR Image Dataset in the DNT Domain with Different Scales And Window Sizes.

	Scale 1			Scale 2			Scale 3		
	FA	MD	TE	FA	MD	TE	FA	MD	TE
3×3	0%	56.88%	0.73%	0%	59.74%	0.76%	0%	63.38%	0.81%
5×5	0.19%	33.59%	0.62%	0.03%	45.63%	0.62%	0.01%	38.35%	0.59%
7×7	0.57%	26.67%	0.90%	1.11%	9.96%	1.22%	0.29%	48.23%	0.77%
9×9	0.63%	34.46%	1.06%	0.37%	42.6%	0.91%	0.21%	40.26%	0.72%
11×11	0.69%	33.85%	1.11%	0.54%	38.7%	1.03%	0.36%	39.31%	0.86%
15×15	1.33%	15.76%	1.51%	1.1%	18.35%	1.32%	0.8%	22.25%	1.07%
19×19	0.54%	49.52%	1.17%	0.63%	43.46%	1.18%	0.68%	39.39%	1.17%
21×21	0.46%	61.56%	1.24%	0.44%	60.78%	1.21%	0.82%	40.09%	1.32%
23×23	0.80%	45.71%	1.37%	1.66%	11.86%	1.79%	1.0%	38.61%	1.48%
25×25	0.28%	84.42%	1.35%	0.64%	53.85%	1.32%	1.21%	28.92%	1.56%
29×29	0.43%	78.10%	1.42%	1.31%	29%	1.66%	1.13%	29.78%	1.49%
33×33	0.86%	53.07%	1.53%	1.78%	10.48%	1.89%	0.1%	40.87%	1.50%

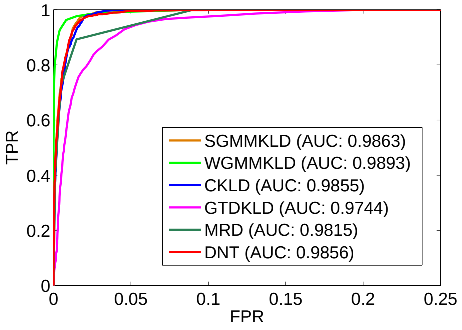
Table 6.3: Performance Results of the Proposed Change Detection Algorithm on the Bangladesh SAR Image Dataset in the DNT Domain with Different Scale And Window Size.

	Scale 1			Scale 2			Scale 3		
	FA	MD	TE	FA	MD	TE	FA	MD	TE
3×3	0%	38.28%	4.66%	0%	38.56%	4.70%	0%	39.69%	4.84%
5×5	0%	24.41%	2.97%	0%	23.52%	2.87%	0%	22.55%	2.75%
7×7	0%	27.27%	3.32%	0%	28.67%	3.49%	0%	27.93%	3.4%
9×9	0%	31.08%	3.79%	0%	30.46%	3.71%	0%	28.36%	3.45%
11×11	0%	33.81%	4.12%	0%	33.5%	4.08%	0%	31.46%	3.83%
15×15	0%	33.37%	4.08%	0%	34.27%	4.17%	0%	34.59%	4.22%
19×19	0.04%	32.79%	4.32%	0.07%	47.81%	5.90%	0.08%	36.11%	4.91%
21×21	0.5%	31.82%	4.31%	2.02%	18.56%	4.04%	0.25%	34.91%	4.47%
23×23	0.56%	32.15%	4.41%	0.54%	32.07%	4.39%	2.01%	15.64%	3.67%
25×25	1.19%	26.84%	4.32%	0.73%	30.45%	4.35%	2.22%	20.02%	4.39%
29×29	2.06%	22.33%	4.53%	1.36%	27.08%	4.50%	3.75%	14.67%	5.08%
33×33	1.45%	31.93%	5.17%	2.07%	25.89%	4.98%	4.18%	16.66%	5.7%

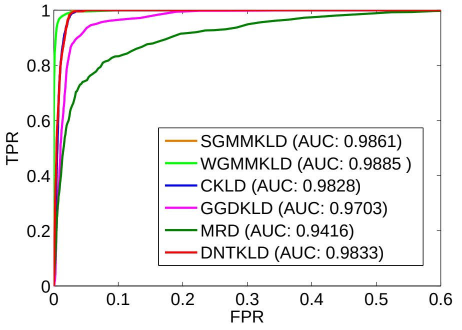
ed. Figs. 6.8 and 6.9, for the Bern and Bangladesh image datasets, respectively, show the ROC plots corresponding to the similarity maps obtained using the proposed DNTKLD change detector and the aforementioned existing change detector for different window sizes (small, medium, and large). In particular, for the wavelet-domain change detectors (GΓKLD, WGMMKLD), the results in Figs. 6.8 and 6.9 are obtained at decomposition level 3 (scale = 3); while the results obtained by the proposed DNTKLD change detector are obtained at decomposition level 1 (scale = 1).



(a) 5×5

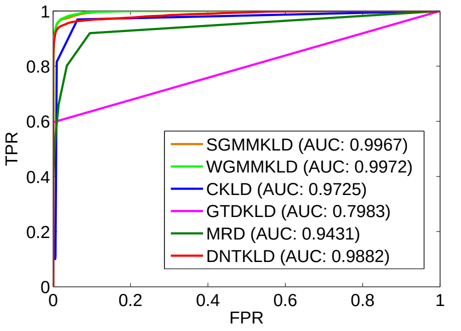


(b) 15×15

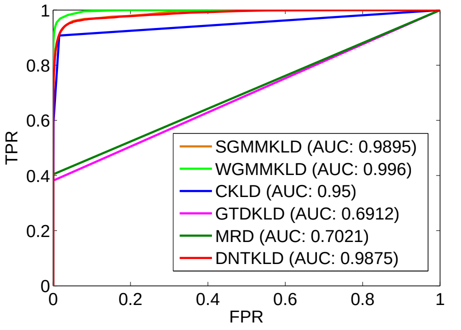


(c) 33×33

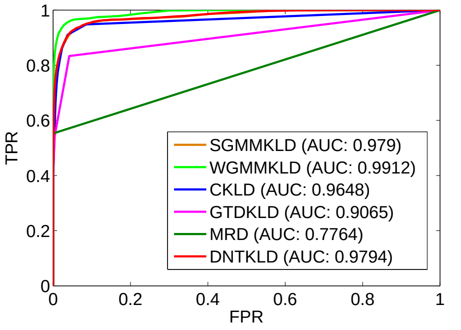
Figure 6.8: ROC Plots Comparison between G-MMKLD [160], CKLD [54], GΓDKLD [34], MRD [78], And the Proposed DNTKLD with Different Window Sizes for the Bern SAR Image Dataset.



