



**De La Salle University**

**Design and Development of a Computer Vision-based  
Breast Self-Examination Instruction and Supervision System**

A Thesis  
Presented to the  
Department of Electronics & Communications Engineering  
Gokongwei College of Engineering  
De La Salle University

In Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Electronics and Communications Engineering

Rey Anthony A. Masilang  
MS-ECE

Engr. Melvin K. Cabatuan  
Adviser

2014



## Abstract

Breast cancer is the most prominent cause of cancer-related deaths among women globally and is the 2<sup>nd</sup> leading cause of cancer-related deaths in the Philippines. Detecting cancer early significantly affects its curability. Breast self-examination (BSE) is a cost-effective and non-invasive method which can be performed by women themselves to monitor their breasts and detect the presence of lumps which can be cancerous. BSE is difficult to master especially without supervision and guidance from a professional. In response to this issue, this thesis aims to develop a computer vision-based BSE instruction and supervision system which can monitor and guide women while they perform BSE. The proposed system requires just a personal computer and a simple webcam. The proposed system is comprised of four major computer vision algorithms. The first one is devised to identify the entire areas of the left and right breasts using extensive use of color features, edges, contours, and curves. The second algorithm is devised to divide the breast area into smaller blocks, which the user will be asked to examine consecutively, using integral images and genetic algorithm. The third algorithm is devised to track the hand of the user using corner features and sparse optical flow. The fourth algorithm is devised to detect palpation motions using time series regression analysis. These algorithms were integrated together to create a complete BSE instruction and supervision with additional features such as real-time visual and audio feedback and a custom graphical user interface. This research aims to improve the proficiency of women in performing BSE through the use of the proposed interactive system in order to increase breast awareness among women and ultimately, if possible, reduce breast cancer mortality rate.



## Acknowledgments

I would like to express my gratitude to my mentors who made this thesis possible. First, I am deeply grateful to Dr. Elmer Dadios for allowing me to work and develop my thesis in this very special research. Second, I would like to express my deepest gratitude to my adviser, Engr. Melvin Cabatuan, for teaching and guiding me throughout this thesis, for providing me with important suggestions, improvements, and revisions to my research, and for constantly encouraging me to continue, improve, and finish this thesis.

I also am thankful to everyone who has contributed to the contents, substance, and meaning of this thesis. I thank my adviser again, Sir Melvin, and my colleague, John Anthony Jose for their ideas and suggestions which has allowed me to develop significant portions of this thesis. I also thank Dr. Elmer Dadios, Dr. Laurence Gan Lim, and Engr. Edwin Sybingco for their comments and suggestions which helped me improve this thesis further.

I also wish to thank the Engineering Research and Development for Technology Consortium (ERDT) for funding my studies and research as well as the Intelligent Systems Laboratory and all the people working there for providing me a fun and ideal working and learning environment. Also, I'd like to give special thanks to Engr. Argel Bandala, for teaching me important things which proved to be very helpful in the completion of this thesis.

Lastly, I am deeply grateful to everyone who has supported me in all possible ways especially my parents, friends, and Allaiza who has kept me inspired and motivated throughout this thesis. Their love and support has kept me going even when my goals seemed impossible and unattainable. I would like to share this success with each and every one of them.



## Contents

<b>Abstract.....</b>	i
<b>Acknowledgments.....</b>	ii
<b>Contents .....</b>	iii
<b>List of Figures.....</b>	viii
<b>List of Tables .....</b>	xvii
<b>1 Introduction .....</b>	1
1.1 Background of the Study .....	1
1.2 Statement of the Problem .....	2
1.3 Significance of the Study .....	3
1.4 Objectives of the Study.....	4
1.4.1 General Objective.....	4
1.4.2 Specific Objectives.....	4
1.5 Scope and Limitations .....	5
<b>2 Review of Related Literature.....</b>	6
2.1 Breast Cancer.....	6
2.1.1 Definition .....	6
2.1.2 Signs and Symptoms .....	6
2.2 Breast Cancer Screening.....	7
2.2.1 Mammography .....	7
2.2.2 Ultrasonography .....	7
2.2.3 Magnetic Resonance Imaging (MRI) .....	8
2.2.4 Clinical Breast Examination (CBE) .....	8



2.3 Breast Self-Examination (BSE) .....	9
2.3.1 Definition .....	9
2.3.2 Procedure .....	9
2.4 Zeng et al's Computer-based BSE Training System .....	12
2.4.1 Finger Tracking .....	12
2.5 BIOCORE's BSE Multimedia System.....	13
2.5.1 Hand Modeling .....	13
2.5.2 Hand Tracking .....	15
2.5.3 Hand Segmentation .....	16
2.5.4 Breast Area Delineation.....	17
2.6 Previous Thesis.....	17
2.6.1 Breast Detection, Nipple Detection, and Nipple Tracking .....	18
2.6.2 Breast Region Identification.....	18
2.6.3 Detecting Palpation.....	20
2.7 Related Computer Vision Algorithms.....	21
2.7.1 Breast Boundary Detection in Digital Mammograms.....	21
2.7.2 Hand Tracking .....	22
2.8 Research Gap .....	23
<b>3 Theoretical Framework .....</b>	<b>26</b>
3.1 Computer Vision Fundamentals.....	26
3.1.1 Digital Image .....	26
3.1.2 Digital Video .....	27
3.2 Color Models .....	28
3.2.1 RGB.....	28
3.2.2 Grayscale.....	29
3.2.3 HSV .....	30
3.2.4 Binary.....	31



3.3	Image Processing .....	32
3.3.1	Image Scaling .....	32
3.3.2	Region of Interest.....	33
3.3.3	Thresholding .....	33
3.3.4	Smoothing .....	34
3.4	Feature Extraction.....	35
3.4.1	Shi-Tomasi Corner Detection.....	35
3.4.2	Canny Edge Detection.....	36
3.5	Haar-like Features .....	37
3.6	Integral Image .....	38
3.7	Genetic Algorithm.....	39
3.7.1	Representation .....	39
3.7.2	Initialization .....	40
3.7.3	Selection.....	41
3.7.4	Crossover .....	42
3.7.5	Mutation.....	43
3.7.6	Termination.....	44
3.8	Optical Flow .....	45
3.8.1	Lucas-Kanade Optical Flow .....	47
3.8.2	Kanade-Lucas-Tomasi Feature Tracker .....	48
<b>4</b>	<b>Methodology.....</b>	<b>50</b>
4.1	Experiment Setup .....	50
4.2	Important Assumptions.....	51
4.3	Data Set .....	52
4.4	Breast Area Identification.....	56
4.4.1	Rationale .....	56
4.4.2	Framework.....	56



# De La Salle University

4.4.3 Stage 1: Skin Segmentation and Locating the Collar Bone .....	57
4.4.4 Stage 2: Identifying the Torso Region .....	62
4.4.5 Stage 3: Locating the Inframammary Fold .....	67
4.4.6 Stage 4: Locating the Outer Breast Edges and the Sternum .....	72
4.5 Breast Area Division .....	75
4.5.1 Rationale .....	75
4.5.2 Locating the Nipple.....	76
4.5.3 Genetic Algorithm .....	78
4.5.4 Quadrants and Blocks .....	81
4.5.5 Implementation.....	82
4.6 Hand Tracking.....	83
4.6.1 Rationale .....	83
4.6.2 Initialization .....	84
4.6.3 Tracking .....	88
4.7 Palpation Detection.....	93
4.7.1 Rationale .....	93
4.7.2 Signal Sampling.....	94
4.7.3 Time Series Regression Analysis.....	95
4.7.4 Criteria for Evaluation .....	97
4.7.5 Detecting Palpation.....	103
4.8 System Integration .....	104
4.8.1 System Overview .....	105
4.8.2 Process Overview .....	106
4.8.3 Graphical User Interface .....	108
<b>5 Results and Discussion .....</b>	<b>109</b>
5.1 Performance Measurement.....	109
5.1.1 Tracking Errors .....	109



5.1.2 F-score .....	109
5.1.3 Deviation.....	110
5.1.4 F1-score.....	110
5.1.5 Binary Classification Metrics .....	111
5.2 Breast Area Identification.....	112
5.2.1 Results.....	112
5.2.2 Discussion.....	113
5.3 Nipple Location and Breast Area Division .....	115
5.3.1 Results.....	116
5.3.2 Discussion.....	117
5.4 Hand Tracking.....	118
5.4.1 Results.....	119
5.4.2 Discussion.....	121
5.5 Palpation Detection.....	123
5.5.1 Part 1: Evaluation of the New Algorithm .....	123
5.5.2 Part 1: Results and Discussion .....	123
5.5.3 Part 2: Comparative Evaluation of the Old and New Algorithms .....	125
5.5.4 Part 2: Results and Discussion .....	126
5.6 Integrated System.....	129
<b>6 Conclusions and Recommendations .....</b>	<b>130</b>
6.1 Summary .....	130
6.2 Conclusion.....	131
6.3 Recommendations for Future Work.....	132
<b>References .....</b>	<b>135</b>