(a) 5×5



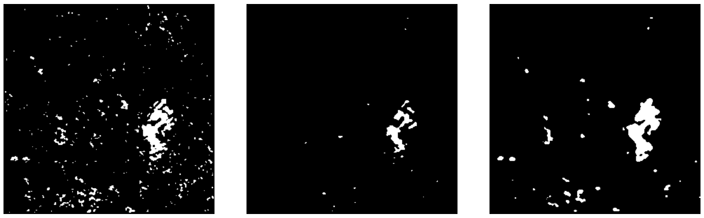
(b) 15×15



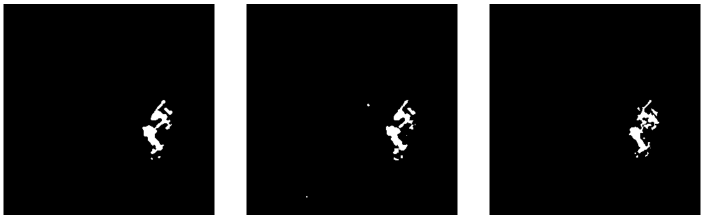
(c) 33×33

Figure 6.9: ROC Plots Comparison between G-MMKLD [160], CKLD [54], GΓDKLD [34], MRD [78], And the Proposed DNTKLD with Different Window Sizes for the Bangladesh SAR Image Dataset.

For the Bern dataset, which corresponds to a small change detection area, Fig. 6.8 shows that the proposed DNTKLD outperforms MRD, GΓKLD and has a comparable performance to SG-MMKLD and CKLD for all considered window sizes. From Fig. 6.8, it can also be seen that our previously proposed WGMMKLD performs slightly better than the proposed DNTKLD but this



(a) MRD [78] (b) CKLD [54] (c) GΓDKLD [34]



(d) SGMMKLD (e) WGMMKLD (f) DNTKLD

Figure 6.10: Change Detection Results from Existing Algorithms And Our Proposed DNTKLD for the Bern SAR Image Dataset with A Window Size of 5×5.

Table 6.4: FD, MD, and TE of the Binary Change Maps (as Shown in Fig. 6.10) Resulting from Different Change Detection Methods on the Bern SAR Image Dataset.

	False detections	Missed detections	Total errors
MRD [78]	3%	2.68%	3%
CKLD [54]	0.21%	33.25%	0.63%
GΓDKLD [34]	1.14%	23.72%	1.43%
SGMMKLD	0.14%	21.04%	0.41%
WGMMKLD	0.15%	25.02%	0.49%
DNTKLD	0.01%	38.35%	0.59%

comes at the expense of higher computational complexity as discussed later in this section. For the Bangadesh SAR image dataset, which corresponds to a relatively large change area, Fig. 6.9 shows that the proposed DNTKLD achieves a better performance as compared to CKLD, MRD and GΓKLD and exhibits a performance similar to SGMMKLD. Furthermore, form Fig. 6.9, it can be seen that, as in the Bern case, the WGMMKLD proposed in Chapter 5 slightly outperforms the proposed DNTKLD. This is due to the fact that the divisive normalization transform removes high-order statistical dependencies between neighboring wavelet coefficients and result in the loss of small variation.

From Figs. 6.8 and 6.9, it can be seen that the small window size 5×5 achieves better performance as compared to larger window sizes while achieving lower computational complexity. For

(a) MRD [78] (b) CKLD [54] (c) GΓDKLD [34]

(d) SGMMKLD (e) WGMMKLD (f) DNTKLD

Figure 6.11: Change Detection Results from Existing Algorithms And Our Proposed DNTKLD for the Bangladesh SAR Image Dataset with A Window Size of 5×5.

Table 6.5: FD, MD, and TE of the Binary Change Maps (as Shown in Fig. 6.11) Resulting from Different Change Detection Methods on the Bangladesh SAR Image Dataset.

	False detections	Missed detections	Total errors
MRD [78]	3.55%	4.32%	3.64%
CKLD [54]	0.32%	22.56%	2.95%
GΓDKLD [34]	0%	49.98%	6.09%
SGMMKLD	0.22%	13.11%	1.79%
WGMMKLD	0.5%	8.61%	1.49%
DNTKLD	0%	22.55%	2.75%

this purpose, the change similarity maps that were using a window size of 5×5 are used to compute the final binary maps. The change detection maps obtained using different change detectors are shown in Figs.6.10 and 6.11 for the Bern and Bangladesh datasets, respectively. Tables 6.4 and 6.5 show the corresponding FA, MD and TE performance measures for Bern and Bangladesh datasets, respectively. Tables 6.4 and 6.5 show that the proposed DNTKLD detector produces the relatively low total error rate of 0.59% and 2.75%, respectively.

However, the computation of the proposed DNTKLD is much less expensive than most of the existing change detectors. The computational complexity of each change detector is shown in Table 6.6 according to the analysis in Section 6.3.2. In our implementation, the DNTKLD change detector is computed at scale 1 ($L = 1$ and $M = 4$) and $N_{DNT} = 1$ while the WGMMKLD

Table 6.6: Operations Per Pixel (OPP) and Runtime of Proposed DNTKLD, SGMMKLD, and WGMMKLD Change Detectors and Other Existing Change Detectors for An $n_{Ws} \times n_{Ws}$ Window.

		OPP	Time (s)
Spatial-domain	MRD [78]	$2n_{WS}^2 + 4$	0.024
	CKLD [54]	$16n_{WS}^2 + 259$	205
	SGMMKLD	$640n_{WS}^2 + 2180$	1430
Wavelet-domain	GΓDKLD [34]	$732n_{WS}^2 + 1944$	1865
	WGMMKLD	$7932n_{WS}^2 + 26172$	20160
DNT-domain	DNTKLD	$332n_{WS}^2 + 32$	323

and GΓKLD are computed at scale 3 ($L = 3$ and $M = 4$). To estimate the GMM parameters, the initial mixture is set to 4 and the number of iterations is typically around 10 ($N_{loop} = 10$ and $K_{max} = 4$). In addition, Table 6.6 also shows the runtime of each change detector. Note that all the algorithms are implemented on a Pentium IV 2.4 GHz CPU computer with Matlab 2012b without any optimization. As shown in Table 6.6, the first-order MRD is a very simple change detector and thus its computational complexity is the lowest among all change detectors. In contrast, the change detectors based on the higher order statistics, which we refer to as high-order change detectors, such as CKLD, GΓKLD, SGMMKLD and WGMMKLD have significantly higher computational complexity since the estimation of their parameters involves a large number of iterations in order to obtain an accurate convergence. It is also shown in Table 6.6 that the proposed DNTKLD change detector results in significantly less computations than the high-order change detectors with the exception of CKLD due to the fact that the distribution of the coefficients after DNT can be well fitted using a single-parameter Gaussian model, which is easy to estimate and which results in a simple closed-form expression for the KL-divergence between two Gaussian models. The proposed wavelet-domain DNTKLD exhibits similar computational complexity as the spatial-domain CKLD but results in lower FA and TE as compared to CKLD.

In summary, the high-order change detectors exhibit relatively good detection performance at the expense of high computation cost, while the low-computation first-order MRD detector achieves a significantly lower detection performance. The proposed DNTKLD detector has lower computational complexity as compared to the high-order change detectors without sacrificing the detection performance. In fact, the proposed DNTKLD was shown to outperform or achieve a comparable performance to the higher-complexity change detectors, except for the WGMMKLD which slightly outperforms the DNTKLD but at the expense of significantly higher computational complexity.

6.5 Summary

In this chapter, we proposed a novel change detection algorithm using statistical features of the divisive normalization-based image representation. Compared to existing detectors with higher-order statistics, the proposed method exhibits lower computational complexity with comparable change detection performance to the high-complexity change detectors.

Chapter 7

CONCLUSION

This research work contributes to the area of image analysis with a focus on spatial change detection for natural images and multi-temporal change detection for SAR imagery. This chapter summarizes the major contributions of this work and presents possible future extensions.

7.1 Contributions

This work presents a robust and noise resilient distributed Canny edge detection algorithm. In addition, this work also proposes new spatial-domain and wavelet-domain statistical similarity measures with applications in multi-temporal SAR change detection. The main contributions of the presented work can be summarized as follows:

- Distributed Canny edge detection algorithm

  The original Canny algorithm relies on frame-level statistics to predict the high and low thresholds for hysterisis thresholding. This puts large constraints on memory requirements and is not suitable for popular systems that perform block-based image processing. Furthermore, the frame-level Canny results in a large latency that is proportional to the image size. In order to resolve these issues, a novel distributed Canny edge detection algorithm that can compute the high and low thresholds for each block based on the block type and the local distribution of pixel gradients in the block, is presented. Each block can be processed independently, which decreases the required memory resources and enables efficient pipelining with block-based image processing methods. Alternatively, the new distributed Canny edge detector enables the blocks to be processed simultaneously, thus reducing the latency significantly, which is important for meeting real-time requirements. More importantly, conducted conformance evaluations and subjective tests show that, compared with the frame-based Canny edge detector, the proposed algorithm yields better edge detection results for both clean and noisy images. The distributed Canny detection algorithm is mapped onto a Xilinx Virtex-5 FPGA platform. The synthesized results show 64% slice utilization and 87% BRAM memory utilization. The FPGA implementation of the proposed Canny edge detector takes only 0.721ms (including the SRAM read/write time and the computation time) to detect edges of 512×512 images in the USC SIPI database when clocked at 100 MHz. Thus the FPGA-based implementation shows that the proposed distributed Canny edge detector is capable of supporting fast real-time edge detection of images and videos including those with full-HD content.

- Modeling the local statistics of SAR images by using a Gaussian Mixture Model
  Existing methods for modeling SAR images are mainly suitable for SAR images containing homogenous regions, even though each homogenous region shows different land-cover typologies. However, in general, actual SAR images correspond to a varied scene presenting several different regions with different characteristics. To address this problem, the locally adaptive Gaussian Mixture Model is shown as part of this work to be more suitable and flexible than the single parametric mathematical distribution models to approximate the local distribution of the SAR image. Experiments on real SAR image data shows that the GMM model fits the local distribution of the SAR image better than the existing sophisticated single parametric mathematical distribution models.

- Change detection in multi-temporal SAR images based on a proposed GMM-based similarity measure in the spatial domain
  A novel change detector for the analysis of multi-temporal SAR images is proposed in the spatial domain. This change detector is based on the analysis of the similarity of local distributions of SAR images using a Gaussian Mixture Model and the KL distance. The effect of the size of the local neighborhood window on the performance of the proposed algorithm in the spatial domain is evaluated based on the analysis of real SAR image datasets. Compared with existing change detectors with higher order statistics in the spatial domain, the proposed spatial-domain GMM-based change detector has better change detection performance in terms of the quality of the similarity maps, which are assessed using the receiver operating characteristic (ROC) curves, and in terms of the total error rates of the final change detection maps.