## List of Figures

Figure 2.3.1. BSE visual inspection part, steps 1 to 4 from left to right [27] .....	10
Figure 2.3.2. BSE physical examination positions, while lying down (left) and while standing or sitting (right) [27] .....	10
Figure 2.3.3. From left to right: small coin-sized circular motion of fingers during palpation, vertical strip pattern, concentric circle pattern, and wedge pattern [28] [29] .....	11
Figure 2.4.1. Results of feature extraction and group formation (left), group verification and pattern checking (middle), and sample tracking results (right) [31].....	12
Figure 2.5.1. Full degrees-of-freedom of a hand (left), simplified hand model with 8 degrees-of-freedom (center and right) [33].....	13
Figure 2.5.2. HCRA: Sobel edge detection and morphological operations (left) and pixel summing technique (right) [34] .....	14
Figure 2.5.3. Improved HCRA: pixel summing technique (left), detected finger pads (middle), and sample motion trajectory of the finger pads (right) [35].....	15
Figure 2.5.4. Sample hand segmentation results from [36]: initial segmentation using image difference (top row) and localized results using chest region identification (bottom row) [36] .....	16
Figure 2.5.5. BSE coverage area (left), BADA1 result (middle), and BADA2 result (right) [37].....	17
Figure 2.6.1. Sample results of the breast region identification method used [38] .....	18



# De La Salle University

Figure 2.6.2. Sample BSE performance with the breast region identified (a) and the cropped breast region with blocks 2 and 3 already highlighted signifying completed palpation (b) [38].....	20
Figure 2.7.1. Sample results of breast contour detection on digital mammograms using polynomial modeling: original mammogram (left), binary image (middle), and final contour (right) [39] .....	21
Figure 2.7.2. Sample hand tracking/segmentation results: original image (left) and segmentation result (right) [45] .....	22
Figure 3.1.1. Sample digital image (left) and array of integer component values for the 10x10 pixel region (right) .....	26
Figure 3.1.2. Digital video represented as a sequence of digital images or frames.....	27
Figure 3.2.1. Sample image and its RGB components shown as intensity images: original image (a), red component (b), green component (c), and blue component (d).....	28
Figure 3.2.2. (a) Original colored image and (b) its grayscale image counterpart .....	29
Figure 3.2.3. Sample image and its HSV components shown as intensity images: (a) original image, (b) Hue component, (c) Saturation component, and (d) Value component.....	30
Figure 3.2.4. Sample binary images derived from (a) the original image: (b) thresholded image and (c) image showing detected edges.....	31
Figure 3.3.1. Example of image scaling: (a) original image and downscaled version using a factor of 0.2 shown in its (b) actual size and (c) stretched to original size.....	32
Figure 3.3.2. (a) Original image with ROI selected and (b) the extracted ROI .....	33



# De La Salle University

Figure 3.3.3. (a) Sample grayscale image, thresholded image using a threshold of (b) 0.4 and (c) 0.6.....	33
Figure 3.3.4. (a) Sample grayscale image and (b) filtered image using median filter .....	34
Figure 3.4.1. Sample corner detection result: red dots correspond to the detected corners [56].....	35
Figure 3.4.2. (a) Original image and (b) binary image showing the detected edges [59] .....	36
Figure 3.5.1 Examples of regular Haar-like features (top) and tilted Haar-like features (bottom) [63] .....	38
Figure 3.6.1 Calculating the sum of a rectangular section of an image using Integral Image .....	39
Figure 3.7.1. Sample 2-dimensional search space of the breast area (a) and the initial population with randomly generated solutions [blue circles] (b) .....	40
Figure 3.7.2. Randomly generated solutions [blue circles] after the initialization phase (a) and solutions selected to form the elite pool [red circles] after the selection phase (b) .....	41
Figure 3.7.3. The elite pool of candidate solutions [red circles] after the selection phase (a) including the crossover children [green circles] after the crossover phase (b) .....	43
Figure 3.7.4. The current population [red circles] after the crossover phase (a) and the mutated children [yellow circles] after the mutation phase (b).....	44
Figure 3.7.5. Evolution of the candidate solutions: initial population (a), population after 5 generations (b), and the final population after 20 generations (c).....	45



Figure 3.8.1. Optical flow examples: (a) cars on a highway with sparse optical flow results, and (b) pedestrians on a street with (c) dense optical flow results [62] .....	46
Figure 3.8.2. The Lucas-Kanade optical flow/KLT tracker applied on the hand tracking problem .....	48
Figure 4.1.1. Block diagram of the experimental setup of the BSE supervision system .....	50
Figure 4.2.1. Recommended position of user (a) at the start of the BSE procedure and the recommended area captured by the webcam (b) .....	51
Figure 4.3.1 Sample images of the female torso included in Data Set 1 .....	53
Figure 4.3.2 The 14 video sequences included in Data Set 2 .....	54
Figure 4.3.3 The 12 video sequences included in Data Set 3 .....	55
Figure 4.4.1 Recommended area of coverage for BSE for the left and right breasts (a) and visual cues for the automatic delineation (b) .....	56
Figure 4.4.2 Block diagram of the Breast Area Identification Framework.....	57
Figure 4.4.3 Block diagram of Stage 1 of the Breast Area Identification Algorithm .....	58
Figure 4.4.4 Feature extraction: input image (a), YCbCr channels (b), extracted Cr channel (c) .....	59
Figure 4.4.5 Skin segmentation process: Cr channel (a), preliminary skin mask (b), skin mask after closing (c), and skin mask after erosion (d) .....	59
Figure 4.4.6 Illustration of the skin mask reference level and collar bone location .....	60
Figure 4.4.7 Simplified code for Stage 1 of the Breast Area Identification Algorithm .....	61



Figure 4.4.8 Block diagram of Stage 2 of the Breast Area Identification Algorithm .....	62
Figure 4.4.9 Preprocessing step: reference column (a) and the initial torso bounding box (b)v .....	63
Figure 4.4.10 Edge detection and filtering step: original colored image (a), grayscale image (b), filtered image (c), Canny edge detection output (d), contour detection output (e), contour filtering output (f) .....	64
Figure 4.4.11 Search for torso edges: edges within the initial torso bounding box (a), sliding summation windows (b), torso edges located (c), final torso bounding box (d) .....	65
Figure 4.4.12 Simplified code for Stage 2 of the Breast Area Identification Algorithm .....	66
Figure 4.4.13 Block diagram of Stage 3 of the B.A.I. algorithm.....	67
Figure 4.4.14 Preprocessing step: cropped half of the torso region (a), grayscale image (b), filtered image (c), Canny edges (d), detected contours (e), filtered contours (f).....	68
Figure 4.4.15 Curve fitting process: half torso image in grayscale (a), contours detected and filtered (b), curve fitness evaluation (c), best-fitting parameters (d), and best-fitting curve (e).....	69
Figure 4.4.16 Defining the breast area: results of the previous step (a), best-fitting curve superimposed on the original image (b), breast area bounding box (c), left and right breast areas (d) .....	70
Figure 4.4.17 Simplified code for Stage 3 of the Breast Area Identification Algorithm .....	71
Figure 4.4.18 Block diagram of Stage 4 of the Breast Area Identification Algorithm .....	72



Figure 4.4.19 Stage 4 process: extended breast region (a), grayscale image (b), filtered image (c), Canny edges (d), detected contours (e), filtered contours (f), longest contour (g), outer boundary (h), adjusted left and right breast area bounding boxes (i).....	73
Figure 4.4.20 Simplified code for Stage 4 of the Breast Area Identification Algorithm .....	74
Figure 4.5.1. Diagram of the breast quadrant system showing the five major regions: nipple-areola complex, upper outer quadrant, upper inner quadrant, lower outer quadrant, lower inner quadrant, and axillary tail [66] (a) and the proposed breast area division into smaller blocks (b).....	75
Figure 4.5.2 Center-surround Haar-like feature (a) and sample nipple tracking results (square) within the breast area (red box) (b) .....	76
Figure 4.5.3 Block diagram of Genetic Algorithm (a) and structure of a chromosome (b).....	78
Figure 4.5.4 Block diagram of the crossover phase used in the genetic algorithm.....	79
Figure 4.5.5 Sample breast areas divided primarily into four quadrants each .....	81
Figure 4.5.6 Sample breast areas divided into quadrants and smaller blocks.....	82
Figure 4.6.1 Sample results of the proposed hand tracking methodology.....	83
Figure 4.6.2 Block diagram of the Initialization Stage of the Hand Tracking Algorithm .....	84
Figure 4.6.3 Preprocessing step: original breast region (a), extended breast region (b), ROI in grayscale (c), and ROI after applying Gaussian blur (d).....	85