- Change detection in multi-temporal SAR images based on the proposed GMM-based similarity measure in the wavelet domain

A novel change detection algorithm for multi-temporal SAR images is proposed in the wavelet domain based on the aforementioned proposed GMM-based image similarity measure. There are two benefits to extend our framework to the wavelet domain. First, each wavelet scale is characterized by a different tradeoff between speckle reduction and preservation of geometrical details. Thus, the geometrical details can be preserved well while the speckle noise can be largely reduced in the wavelet domain. Secondly, texture can be easily represented and discriminated in the wavelet domain. The effect of the wavelet scales and window sizes on the performance of the proposed algorithm in the wavelet domain is analyzed. Tests on real SAR image datasets show that the proposed wavelet-domain change detector outperforms existing wavelet-domain change detectors in terms of the quality of the similarity maps, which are assessed using the receiver operating characteristic (ROC) curves, and in terms of the total error rates of the final change detection maps.

- Change detection based on the divisive normalization transform
  A novel change detection algorithm based on a divisive normalization image representation is presented, which reduces the computational complexity compared to existing wavelet-domain model-based change detectors. The DNT was empirically shown to produce approximately Gaussian marginal distributions and to reduce the statistical dependencies of the original wavelet-domain representation, and results in a one-parameter function that allows efficient change detection. Tests on real data show that the proposed DNT-based detector has competitive detection performance with the proposed GMM-based change detectors while achieving lower computational complexity as compared to previously suggested methods.

7.2 Future Work

The work presented herein can be extended and optimized for different applications. Future possible directions of the presented work include the following:

- Possible improvements to the proposed distributed Canny edge detection algorithm include the use of alternative block classification methods to improve the performance, and developing an automatic filter size selection scheme to adaptively choose the optimal gradient filter size according to the image characteristics.

- Hardware implementation of the distributed Canny edge detector

  Future directions in the hardware implementation of the distributed Canny edge detector include mapping the block-based algorithm to GPU-based hardware platforms or other multi-core hardware platforms.

- Not limited to the application of the change detection in multi-temporal SAR images, the proposed GMM-based image similarity can be extended to other applications, such as image quality assessment, texture retrieval, etc.

- Possible future directions in the area of change detection in multi-temporal SAR images include extensions of the current Gaussian mixture model to other types of mixture models such as Rayleigh mixture model, generalized Gaussian mixture model and generalized Gamma mixture model. Other further work include considering other similarity metrics besides the KL divergence.

REFERENCES

[1] L. Ganesan and P. Bhattacharyya, “Edge detection in untextured and textured images - a common computational framework,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 5, pp. 823–834, 1997.

[2] V. V. Starovoitov, S.-Y. Jeong, and R.-H. Park, “Texture periodicity detection: Features, properties, and comparisons,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 28, no. 6, pp. 839–849, 1998.

[3] B. Bhanu and J. Peng, “Adaptive integrated image segmentation and object recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 30, no. 4, pp. 427–441, 2000.

[4] C. Munteanu and A. Rosa, “Gray-scale image enhancement as an automatic process driven by evolution,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 2, pp. 1292–1298, 2004.

[5] A. Bonci, T. Leo, and S. Longhi, “A Bayesian approach to the Hough transform for line detection,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 35, no. 6, pp. 945–955, 2005.

[6] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[7] C. Li, C. Xu, C. Gui, and M. D. Fox, “Level set evolution without re-initialization: a new variational formulation,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 430–436.

[8] Z. Ying, L. Guangyao, S. Xiehua, and Z. Xinmin, “Geometric active contours without re-initialization for image segmentation,” *Elsevier, Pattern Recognition*, vol. 42, no. 9, pp. 1970–1976, 2009.

[9] M. Bennamoun and B. Boashash, “A structural-description-based vision system for automatic object recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 6, pp. 893–906, 1997.

[10] E. Danahy, K. Panetta, and S. Agaian, “Feature extraction system for contextual classification within security imaging applications,” *IEEE International Conference on System of Systems Engineering*, pp. 1–6, 2007.

[11] J. R. Parker, *Algorithms for image processing and computer vision*. New York: John Wiley & Sons, 2010.

[12] K. A. Panetta, E. J. Wharton, and S. S. Agaian, “Human visual system-based image enhancement and logarithmic contrast measure,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 1, pp. 174–188, 2008.

[13] E. J. Wharton, K. Panetta, and S. S. Agaian, “Logarithmic edge detection with applications,” *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3346–3351, 2007.

[14] M. Yagi, M. Adachi, and T. Shibata, “A hardware-friendly soft-computing algorithm for image recognition,” *Proceedings of European Signal Processing Conference*, pp. 729–732, 2000.

[15] X. S. Zhou and T. S. Huang, “Edge-based structural features for content-based image retrieval,” *Pattern Recognition Letters*, vol. 22, no. 5, pp. 457–468, 2001.

[16] M. Bulacu, L. Schomaker, and L. Vuurpijl, “Writer identification using edge-based directional features,” in *IEEE International Conference on Document Analysis and Recognition*, vol. 2, Aug. 2003, pp. 937–937.

[17] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of the human body,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[18] L. Lee, R. Romano, and G. Stein, “Monitoring activities from multiple video streams: establishing a common coordinate frame,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 758–767, 2000.

[19] C. Stauffer and W. E. L. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[20] J. B. Collins and C. E. Woodcock, “An assessment of several linear change detection techniques for mapping forest mortality using multitemporal Landsat TM data,” *Remote sensing of Environment*, vol. 56, no. 1, pp. 66–77, 1996.

[21] A. Huertas and R. Nevatia, “Detecting changes in aerial views of man-made structures,” *Image and Vision Computing*, vol. 18, no. 8, pp. 583–596, 2000.

[22] L. Bruzzone and D. F. Prieto, “An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images,” *IEEE Transations on Image Processing*, vol. 11, no. 4, pp. 452–466, Apr. 2002.

[23] J. P. Thirion and G. Calmon, "Deformation analysis to detect and quantify active lesions in three-dimensional medical image sequences," *IEEE Transactions on Medical Imaging*, vol. 18, no. 5, pp. 429–441, 1999.

[24] D. Rey, G. Subsol, H. Delingette, and N. Ayache, "Automatic detection and segmentation of evolving processes in 3D medical images: Application to multiple sclerosis," *Medical Image Analysis*, vol. 6, no. 2, pp. 163–179, 2002.

[25] M. Bosc, F. Heitz, J. P. Armspach, I. Namer, D. Gounot, and L. Rumbach, "Automatic change detection in multimodal serial MRI: application to multiple sclerosis lesion evolution," *NeuroImage*, vol. 20, no. 2, pp. 643–656, 2003.

[26] E. N. Landis, E. N. Nagy, D. T. Keane, and G. Nagy, "Technique to measure 3D work-of-fracture of concrete in compression," *Journal of Engineering Mechanics*, vol. 125, no. 6, pp. 599–605, 1999.

[27] G. Nagy, T. Zhang, W. Franklin, E. Landis, E. Nagy, and D. T. Keane, "Volume and surface area distributions of cracks in concrete," in *Visual Form*, 2001, pp. 759–768.

[28] J. Whorff and L. Griffing, "A video recording and analysis system used to sample intertidal communities," *Journal of Experimental Marine Biology and Ecology*, vol. 160, no. 1, pp. 1–12, 1992.

[29] K. Lebart, E. Trucco, and D. Lane, "Real-time automatic sea-floor change detection from video," in *MTS/IEEE Conference and Exhibition on Oceans*, vol. 2, 2000, pp. 1337–1343.

[30] D. R. Edgington, K. A. Salamy, M. Risi, R. Sherlock, D. Walther, and C. Koch, "Automated event detection in underwater video," in *MTS/IEEE Conference and Exhibition on Oceans*, vol. 5, 2003, pp. P2749–P2753.

[31] W. Y. Kan, J. V. Krogmeier, and P. C. Doerschuk, "Model-based vehicle tracking from image sequences with an application to road surveillance," *Optical Engineering*, vol. 35, no. 6, pp. 1723–1729, 1996.

[32] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, "Automatic change detection of driving environments in a vision-based driver assistance system," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 646–657, 2003.

[33] Y. Bazi, L. Bruzzone, and F. Melgani, "An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 874–887, Apr. 2005.

[34] S. Cui and M. Datcu, "Statistical wavelet subband modeling for multi-temporal SAR change detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 4, pp. 1095–1109, 2012.

[35] L. G. Roberts, "Machine perception of three-dimensional soups," Ph.D. dissertation, Massachusetts Institute of Technology, 1963.

[36] J. M. Prewitt, "Object enhancement and extraction," *Picture Processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[37] J. Kittler, "On the accuracy of the Sobel edge detector," *Image and Vision Computing*, vol. 1, no. 1, pp. 37–42, 1983.

[38] W. Frei and C. Chen, "Fast boundary detection: A generalization and a new algorithm," *IEEE Transactions on Computers*, vol. 26, no. 10, pp. 988–998, 1977.

[39] J. Canny, "A computation approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 769–798, 1986.

[40] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1485–1490, 2005.

[41] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[42] R. M. Haralick, "Digital step edges from zero crossing of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 58–68, 1984.

[43] R. Deriche, "Using Canny criteria to derive a recursively implemented optimal edge detector," *International Journal of Computer Vision*, pp. 167–187, 1987.

[44] C. Y. Lee, F. Catthoor, and H. De Man, "An efficient ASIC architecture for real-time edge detection," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 10, pp. 1350–1359, 1989.

[45] L. Torres, M. Robert, E. Bourennane, and M. Paindavoine, "Implementation of a recursive real time edge detector using retiming technique," in *Proceedings of the Asia and South Pacific IFIP International Conference on Very Large Scale Integration*, 1995, pp. 811–816.

[46] F. G. Lorca, L. Kessal, and D. Demigny, "Efficient ASIC and FPGA implementation of IIR filters for real time edge detection," in *IEEE International Conference on Image Processing (ICIP-97)*, vol. 2, 1997, pp. 406–409.

[47] D. V. Rao and M. Venkatesan, "An efficient reconfigurable architecture and implementation of edge detection algorithm using Handle-C," in *IEEE Conference on Information Technology: Coding and Computing (ITCC)*, vol. 2, Apr. 2004, p. 843 C 847.

[48] H. Neoh and A. Hazanchuck, "Adaptive edge detection for real-time video processing using FPGAs," *Application notes, Altera Corporation*, 2005.

[49] C. Gentsos, C. Sotiropoulou, S. Nikolaidis, and N. Vassiliadis, "Real-time Canny edge detection parallel implementation for FPGAs," in *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Dec. 2010, pp. 499–502.

[50] Y. Luo and R. Duraiswami, "Canny edge detection on NVIDIA CUDA," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 08)*, 2008, pp. 1–8.

[51] R. Palomar, J. M. Palomares, J. M. Castillo, J. Olivares, and J. Gomez-Luna, "Parallelizing and optimizing Lip-Canny using NVIDIA CUDA," in *Trends in Applied Intelligent Systems*, 2010, pp. 389–398.

[52] L. Lourenco, "Efficient implementation of Canny edge detection filter for ITK using CUDA ," in *13th Symposium on Computer Systems*, 2012, pp. 33–40.

[53] G. Jianya, S. Haigang, M. Guorui, and Z. Qiming, "A review of multi-temporal remote sensing data change detection algorithms," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, no. B7, pp. 757–762, 2008.