Figure 4.6.4 Corner detection results (a) and outlier elimination results (b) showing detected corners (red) and selected corners (blue) .....	86
Figure 4.6.5 Sample initialization result showing detected corners (blue), best-fitting ellipse (green) and estimated finger pad location (red) .....	87
Figure 4.6.6 Simplified code for the initialization stage of the Hand Tracking Algorithm.....	87
Figure 4.6.7 Block diagram of the tracking stage of the Hand Tracking Algorithm .....	88
Figure 4.6.8 Example of large motion vectors {red arrows} (a) and very small motion vectors {red arrows} (b).....	89
Figure 4.6.9 Hand motion estimation: motion vectors of the feature set (blue arrows) and the average motion vector (white arrow) .....	90
Figure 4.6.10 Simplified code for the optical flow tracking step of the Hand Tracking algorithm.....	91
Figure 4.6.11 Simplified code for the Feature Replacement step of the Hand Tracking algorithm.....	92
Figure 4.7.1 Sample X signal (a) and Y signal (b) .....	94
Figure 4.7.2 Sample X buffer with values for the waveform shown in Figure 4.7.1 (a).....	94
Figure 4.7.3 Sample X signal (a) and Y signal (b) during palpation with their respective linear regression models (green line) .....	95
Figure 4.7.4 Code snippet for the Linear Regression Analysis in the Palpation Detection Algorithm.....	96
Figure 4.7.5 Sample X signal (a) and Y signal (b) for non-palpation motion and sample X signal (c) and Y signal (d) for palpation motion with their respective linear regression models (green line) .....	97



Figure 4.7.6	Code snippet for the evaluation of the first criterion for correct palpation.....	98
Figure 4.7.7	Sample X signals showing palpation at different speeds: too slow (a), ideal (b), and too fast (c) .....	99
Figure 4.7.8	Code snippet for the evaluation of the second criterion for correct palpation.....	99
Figure 4.7.9	Sample X signal (a) and Y signal (b) for palpation motion performed within the specified block (blue region) (c) .....	100
Figure 4.7.10	Sample X signal (a) and Y signal (b) for palpation motion performed outside the specified block (blue region) (c); The Y signal lies within the allowable region, the X signal however does not. ....	101
Figure 4.7.11	Code snippet for the evaluation of the third criterion for correct palpation.....	101
Figure 4.7.12	Sample X signals showing hand motions of various sizes: too small (a), ideal (b), and too large (c) .....	102
Figure 4.7.13	Code snippet for the evaluation of the last criterion for correct palpation.....	103
Figure 4.7.14	Sample X signals showing non-palpation motion (a) and palpation motion (b) .....	104
Figure 4.8.1	General Block Diagram of the BSE Instruction and Supervision System.....	105
Figure 4.8.2	Flowchart of the complete BSE Instruction and Supervision System.....	106
Figure 4.8.3	Graphical User Interface (GUI) of the integrated system.....	108
Figure 5.2.1	Performance results for the 200 test images using the Old Algorithm .....	113



Figure 5.2.2 Performance results for the 200 test images using the New Algorithm .....	113
Figure 5.2.3 Sample images from Data Set 1 showing the results of the Old Algorithm (top), results of the New Algorithm (middle), and ground truth (bottom).....	115
Figure 5.2.4 Sample images from Data Set 1 showing the results of the Old Algorithm (top), results of the New Algorithm (middle), and ground truth (bottom).....	115
Figure 5.3.1 Left Nipple deviations for the 200 test images .....	116
Figure 5.3.2 Right Nipple deviations for the 200 test images .....	117
Figure 5.3.3 Sample images from Data Set 1 showing the tracked left and right nipples using the Nipple Location Algorithm .....	118
Figure 5.3.4 Sample images from Data Set 1 showing the breast area divided into 4 quadrants and further into smaller blocks using the Breast Area Division Algorithm.....	118
Figure 5.4.1 Precision values for the 14 videos in Data Set 2 using Hu's algorithm (blue) and the new algorithm (red) .....	120
Figure 5.4.2 Recall values for the 14 videos in Data Set 2 using Hu's algorithm (blue) and the new algorithm (red) .....	121
Figure 5.4.3 F-scores for the 14 videos in Data Set 2 using Hu's algorithm (blue) and the new algorithm (red).....	121
Figure 5.4.4 Sample frames from the 14 videos in Data Set 2 showing the tracking results (red circle) using the new algorithm.....	122
Figure 5.5.1 Performance of the Old Algorithm with threshold=50k and New Algorithm.....	128
Figure 5.6.1 Sample frames from the BSE trials run using the complete system .....	129



## List of Tables

Table 5.2.1	Performance of Old and New Breast Area Identification Algorithms .....	112
Table 5.3.1	Performance of the Nipple Tracking Algorithm .....	116
Table 5.4.1	Performance of Hu et al's Hand Tracking Algorithm .....	119
Table 5.4.2	Performance of the newly developed Hand Tracking Algorithm .....	120
Table 5.5.1	Binary classification results of the Palpation Detection Algorithm .....	124
Table 5.5.2	Confusion matrix for the Palpation Detection Algorithm .....	124
Table 5.5.3	Performance of the Palpation Detection Algorithm .....	124
Table 5.5.4	Binary classification results of the Old and New Palpation Detection Algorithms .....	126
Table 5.5.5	Performance of the Old and New Palpation Detection Algorithms .....	127



## Chapter 1

### Introduction

#### 1.1 Background of the Study

Breast cancer is the most prominent cause of cancer-related deaths among women globally. It is the most common cancer in women in both developed and developing countries [1]. In the Philippines, breast cancer is 20<sup>th</sup> among the top 50 causes of deaths [2]. It accounts for 4,085 deaths in 2010 alone [2]. Breast cancer is also the 2nd leading cause of death due to cancer in the Philippines behind lung cancer and is ranked 135th in the world [2].

The curability of breast cancer is dependent on many factors. One of these is how early the disease is detected, diagnosed, and classified. Imaging tests such as mammography and magnetic resonance imaging (MRI) are used to detect breast cancer. However, most of these procedures are costly and not readily available to people especially to those in developing countries [3]. In the Philippines, the average cost for mammography is around P1000 while breast MRI costs around P8000 [4]. Breast self-examination (BSE) is a method which can be performed by women which does not require special procedures and personnel. Though not all breast cancer cases can be found this way, BSE is still an important tool in the early detection of breast cancer when there is still high capability of being cured [5].

Breast self-examination involves a woman visually and physically inspecting each of her breasts for changes, discharges, lumps and swelling. Physical



examination of the breast is difficult to learn to perform proficiently. It involves training the fingers to feel and detect subtle changes in the breast tissue which can signal early breast cancer. Due to this difficulty, breast self-examination is known for its ineffectiveness in reducing breast cancer mortality rates [6]. At present, the importance of screening by BSE has not yet been established. However, there is case-control evidence that BSE, if properly and efficiently performed, can be as effective as mammography in reducing breast cancer mortality rates [7].

## 1.2 Statement of the Problem

The apparent ineffectiveness of BSE can be attributed to the poor proficiency of women practicing this method, lack of training from BSE instructors, inconsistency in performing BSE, and inability to continue performing BSE. The three general methods of BSE training include the use of pamphlets and posters, instructional videos, and actual supervised practice of BSE on a life-like silicone breast model. Experimental results show that passive methods were not helpful at all in training women in BSE. Only the third method yielded positive results in which lump detection performance of women increase. Corrective feedback from a BSE instructor proved to be crucial in the effectiveness of the BSE practice performance [7].

By as early as 2001, researchers in UK proposed the use of computer vision and multimedia techniques as replacement for traditional BSE training methods. This involves the development of computer algorithms for the corrective feedback part of BSE instruction. They were able to produce a number of preliminary algorithms. Also, in a recent thesis by E. Mohammadi Nejad [8], a prototype BSE guidance system was developed which provides a number of essential features in a BSE guidance system. However, most of these existing algorithms must still be improved



further in order to come up with an intelligent BSE guidance system. Hence, this thesis aims to continue and improve upon the works of BIOCORE and Mohammadi Nejad to better solve the task of using computer vision in training women in the use of BSE. Specifically, this thesis will address the problem of developing/improving the following major components:

1. **Breast area characterization.** This involves the identification of the entire left and right breast areas and dividing them into appropriately sized blocks for palpation.
2. **Palpation tracking.** This involves tracking of hand activity within the breast area specifically those which correspond to palpation.
3. **System integration.** This refers to the integration of all algorithms/components necessary in a BSE instruction and supervision system.

### 1.3 Significance of the Study

This research will be beneficial to the female population especially to those who cannot afford a regular physical check-up and to the medical community as a whole. BSE remains as a viable alternative for women to monitor the condition of their breasts. Through this study, an instruction and supervision system can be developed and implemented which can be used by women to learn about BSE and how to perform it proficiently without the assistance of trained professionals. With this, women will have another option in keeping their breast health in check without the embarrassment, discomfort, and expenditure involved in other screening procedures. To the medical community, this research will help in the standardization of breast self-examination. With this, the inconsistencies involved in



the training of women in performing BSE can be minimized. Hence, the disadvantages attributed to the non-standardized BSE procedure can be diminished.