[54] J. Inglada and G. Mercier, "A new statistical similarity measure for change detection in multitemporal SAR images and its extension to multiscale change analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1432 –1445, May 2007.

[55] Y. Y. Tang, L. Yang, and J. Liu, "Characterization of dirac-structure edges with wavelet transform," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 30, no. 1, pp. 93–109, 2000.

[56] Q. Hu, X. He, and J. Zhou, "Multi-scale edge detection with bilateral filtering in spiral architecture," in *Proceedings of the Pan-Sydney Area Workshop On Visual Information Processing*, 2004, pp. 29–32.

[57] R. Conzalez, R. Woods, and S. Eddins, *Digital Image Processing Using MATLAB*. Prentice Hall, Upper Saddle River, 2^{nd} Edition, 2002.

[58] P. H. Eichel and E. J. Delp, "Quantitative analysis of a moment-based edge operator," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 20, no. 1, pp. 58–66, 1990.

[59] A. Koschan and M. Abidi, "Detection and classification of edges in color images," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 64–73, 2005.

[60] E. E. Danahy, S. S. Agaian, and K. A. Panetta, "Directional edge detection using the logical transform for binary and grayscale images," in *Proceedings of SPIE Defense and Security Symposium*, 2006, pp. 1–12.

[61] S. Nercessian, "A new class of edge detection algorithms with performance measure," *Master thesis, Tufts University*, May 2009.

[62] J. Li, "A wavelet approach to edge detection," Ph.D. dissertation, Sam Houston State University, 2003.

[63] X. Wang, "Laplacian operator-based edge detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 886–890, 2007.

[64] W. He and K. Yuan, "An improved Canny edge detector and its realization on FPGA," in *IEEE Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA)*, June 2008, pp. 6561–6564.

[65] I. K. Park, N. Singhal, M. H. Lee, S. Cho, and C. W. Kim, "Design and performance evaluation of image processing algorithms on GPUs," *IEEE Transactions On Parallel And Distributed Systems*, vol. 23, no. 1, pp. 91–104, 2011.

[66] J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krger, A. Lefohn, and T. Purcell, "A survey of general-purpose computation on graphics hardware," in *Computer Graphics Forum*, vol. 26, no. 1, 2007, pp. 80–113.

[67] J. Su and R. Mersereau, "Post-processing for artifact reduction in JPEG-compressed images," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, 1995, pp. 2363–2366.

[68] P. Arbelaez, C. Fowlkes, and D. Martin, "The Berkeley segmentation dataset and benchmark," http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

[69] Q. Xu, C. Chakrabarti, and L. J. Karam, "A distributed canny edge detector and its implementation on FPGA," in *IEEE Digital Signal Processing Workshop (DSP)*, 2011, pp. 500–505.

[70] N. D. Narvekar and L. J. Karam, "A no-reference image blur metric based on the cumulative probability of blur detection (CPBD)," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2678–2683, 2011.

[71] "USC SIPI image database," http://sipi.usc.edu/database/database.php?volume=misc.

[72] "Standard test image database," http://www.imageprocessingplace.com/.

[73] L. Scharff, A. Hill, and A. Ahumada, "Discriminability measures for predicting readability of text on textured backgrounds," *Optics Express*, vol. 6, no. 4, pp. 81–91, Feb. 2000.

[74] "Altera available IP," www.altera.com/products/ip/processors/nios2/features/ni2-peripherals.html.

[75] "Xilinx available IP," www.xilinx.com/ise/embedded/edk_ip.htm.

[76] "Xilinx Vertrix-5 Family Datasheet," http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf.

[77] "SRAM Datasheet," http://www.cypress.com/?docID=31696.

[78] E. J. Rignot and J. J. van Zyl, "Change detection techniques for ERS-1 SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31, no. 4, pp. 896–906, July 1993.

[79] B. P. Salmon, J. C. Olivier, K. J. Wessels, W. Kleynhans, F. van den Bergh, and K. C. Steenkamp, "Unsupervised land cover change detection: Meaningful sequential time series analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. (JSTARS)*, vol. 4, no. 2, pp. 327–335, 2011.

[80] K. S. Chen, H. W.Wang, C. T.Wang, and W. Y. Chang, "A study of decadal coastal changes on western Taiwan using a time series of ERS satellite SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. (JSTARS)*, vol. 4, no. 4, pp. 826–835, 2011.

[81] C. Huo, Z. Zhou, H. Lu, C. Pan, and K. Chen, "Fast object-level change detection for VHR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 1, pp. 118–122, 2010.

[82] F. Bovolo, L. Bruzzone, and M. Marconcini, “A novel approach to unsupervised change detection based on a semisupervised SVM and a similarity measure,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 2070–2082, 2008.

[83] F. Bovolo, G. Camps-Valls, and L. Bruzzone, “A support vector domain method for change detection in multitemporal images,” *Pattern Recognition Letters*, vol. 31, no. 10, pp. 1148–1154, 2010.

[84] A. Singh, “Digital change detection techniques using remotely-sensed data,” *International Journal of Remote Sensing*, vol. 10, no. 6, pp. 989–1003, 1989.

[85] F. G. Hall, D. B. Botkin, D. E. Strebel, K. D. Woods, and S. J. Goetz, “Large-scale patterns of forest succession as determined by remote sensing,” *Ecology*, pp. 628–640, 1991.

[86] B. Jeon and D. A. Landgrebe, “Classification with spatio-temporal interpixel class dependency contexts,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 4, pp. 663–672, 1992.

[87] A. Solberg, T. Taxt, and A. K. Jain, “A Markov random field model for classification of multisource satellite imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 1, pp. 100–113, 1996.

[88] L. Bruzzone and D. F. Prieto, “Unsupervised retraining of a maximum likelihood classifier for the analysis of multitemporal remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 456–460, 2001.