## 1.4 Objectives of the Study

### 1.4.1 General Objective

To develop a computer vision-based breast self-examination instruction and supervision system capable of monitoring and evaluating BSE and providing feedback to the user in real-time

### 1.4.2 Specific Objectives

1. To develop an algorithm for identifying the entire breast examination area recommended by IARC [8] using skin segmentation, edges, contours, and curve-fitting
2. To develop an algorithm for locating the nipple and dividing the breast region into four quadrants and smaller blocks using edges, contours, integral image, and genetic algorithm
3. To develop an algorithm for detecting and tracking the hand using corner features and sparse optical flow
4. To develop an algorithm for detecting palpation using time series regression analysis.
5. To integrate the individual algorithms into a complete system with a graphical user interface and real-time audiovisual cues
6. To validate the improved performance and robustness of each algorithm using the following metrics:
  - a. F-score
  - b. Deviation



- c. F1-score
- d. TPR, TNR, PPV, NPV, and Accuracy

## 1.5 Scope and Limitations

The proposed instruction and supervision system is designed for the use of women only. Breast self-examination for detecting male breast cancers is not part of this study. Due to the efficiency of BSE itself being largely dependent on the woman performing it, the effectiveness of the proposed training and guidance system is also highly dependent on each individual using it. Also, implementing the system on a community and measuring its effectiveness in terms of how much improvement women can attain by using it and/or measuring how much it can contribute to lessening breast cancer mortality rates is not part of this study.

The BSE instruction and supervision system will be written in C++. The OpenCV (Open Source Computer Vision) library will be highly utilized for the computer vision-based algorithms to be developed. The study mainly involves the development of computer vision algorithms for this specific application. Communication between the user and the system is done through button presses, mouse-clicks, real-time video input, and real-time audiovisual feedback. User-to-system interaction through voice commands is not part of the study. Testing of the system is to be conducted using pre-recorded videos of women performing BSE, images of breasts and BSE from the internet, and real-time testing on a mannequin or dummy modified specifically for this research.



## Chapter 2

### Review of Related Literature

#### 2.1 Breast Cancer

##### 2.1.1 Definition

Breast cancer is a type of cancer, a disease which involves unregulated cell growth, which originates from the breast tissue. Breast cancer may originate either from the inner lining of the milk ducts (ductal carcinomas) or the lobules that supply milk to the ducts (lobular carcinomas) [9]. The primary risk factors for the development of breast cancer are gender (female) and old age. Other risk factors include genetics, higher levels of certain hormones, certain dietary patterns, obesity, and exposure to light pollution [10] [11] [12] [13].

##### 2.1.2 Signs and Symptoms

The most common symptoms of breast cancer are the following [14] [15] [16]:

1. Presence of a lump
2. Thickening
3. Change in size and shape of breast/s
4. Change in nipple position or shape
5. Nipple inversion
6. Skin puckering or dimpling
7. Rashes on or around the nipple/s



8. Discharge from nipple/s
9. Constant pain in a part of the breast or armpit
10. Swelling beneath the armpit or collarbone

## 2.2 Breast Cancer Screening

### 2.2.1 Mammography

Mammography, also known as screen-film mammography (SFM), is the most widely used form of and the current standard for breast cancer detection. In this method, the breasts are compressed while an X-ray examination is performed. Sensitivity and specificity of mammography are dependent on breast density which is directly affected by age, menstrual cycle phase, body mass index, genetics, etc. [17]. Mammography is less sensitive in younger women due to their typically denser breast tissue [17]. Mammography requires a dedicated machine, an X-ray technologist, and a radiologist. In this method, the breasts are compressed which causes discomfort to the patient. Also, in this method, the breasts are exposed to radiation which may lead to radiation carcinogenesis [18].

### 2.2.2 Ultrasonography

Ultrasonography, also known as sonomammography, is an often used follow-up test to mammography. In this method, sound waves bounce off the breast tissues to create an image. Ultrasonography is suitable for scanning dense breasts, most common in younger and pregnant women [19]. One advantage of this method is its applicability in real-time imaging [19]. However, ultrasonography is operator-dependent which means that the process is very sensitive to the performance of the operator. Ultrasonographic images are susceptible to speckle, clutter, and blurring



which lowers the image quality, thereby reducing its reliability in detecting tumors [20].

### **2.2.3 Magnetic Resonance Imaging (MRI)**

Magnetic resonance imaging (MRI) uses a magnetic field and radio waves to create an image. A contrast agent, usually gadolinium-based, is used in breast imaging. MRI is not suitable for fatty breasts unlike mammography. Therefore, fat must be suppressed before injecting the contrast agent through post-processing subtraction or other techniques [21]. Unlike mammography, MRI is safer as it does not use radiation [21]. Compared to ultrasonography, MRI is less dependent on the operator and has better image quality and resolution [21]. However, the contrast agent used may affect benign lesions and the fetus inside a pregnant patient [21].

### **2.2.4 Clinical Breast Examination (CBE)**

Clinical breast examination is a physical examination of the breasts by a doctor, nurse, or other health professionals. In this method, a trained professional visually checks the breasts of a patient and physically examines the breasts and underarms while the patient is lying down to check for any changes or abnormalities such as lumps. Clinical breast examination is usually a part of regular checkups and is also used to detect other breast problems [22]. CBE is used complementary to mammography, ultrasound, or breast MRI to detect breast cancer among women [23]. CBE combined with mammography is attributed with a higher breast cancer detection rate [24]. However, CBE alone is not a replacement for mammography. The most common drawback of using CBE is the increased chance of false positive results which occur when a CBE finds lumps which resemble cancer but turn out to be benign [25].



## 2.3 Breast Self-Examination (BSE)

### 2.3.1 Definition

Breast self-examination is a systematic method of inspecting and palpating the breasts of a woman [26]. Breast self-examination is highly similar with clinical breast examination. The main difference is that the procedure is performed by the woman herself on her own breasts.

### 2.3.2 Procedure

Breast self-exams are typically comprised of two parts: the visual inspection part and the physical examination part. For the visual inspection part, the woman faces a mirror which is large enough for the woman to see both breasts clearly. Then, the following steps are employed [27]:

1. With both arms on the sides of the body and the shoulders straight, the woman checks her breasts for puckering, dimpling, or other changes in appearance.
2. With both arms raised over the head, step 1 is repeated.
3. With the hands firmly pressed on the hips and the body slightly bent toward the mirror, step 1 is repeated.
4. Last, the woman gently squeezes each nipple to look for a discharge

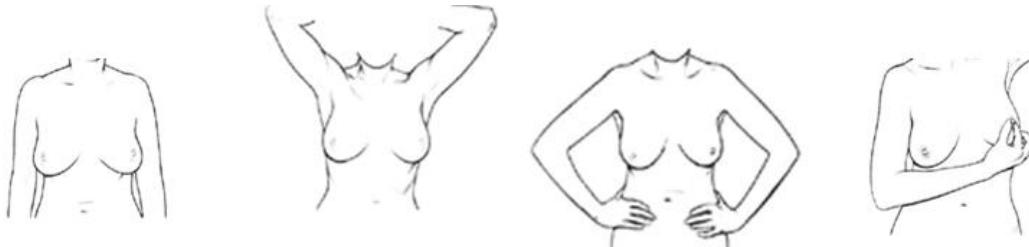


Figure 2.3.1. BSE visual inspection part, steps 1 to 4 from left to right [27]

For the physical examination part, the woman feels each of her breasts. This process is done in two positions [27] [28]:

1. While lying flat on the back, with one arm over the head and a pillow under the shoulder. This position flattens the breast and makes the examination easier to conduct.
2. While standing or sitting, with one arm over the head. This is usually done in the shower because women find it easiest to feel their breasts when their skin is wet and slippery.



Figure 2.3.2. BSE physical examination positions, while lying down (left) and while standing or sitting (right) [27]

The breasts are examined using the first few finger pads of the hand. A circular motion, about the size of a coin, is used [27]. The entire breast area is covered from top to bottom and from side to side. A pattern is usually used to ensure full coverage of the breast area. The most commonly used patterns are the following [28]:



1. **Vertical Strips.** Starting in the underarm area, the fingers are moved downward little by little until they are below the breast. Then the fingers are moved slightly towards the middle and slowly moving back up. The up and down procedure is repeated until the entire breast area is covered.
2. **Concentric Circles.** Starting at the outer edge of the breast, the fingers are moved around the whole breast in a circle. The fingers are moved around the breast in smaller and smaller circles toward the nipple.
3. **Wedges.** Starting at the outer edge of the breast, the fingers are moved toward the nipple and back to the edge. The whole breast is checked, covering one small, wedge-shaped section at a time.

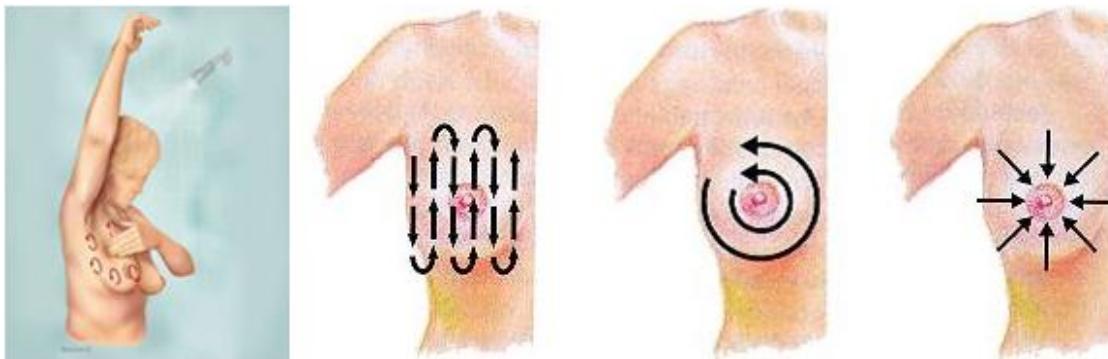


Figure 2.3.3.