[89] L. Bruzzone and S. Serpico, “An iterative technique for the detection of land-cover transitions in multitemporal remote-sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 858–867, 1997.

[90] G. Mercier, S. Derrode, W. Pieczynski, J.-M. Nicolas, A. Joannic-Chardin, and J. Inglada, “Copula-based stochastic kernels for abrupt change detection,” in *IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2006.*, 2006, pp. 204–207.

[91] V. S. Frost, J. A. Stiles, K. S. Shanmugan, and J. Holtzman, “A model for radar images and its application to adaptive digital filtering of multiplicative noise,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 157–166, 1982.

[92] J.-S. Lee, “Digital image enhancement and noise filtering by use of local statistics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 165–168, 1980.

[93] D. T. Kuan, A. A. Sawchuk, T. C. Strand, and P. Chavel, “Adaptive noise smoothing filter for images with signal-dependent noise,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 165–177, 1985.

[94] A. Lopes, E. Nezry, R. Touzi, and H. Laur, “Structure detection and statistical adaptive speckle filtering in SAR images,” *International Journal of Remote Sensing*, vol. 14, no. 9, pp. 1735–1758, 1993.

[95] ——, “Maximum a posteriori speckle filtering and first order texture models in sar images,” in *IEEE International Geoscience and Remote Sensing Symposium*, 1990, pp. 2409–2412.

[96] S. Solbø and T. Eltoft, “λ-WMAP: a statistical speckle filter operating in the wavelet domain,” *International Journal of Remote Sensing*, vol. 25, no. 5, pp. 1019–1036, 2004.

[97] F. Bujor, E. Trouvé, E. Valet, J.-M. Nicolas, and J. Rudant, “Application of log-cumulants to the detection of spatiotemporal discontinuities in multitemporal SAR images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 10, pp. 2073–2084, Oct. 2004.

[98] L. Bruzzone and D. F. Prieto, “Automatic analysis of the difference image for unsupervised change detection,” *IEEE Transactions on Geoscience Remote Sensing*, vol. 38, no. 3, pp. 1171–1182, May 2000.

[99] Y. Bazi, L. Bruzzone, and F. Melgani, “Image thresholding based on the EM algorithm and the generalized Gaussian distribution,” *Pattern Recognition*, vol. 40, no. 2, pp. 619–634, 2007.

[100] T. Celik, “A Bayesian approach to unsupervised multiscale change detection in synthetic aperture radar images,” *Signal Processing*, vol. 90, no. 5, pp. 1471–1485, May 2010.

[101] ——, “Bayesian change detection based on spatial sampling and Gaussian mixture model,” *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1635–1642, 2011.

[102] V. Alberga, “Similarity measures of remotely sensed multi-sensor images for change detection applications,” *Remote Sensing*, vol. 1, no. 3, pp. 122–143, 2009.

[103] A. Winter, H. Maitre, N. Cambou, and E. Legrand, “Entropy and multiscale analysis: a new feature extraction algorithm for aerial images,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 1997, pp. 2765–2768.

[104] L. Gueguen and M. Datcu, “Mixed information measure: Application to change detection in earth observation,” in *MultiTemp 2009: 5h Int. Workshop Analysis of Multi-temporal Remote Sensing Images*, 2009.

[105] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[106] J. Inglada, "Change detection on SAR images by using a parametric estimation of the Kullback-Leibler divergence," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 6, 2003, pp. 4104–4106.

[107] Y. Delignon, R. Garello, and A. Hillion, "Statistical modelling of ocean SAR images," in *IEE Proceedings-Radar, Sonar and Navigation*, vol. 144, no. 6, 1997, pp. 348–354.

[108] N. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions*. New York: John Wiley & Sons, 1994.

[109] A. Stuart and J. Ord, *Kendalls advanced theory of statistics*. London: Arnold, 5th edition, 1991.

[110] P. McCullagh, *Tensor methods in statistics*. London: Chapman and Hall, 1987.

[111] J. Lin, N. Saito, and R. Levine, "Edgeworth approximation of the Kullback-Leibler distance towards problems in image analysis," *University of California, Davis, Technical Report*, 1999.

[112] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[113] K. Sharifi and A. Leon-Garcia, "Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video," *IEEE Transactions onCircuits and Systems for Video Technology*, vol. 5, no. 1, pp. 52–56, 1995.

[114] K. S. Song, "A globally convergent and consistent method for estimating the shape parameter of a generalized Gaussian distribution," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 510–527, 2006.

[115] G. Van de Wouwer, P. Scheunders, and D. V. Dyck, "Statistical texture characterization from discrete wavelet representations," *IEEE Transactions on Image Processing*, vol. 8, no. 4, pp. 592–598, 1999.

[116] K. S. Song, "Globally convergent algorithms for estimating generalized gamma distributions in fast signal and image processing," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1233–1250, 2008.

[117] S. K. Choy and C. S. Tong, “Statistical wavelet subband characterization based on generalized Gamma density and its application in texture retrieval,” *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 281–289, 2010.

[118] J. M. Nicolas, “Introduction to second kind statistics: application of log-moments and log-cumulants to SAR image law analysis,” *Traitement du Signal*, vol. 19, no. 3, pp. 139–168, 2002.

[119] F. T. Ulaby, F. Kouyate, B. Brisco, and T. H. L. Williams, “Textural information in SAR images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. GRS-24, no. 2, pp. 235 –245, Mar. 1986.

[120] F. Bovolo and L. Bruzzone, “An adaptive technique based on similarity measures for change detection in very high resolution SAR images,” in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, 2008, pp. 158–161.

[121] J. Goldberger, S. Gordon, and H. Greenspan, “An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures,” in *Proceedings of IEEE International Conference on Computer Vision*, vol. 1, 2003, pp. 487 –493.