From left to right: small coin-sized circular motion of fingers during palpation, vertical strip pattern, concentric circle pattern, and wedge pattern [28] [29]



## 2.4 Zeng et al's Computer-based BSE Training System

The oldest published papers about the application of computer vision to BSE are two conference papers by Zeng et al. These papers basically tackle the same algorithm developed by these authors [30] [31]. In here, they present a finger tracking technique to be used in a computer-based BSE training system.

### 2.4.1 Finger Tracking

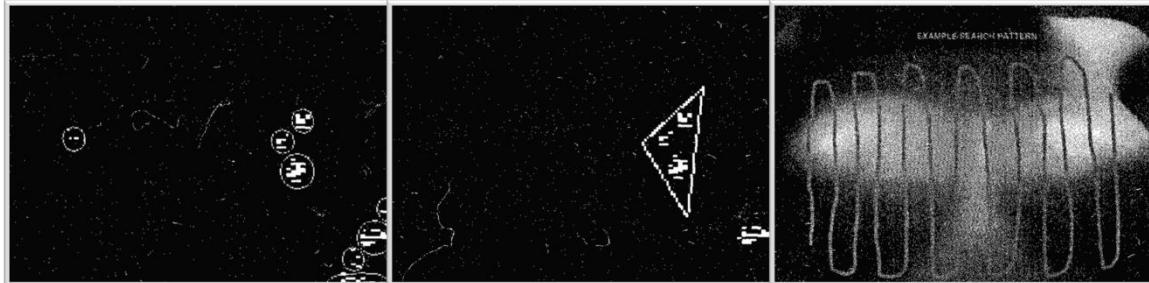


Figure 2.4.1. Results of feature extraction and group formation (left), group verification and pattern checking (middle), and sample tracking results (right) [31]

Their proposed technique uses colored finger nails as markers to be tracked using color features because the background in breast palpation is the breast itself which is very similar in color to the hand [30]. Using these color features, binary thresholding, and feature grouping, the algorithm is shown to be capable of accurate tracking of the fingers while eliminating falsely detected features. Additionally, their paper presents a position calculation method by stereo vision to extract depth information. Results of the depth estimation however were not presented. The proposed method is very robust and accurate. The main limitation however is the strict requirement that the patient wears colored nail polish before using their system.



## 2.5 BIOCORE's BSE Multimedia System

The Applied Research Group in Biomedical Computing and Engineering Technologies (BIOCORE) of Coventry University undertook a collaborative research project named as “Interactive multimedia system for breast awareness” [32]. Taken from articles included in conference proceedings, the following are reviews of relevant computer vision algorithms developed by BIOCORE for their BSE multimedia system.

### 2.5.1 Hand Modeling

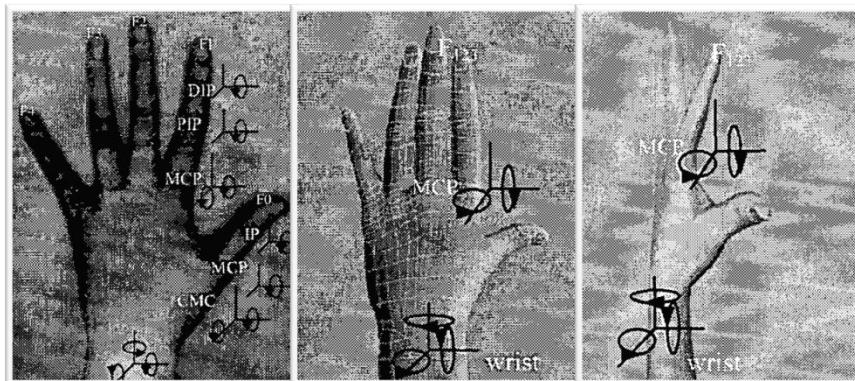


Figure 2.5.1. Full degrees-of-freedom of a hand (left), simplified hand model with 8 degrees-of-freedom (center and right) [33]

BIOCORE has published 4 conference papers specifically concerned about the hand. The first is about the development of a simplified 3D hand model with a minimal 8 degrees-of-freedom [33]. This model presents a marked reduction in complexity of a highly articulated natural hand. This is intended to provide an easy way of analyzing hand motion in a computer vision-based BSE application. This, however, has not yet been utilized alongside computer vision techniques in BSE applications.

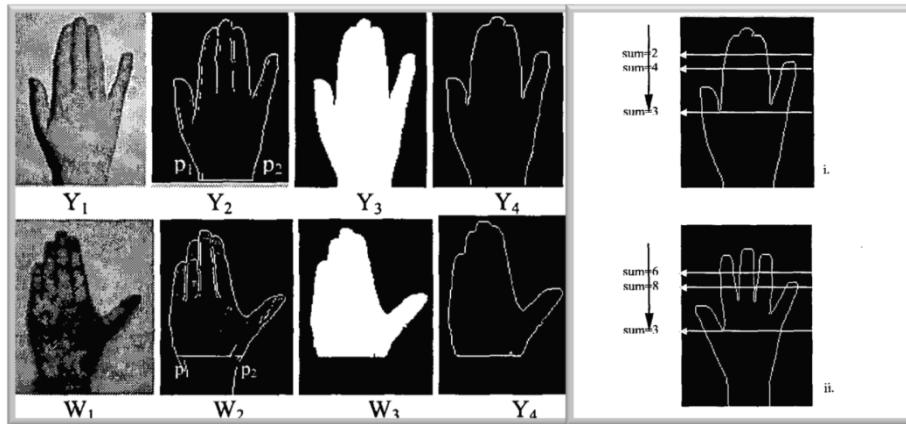


Figure 2.5.2. HCRA: Sobel edge detection and morphological operations (left) and pixel summing technique (right) [34]

In another paper of BIOCORE, they developed a new hand configuration recognition algorithm, HCRA, which is intended to be used in recognizing a correct hand configuration for use in breast self-examination applications [34]. The algorithm is mainly comprised of Sobel edge detection, morphological operations, and a pixel summing technique. By creating a class of hand configurations and conducting experiments, the algorithm is shown to be capable of recognizing correct BSE hand configurations with high sensitivity and specificity. The algorithm however is designed only for a flat view of the hand. The algorithm is not designed to handle different hand orientations which are very common during actual BSE performance.



## 2.5.2 Hand Tracking



Figure 2.5.3. Improved HCRA: pixel summing technique (left), detected finger pads (middle), and sample motion trajectory of the finger pads (right) [35]

The HCRA algorithm is further developed in another paper to specifically locate the finger pads and track the motion of the hand in a video [35]. Tracking the finger pads is the primary concern in BSE evaluation because the finger pads are the specific part of the hand used to examine the breasts in BSE. This improved algorithm however suffers the same weakness as that of the HCRA algorithm. The effectiveness of this algorithm is highly dependent on the orientation of the hand being tracked. Another weakness, which was acknowledged by the authors, is how to properly segment the hand in a BSE video as the experiments conducted were only on images of a hand against a wall. In this setup the hand is easily distinguishable from the wall through skin color techniques which isn't the case in an actual BSE video because the hand is of the same color with the breast background.



### 2.5.3 Hand Segmentation

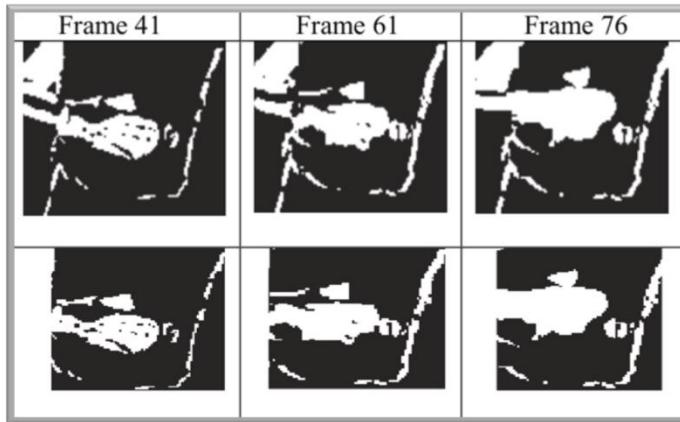


Figure 2.5.4. Sample hand segmentation results from [36]: initial segmentation using image difference (top row) and localized results using chest region identification (bottom row) [36]

In their last paper among the series of 4 conference papers tackling hand tracking and segmentation, they addressed the hand segmentation problem against skin color background [36]. The algorithm starts with identifying the chest region through a simple skin modeling technique. This is used to limit the region of analysis for the hand motion segmentation. Next, the hand is tracked simply through image difference, wherein subsequent frames are subtracted. An important concern in this technique is that within the chest region, both the hand and the breast being examined must be moving during palpation. Using simply image subtraction, motion due to the breast's movement will also be detected. In this case, hand segmentation will include the moving areas of the breast.



## 2.5.4 Breast Area Delineation

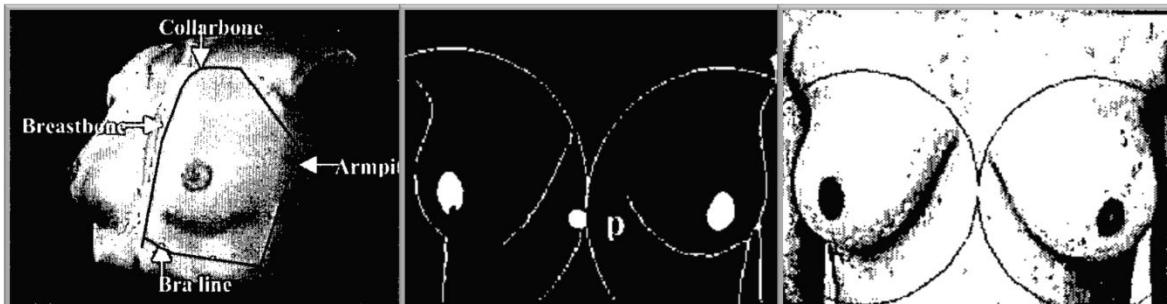


Figure 2.5.5. BSE coverage area (left), BADA1 result (middle), and BADA2 result (right) [37]

BIOCORE also published a conference paper which presents their two breast area delineation algorithms [37]. They named these two algorithms as BADA1 and BADA2. BADA1 involves locating the nipple-areola complexes of the chest region and drawing two circles at the centroid of these targets. The radii of these two circles are equal to half of the distance between the two centroids. BADA2 on the other hand utilizes a probabilistic Hough transform to draw two circles coinciding with the contours of the two breasts. These two circles are both enlarged to meet tangentially. In these two algorithms, the authors define the BSE area as the area within the two circles. However, based from the figure above which is sourced from their paper, the actual BSE coverage area is very different from the areas indicated by BADA1 and BADA2 results. Further research is necessary to specifically delineate the actual BSE area.

## 2.6 Previous Thesis

In a previous thesis authored by Eman Mohammadi Nejad, he developed a complete computer vision-based BSE system [38]. The system is comprised of the



following parts: breast detection, nipple detection, nipple tracking, breast region identification, and palpation detection. A summary of each part is provided below.

## 2.6.1 Breast Detection, Nipple Detection, and Nipple Tracking

For the breast detection part, Haar-like features are utilized alongside a cascaded AdaBoost classifier in order to arrive at regions-of-interest (ROIs) which contain the left and right breasts of the patient. Within these ROIs, the nipples are located by dividing each ROI into 6 blocks, calculating the sum of pixel intensities within each block using integral imaging, and comparing which block has the smallest sum of pixel intensities. To minimize false positives, the red components of all pixels within the block identified as the nipple along with the mean, standard deviation, and HOG (histogram of oriented gradients) features are fed input to an ANN (artificial neural network) to classify whether the block indeed corresponds to the nipple location or not. Last, the detected nipple is used as a template for tracking the nipple in subsequent frames using the normed correlation coefficient matching method.

## 2.6.2 Breast Region Identification



Figure 2.6.1. Sample results of the breast region identification method used [38]



For the breast region identification part, the system superimposes a matrix of 2x3 blocks on each breast. The width of the matrix is estimated as half the distance between the left and right nipples while the height of the matrix is estimated based on the height of the detected ROI in the breast detection part. During the BSE proper, that is while the patient is palpating her breast, the matrix is superimposed on the breast being palpated by using the nipple of the opposite breast as reference. This necessitates the patient to use her left hand for palpating her left breast and vice versa to prevent occlusion on the reference nipple.

Specific areas for improvement regarding this technique include the assumptions used for estimating the width and height of the breast area as well as the method of palpation required by the technique. First, the distance between the two nipples is not a reliable basis for estimating the width of each breast. Same case with the height estimation, the technique used is highly dependent on the breast detection part. Based from results, the breast region identification method proposed is not accurate as shown in the figure above. Second, the requirement that the patient uses her left hand for palpating her left breast and right hand otherwise is opposite the common standards prescribed by doctors in performing BSE [28].



### 2.6.3 Detecting Palpation

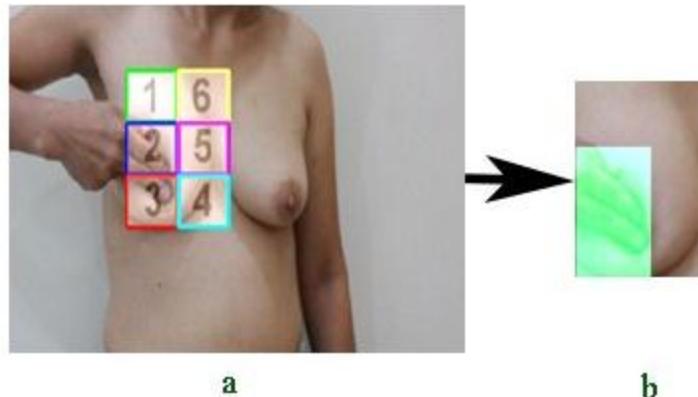


Figure 2.6.2. Sample BSE performance with the breast region identified (a) and the cropped breast region with blocks 2 and 3 already highlighted signifying completed palpation (b) [38]

Palpation is detected in the system by detecting motion within the blocks of the breast region matrix. Using Farneback optical flow on a block, the motion vectors within the block are extracted. The array of motion vectors are grouped into four equal tiles. The overall magnitude (velocity) of motion vectors for each tile is evaluated through Pythagorean addition to calculate the L2-norm. Using a threshold on this value, each tile for that specific block for that specific frame is classified as either palpated or non-palpated. Using the recommended 10-second palpation for each region, a specific tile must be classified as palpated for 200 consecutive frames. All four tiles of a block must be palpated completely before concluding that the patient has finished palpating that block.

The proposed method here implies that motions within a block, as long as it is within the specified threshold, correspond to palpation. However, there can also be non-palpation motions detected. Further research can be conducted to better characterize palpation motions from non-palpation motions.



## 2.7 Related Computer Vision Algorithms

### 2.7.1 Breast Boundary Detection in Digital Mammograms

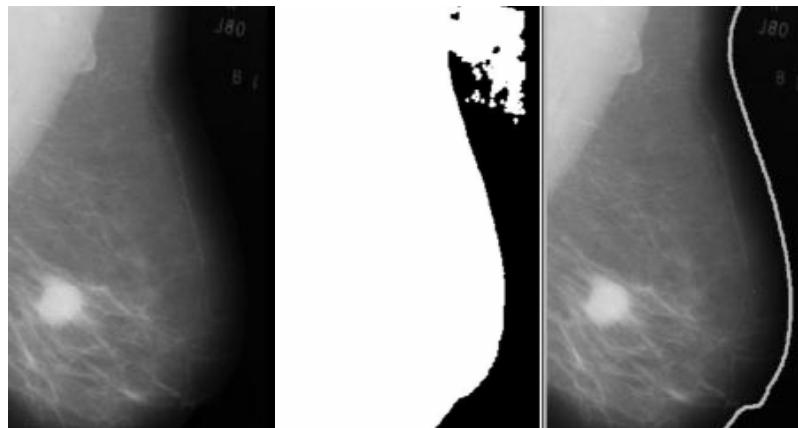


Figure 2.7.1. Sample results of breast contour detection on digital mammograms using polynomial modeling: original mammogram (left), binary image (middle), and final contour (right) [39]

Computer vision has found its way into medical diagnosis through Computer Aided Diagnosis (CAD) systems. In specific, CAD systems have improved the diagnosis of abnormalities in mammogram images. An important pre-processing for these mammograms before proceeding to the detection of abnormalities is accurate detection of the breast boundaries. Numerous approaches have been proposed so far for this task. These approaches have utilized image processing operations such as thresholding, gradients, and polynomial curve fitting.

Among early approaches include the algorithm proposed by Masek et al which uses an adaptive local threshold for segmenting the breast tissue area [40]. Another algorithm proposed by Lou et al is based on the assumption that the trace of intensity values from the breast region to the air-background is a monotonic decreasing function [41]. Wirth et al proposed the use of curve approximation through active contours to extract breast contours using a principal rationale that



the breast is a well-defined curve [42]. More recent approaches include the work of Mirzaalian et al which is based on polynomial modeling for accurately detecting the breast contour [39]. Another recent method by Wirth et al uses a rule-based fuzzy reasoning algorithm for segmenting the breast region [43]. Among the most recent is the proposed method by Maitra et al which introduces a new homogeneity enhancement process which they call Binary Homogeneity Enhancement Algorithm [44].