[122] G. Moser, J. Zerubia, and S. Serpico, “SAR amplitude probability density function estimation based on a generalized Gaussian model,” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1429–1442, Jun. 2006.

[123] E. Parzen, “On estimation of a probability density function and mode,” *The Annals of Mathematical Statistics*, 1962.

[124] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford University Press, Inc., 1995.

[125] P. Mantero, G. Moser, and S. Serpico, “Partially supervised classification of remote sensing images through SVM-based probability density estimation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 559–570, Mar. 2005.

[126] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. New York: John Wiley & Sons, 2012.

[127] J. Jao, “Amplitude distribution of composite terrain radar clutter and the κ-distribution,” *IEEE Transactions on Antennas and Propagation*, vol. 32, no. 10, pp. 1049–1062, 1984.

[128] A. Frery, H. Muller, C. Yanasse, and S. Sant'Anna, "A model for extremely heterogeneous clutter," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 3, pp. 648–659, 1997.

[129] H.-C. Li, W. Hong, Y.-R. Wu, and P.-Z. Fan, "An efficient and flexible statistical model based on generalized Gamma distribution for amplitude SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 6, pp. 2711–2722, Jun. 2010.

[130] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images with CDROM*. Raleigh, NC: SciTech Publishing, 2004.

[131] C. Tison, J. Nicolas, F. Tupin, and H. Maître, "A new statistical model for Markovian classification of urban areas in high-resolution SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 10, pp. 2046–2057, 2004.

[132] H.-C. Li, W. Hong, Y.-R. Wu, and P.-Z. Fan, "On the empirical-statistical modeling of sar images with generalized Gamma distribution," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 386–397, 2011.

[133] J. W. Goodman, "Some fundamental properties of speckle," *J. Opt. Soc. Amer.*, vol. 66, no. 11, pp. 1145–1150, 1976.

[134] E. Kuruoglu and J.Zerubia, "Modeling SAR images with a generalizedtion of the Rayleigh distribution," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 527–533, Apr. 2004.

[135] E. W. Stacy and G. A. Mihram, "Parameter estimation for a generalized Gamma distribution," *Technometrics*, vol. 7, no. 3, pp. 349–358, 1965.

[136] D. R. Wingo, "Computing maximum-likelihood parameter estimates of the generalized Gamma distribution by numerical root isolation," *IEEE Transactions on Reliability*, vol. 36, no. 5, pp. 586–590, 1987.

[137] F. Ashkar, B. Bobée, D. Leroux, and D. Morisette, "The generalized method of moments as applied to the generalized Gamma distribution," *Stochastic Hydrology and Hydraulics*, vol. 2, no. 3, pp. 161–174, 1988.

[138] F. Bowman, *Introduction to Bessel functions*. DoverPublications.com, 1958.

[139] S. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian approaches to Gaussian mixture modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1133–1142, 1998.

[140] G. McLachlan and D. Peel, *Finite mixture models*. New York: John Wiley & Sons, 2004.

[141] J. Oliver, R. Baxter, and C. Wallace, "Unsupervised learning using MML," in *Proceedings of the Thirteenth International Conference Machine Learning (ICML)*, 1996, pp. 364–372.

[142] T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 155–176, 1996.

[143] M. Kloppenburg and P. Tavan, "Deterministic annealing for density estimation by multivariate normal mixtures," *Physical Review E*, vol. 55, no. 3, pp. R2089–R2092, 1997.

[144] M. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[145] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A component-wise EM algorithm for mixtures," *Journal of Computational and Graphical Statistics*, vol. 10, no. 4, 2001.

[146] J. R. Hershey and P. A. Olsen, "Approximating the Kullback-Leibler divergence between Gaussian mixture models," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2007, pp. 317 –320.

[147] F. Bovolo and L. Bruzzone, "A detail-preserving scale-driven approach to change detection in multitemporal SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 12, pp. 2963–2972, 2005.

[148] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001.

[149] O. Schwartz and E. P. Simoncelli, "Natural signal statistics and sensory gain control," *Nature Neuroscience*, vol. 4, no. 8, pp. 819–825, 2001.

[150] D. J. Heeger, "Normalization of cell responses in cat striate cortex," *Visual Neuroscience*, vol. 9, no. 02, pp. 181–197, 1992.

[151] E. P. Simoncelli and D. J. Heeger, "A model of neuronal responses in visual area MT," *Vision Research*, vol. 38, no. 5, pp. 743–761, 1998.

[152] Q. Li and Z. Wang, "Reduced-reference image quality assessment using divisive normalization-based image representation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 202–211, 2009.

[153] S. Lyu and E. P. Simoncelli, “Statistically and perceptually motivated nonlinear image representation,” in *SPIE Electronic Imaging Symposium*, vol. 6492, 2007, pp. 1–15.

[154] A. Rehman and Z. Wang, “Reduced-reference image quality assessment by structural similarity estimation,” *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3378–3389, 2012.

[155] J. Malo, I. Epifanio, R. Navarro, and E. P. Simoncelli, “Nonlinear image representation for efficient perceptual coding,” *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 68–80, 2006.

[156] M. J. Wainwright and E. P. Simoncelli, “Scale mixtures of Gaussians and the statistics of natural images,” in *Proceedings of Advances in Neural Information Processing Systems*. Citeseer, 1999, pp. 855–861.

[157] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, “Image denoising using scale mixtures of Gaussians in the wavelet domain,” *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

[158] J. Portilla and E. Simoncelli, “Image restoration using Gaussian scale mixtures in the wavelet domain,” in *IEEE International Conference on Image Processing (ICIP)*, vol. 2, 2003, pp. 965–968.

[159] “Intel 64 and IA-32 architectures software developer manuals,” http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html.

[160] Q. Xu and L. J. Karam, “Change detection on sar images by a parametric estimation of the kl-divergence between gaussian mixture models,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 2109–2113.