## 2.7.2 Hand Tracking

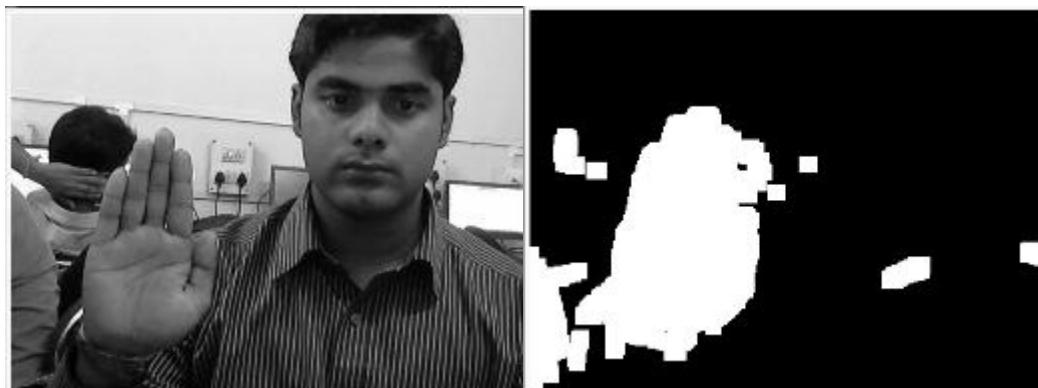


Figure 2.7.2. Sample hand tracking/segmentation results: original image (left) and segmentation result (right) [45]

Currently, there is an abundant supply of literature regarding hand tracking because of its extension, hand gesture recognition, which has great potential to be used for visual interaction with computers in virtual reality environments. Recent approaches include the work of Cheng et al which uses five complementary processes comprised of motion, skin detection, edge detection, movement justification, and background subtraction [46]. Zhang et al, on the other hand, proposed the use of contour tracking based on skin color probability and state estimation model technique for tracking hand gestures [47]. Another interesting



approach is the proposed hand tracking method of Hee et al which allows hand tracking in cluttered background by combining color and motion cues [48].

Majority of the mentioned approaches depend largely on using skin color information in tracking the hand. While this proves to be valid, this kind of approach is still hindered by a number of limitations. In a recent paper by Mazumdar et al, they compared the performance of two color-based hand tracking methods: one uses skin color while the other involves wearing a colored glove on the hand to be tracked [45]. While both methods work accurately under plain background, only the gloved hand tracking method proves to be robust under other background conditions. The bare hand tracking method's performance suffers under complex background and dynamic lighting conditions. This is largely because skin color-based tracking is very sensitive to varying lightning conditions and presence of skin-colored objects in the scene.

An interesting new approach to hand tracking is through the use of distinct features on the hand as proposed by Pan et al [49]. In their method, they select a number of good features to track within the hand and track them onto the next frame. A set of criteria is used for removing unwanted features based on Bayesian skin color probability, velocity and pixel distance from other features. The set of features to be tracked are continuously replenished to maintain continuous hand tracking. While this method still uses skin color information, it marks a transition from color-based to feature-based hand tracking.

## 2.8 Research Gap

Image processing and computer vision continues to find its way in special applications such as computer-aided diagnosis (CAD) and computer-aided instruction. In relation to breast cancer detection, computer-vision based analysis of



mammograms, for example, has remained a prominent area of research throughout the years. However, there has been very minimal progress so far in applying computer vision to BSE which is another important method of finding breast abnormalities.

The studies of Zeng et al and of BIOCORE which started around 1996 have seen minimal progress within the last 9 years. Though the algorithms they have developed looked promising, these are already very outdated in the context of computer vision which has seen rapid development over the recent years. Far more complex and efficient techniques have been developed since with some examples presented in this chapter. This thesis aims to develop newer and more efficient algorithms to be applied on a BSE instruction and supervision system with the recent advancements in computer vision techniques as basis.

In relation to the previous thesis summarized in Section 2.6, this thesis aims to improve from it by developing algorithms which use different approaches. The primary goal of which is to address specific areas for improvement present in the previous thesis. These are the following:

1. The method used for estimating the width of the breast area is based upon inter-nipple distance which is not a reliable source of information for estimating breast width due to factors explained in [50].
2. Ratio and proportion is the primary method used for estimating the height of the breast area. This is based on the height of the ROI detected by the AdaBoost classifier. However, there is no direct relationship between the detected ROI and the recommended BSE breast area coverage.



3. The consulting pathologist recommended the use of a quadrant system for dividing the breast area because this is the standard method recognized by most doctors.
4. Palpation detection was done through thresholding the magnitude of optical flow vectors only. Information such as motion direction and motion trajectories were not used to evaluate palpation.
5. Audio feedback and a graphical user interface (GUI) were not included.



## Chapter 3

### Theoretical Framework

#### 3.1 Computer Vision Fundamentals

Computer vision is a scientific and technological discipline concerned with the theory behind artificial systems which acquire, process, analyze, and understand information from images [51]. The trend in this field is to duplicate or emulate human vision capabilities through electronic means along with geometry, physics, statistics, and machine learning [52].

##### 3.1.1 Digital Image



Figure 3.1.1. Sample digital image (left) and array of integer component values for the 10x10 pixel region (right)



A digital image is a numeric representation of a two-dimensional image and is comprised of a finite set of digital values called pixels (picture elements). The pixels are stored as a two-dimensional array of small integers or a raster map. A raster map or bitmap is characterized by the width and height of the image in pixels (resolution), and by the number of bits per pixel (color depth) which defines the number of colors each pixel can represent. A digital image may have more than 1 channel per pixel. A colored image usually has 3 channels as shown in the figure above while a grayscale image has only one channel.

### 3.1.2 Digital Video

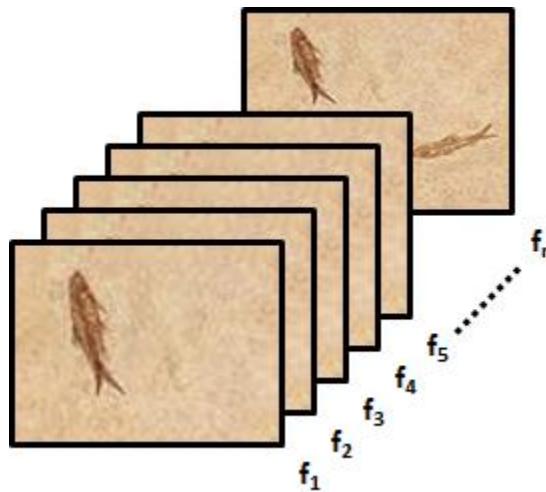


Figure 3.1.2. Digital video represented as a sequence of digital images or frames

A digital video is comprised of a series of bitmap digital images displayed in sequence at a fixed rate. Each digital image is referred to as a frame. A digital video has two important properties, its frame rate and its frame size. The frame rate is the rate at which frames are displayed and is measured in terms of frames per second (fps). The frame size is basically the same with the bitmap image resolution and is measured in pixels.



## 3.2 Color Models

Pixels in a bitmap image or video frame can be represented in different ways. The most common representation is the three color component RGB model. Different models however are utilized in computer vision to suit the specific operations required in different applications. Presented below are some of the most commonly used color models which will be utilized in this study.

### 3.2.1 RGB



Figure 3.2.1. Sample image and its RGB components shown as intensity images: original image (a), red component (b), green component (c), and blue component (d)

In the RGB color model, each pixel is represented by three values which correspond to the three additive colors red, green, and blue. The three values are added together to reproduce a wide array of colors. Each component value is typically ranging from 0 to 255 for a 24-bit encoding.



### 3.2.2 Grayscale



(a)

(b)

Figure 3.2.2. (a) Original colored image and (b) its grayscale image counterpart

The grayscale model, also known as black-and-white, is a model where each pixel is comprised of only one value. Images in grayscale are composed of various shades of gray from black to white. For 8-bit grayscale images, each pixel has a value ranging from 0 to 255. Color images in the RGB model can be converted to grayscale in different ways. One such method used in video systems, called Luma coding, based on the ITU-R BT.709 standard uses the formula below [53] where  $I$  is the grayscale pixel value,  $R$  is the red pixel component,  $G$  is the green pixel component, and  $B$  is the blue pixel component.

$$I = 0.2126(R) + 0.7152(G) + 0.0722(B)$$

Equation 3.2.1



### 3.2.3 HSV

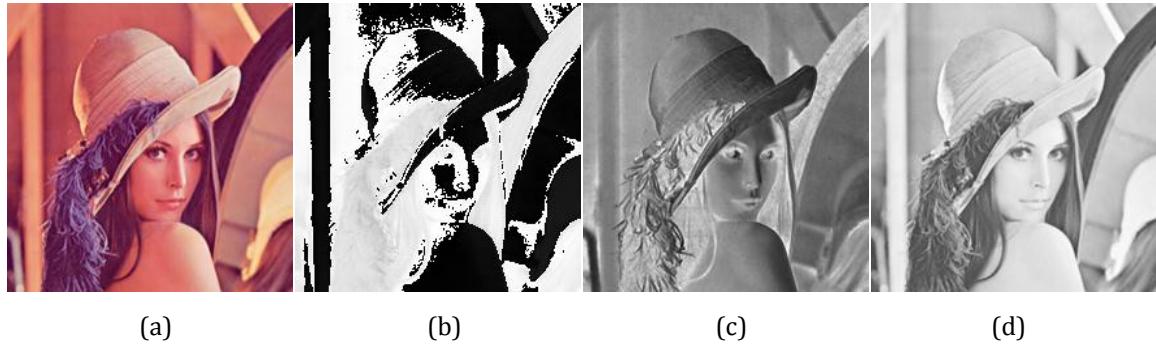


Figure 3.2.3.

Sample image and its HSV components shown as intensity images: (a) original image, (b) Hue component, (c) Saturation component, and (d) Value component

HSV stands for Hue, Saturation, and Value. Hue is defined as the degree to how similar or different a stimulus is from stimuli that are described as red, green, blue, and yellow [54]. Saturation refers to the colorfulness of a color relative to its own brightness [54]. Value, or brightness, refers to how an area appears to emit more or less light [55]. The conversion from RGB to HSV follows a procedure involving the formulas below.

$$M = \max(R, G, B) \quad \text{Equation 3.2.2}$$

$$m = \min(R, G, B) \quad \text{Equation 3.2.3}$$

$$C = M - m \quad \text{Equation 3.2.4}$$

$$H' = \begin{cases} \text{undefined}, & \text{if } C = 0 \\ \frac{G-B}{C} \bmod(6), & \text{if } M = R \\ \frac{B-R}{C} + 2, & \text{if } M = G \\ \frac{R-G}{C} + 4, & \text{if } M = B \end{cases} \quad \text{Equation 3.2.5}$$

$$Hue = 60^\circ \times H' \quad \text{Equation 3.2.6}$$



$$Saturation = \begin{cases} 0, & \text{if } C = 0 \\ \frac{C}{M}, & \text{otherwise} \end{cases} \quad \text{Equation 3.2.7}$$

$$Value = M \quad \text{Equation 3.2.8}$$

### 3.2.4 Binary



Figure 3.2.4. Sample binary images derived from (a) the original image: (b) thresholded image and (c) image showing detected edges

A binary image is a digital image comprised of pixels which can have only two possible values. Typically, binary images are rendered as black and white. Each pixel in a binary image is stored as a single bit, i.e. 0 or 1. Binary images are produced through certain image processing operations such as segmentation, thresholding, etc.



## 3.3 Image Processing

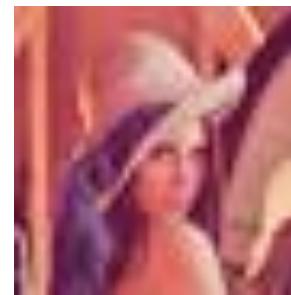
### 3.3.1 Image Scaling



(a)



(b)



(c)

Figure 3.3.1. Example of image scaling: (a) original image and downscaled version using a factor of 0.2 shown in its (b) actual size and (c) stretched to original size

Image scaling is the process of resizing digital images. Scaling involves a trade-off between efficiency, smoothness, and sharpness. In this study, video frames are scaled down to decrease processing time and meet the necessary frame rate for real-time application. Image scaling is done through different methods. The most commonly used methods are nearest-neighbor interpolation, bilinear interpolation, bicubic interpolation, etc.



### 3.3.2 Region of Interest

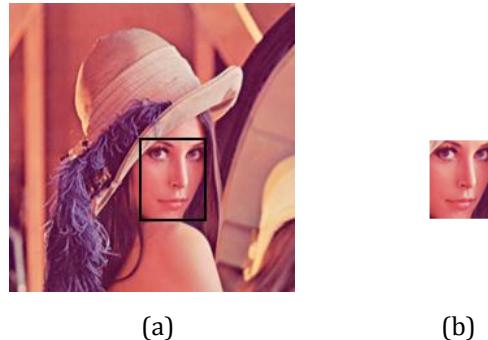


Figure 3.3.2. (a) Original image with ROI selected and (b) the extracted ROI

A region of interest (ROI) is a subset of pixels within a digital image selected either manually or automatically. An example of an ROI, in object detection, is the area within the boundaries of an object. In computer vision, defining an ROI is used in limiting the analysis to one specific area either to increase computational efficiency or to minimize detection of features not relevant to the analysis. The ROI is most commonly annotated using a bounding box to simplify the selection process but an ROI can be defined through any closed polygonal selection.

### 3.3.3 Thresholding

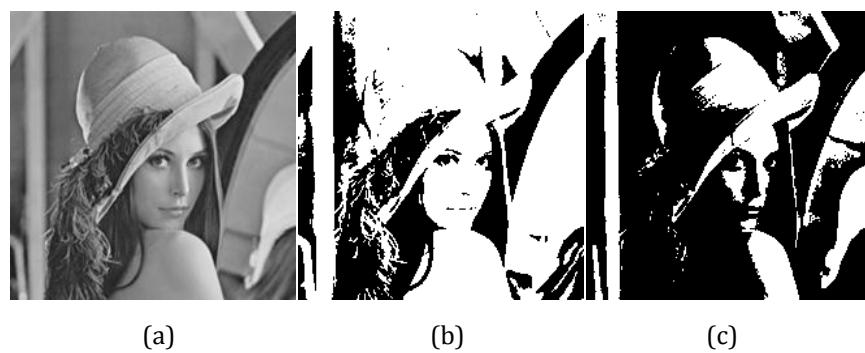


Figure 3.3.3. (a) Sample grayscale image, thresholded image using a threshold of (b) 0.4 and (c) 0.6



Image thresholding is one method of image segmentation. The common product of this process is a binary image, though thresholding can also be used for truncation purposes. Thresholding is usually applied on a single-channel image, e.g. grayscale image, or on one channel of a multi-channel image such as the Hue channel of an HSV image [51]. An application of thresholding in this study is in skin segmentation. In here, an upper threshold and a lower threshold are used in the Hue channel of an image in the HSV color space to retain only the pixels within the range of hue of the skin.

### 3.3.4 Smoothing



(a)

(b)

Figure 3.3.4. (a) Sample grayscale image and (b) filtered image using median filter

Smoothing is an image processing technique used either to lessen the image's pixelation or to reduce noise within an image. One of the most common algorithms and the simplest one to smooth images is the normalized box filter. The algorithm uses a rectangular window wherein each pixel in the image is replaced with the average of adjacent pixels around it and within the rectangular window. Another common algorithm is the median filter which uses median values instead of mean values. Other algorithms use weighted windows with various methods of approximating the weights at the expense of computational cost.



## 3.4 Feature Extraction

Feature extraction involves algorithms which produce from an input image a set of features which provide relevant information for further analysis. Feature extraction in the field of image processing involves detection of corners, edges, blobs, curves, shapes, etc.

### 3.4.1 Shi-Tomasi Corner Detection

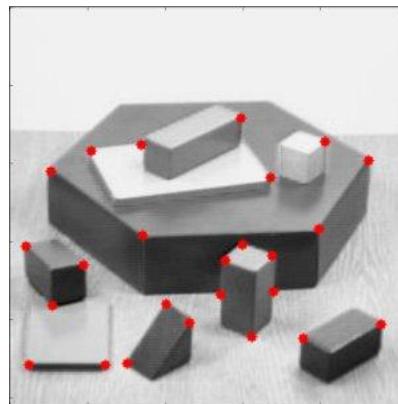


Figure 3.4.1. Sample corner detection result: red dots correspond to the detected corners [56]

A corner is formally defined as either the intersection of two edges or a point for which there are two dominant and different edge directions in the local neighborhood. The most commonly used corner detection algorithm today is the Shi-Tomasi corner detector. This algorithm is an improvement to the method proposed by Harris and Stephens which in turn is an improvement to a much earlier method proposed by Moravec.

Moravec's method defines a corner to be a point with low self-similarity [57]. Moravec's algorithm measures the similarity of a patch centered on the pixel of interest to nearby overlapping patches. Harris corner detection improves on this by considering the differential of the corner score with respect to direction directly



instead of overlapping patches [58]. Shi-Tomasi corner detection is very much similar to Harris corner detection save from a different scoring function [59].

### 3.4.2 Canny Edge Detection



(a)

(b)

Figure 3.4.2. (a) Original image and (b) binary image showing the detected edges [59]

An edge refers to a set of points in a digital image at which the image brightness has discontinuities. These sharp changes in brightness correspond to discontinuities in depth, discontinuities in surface orientation, changes in material properties, and variations in scene illumination [60]. Edge detection refers to a set of mathematical methods aimed at identifying edges in a digital image. An optimal edge detection algorithm has the following three properties [61]:

- *Low error rate* – the algorithm should mark as many real edges as possible and minimize detection of non-existent edges
- *Good localization* – detected edges are as close as possible to the real edges in the real image
- *Minimal response* – a specific edge in the image triggers only one response from the detector



Canny edge detection, known as the optimal edge detector, aimed to satisfy the three main criteria above. It is comprised of four main stages namely: noise reduction, finding the intensity gradient, non-maximum suppression, and hysteresis [61]. First, noise is reduced by convolving a Gaussian filter with the original image. Second, the intensity gradient of the image in four directions is computed in order to detect  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$  edges. Third, non-maximum suppression is applied to remove pixels which are not considered to be part of an edge so that only thin lines remain. The last stage is the hysteresis stage which uses an upper and a lower threshold to finalize which pixels correspond to edges and which don't.

### 3.5 Haar-like Features

Haar-like features are simple digital image features developed by Paul Viola and Michael Jones [62]. A Haar-like feature takes the difference between the sums of pixel intensities within adjacent rectangular regions. This is used to classify subsections of an image. The use of these Haar-like features became widespread due to its calculation speed being a key advantage over other types of image features. A Haar-like feature of any size is calculated in constant type through the use of integral images.

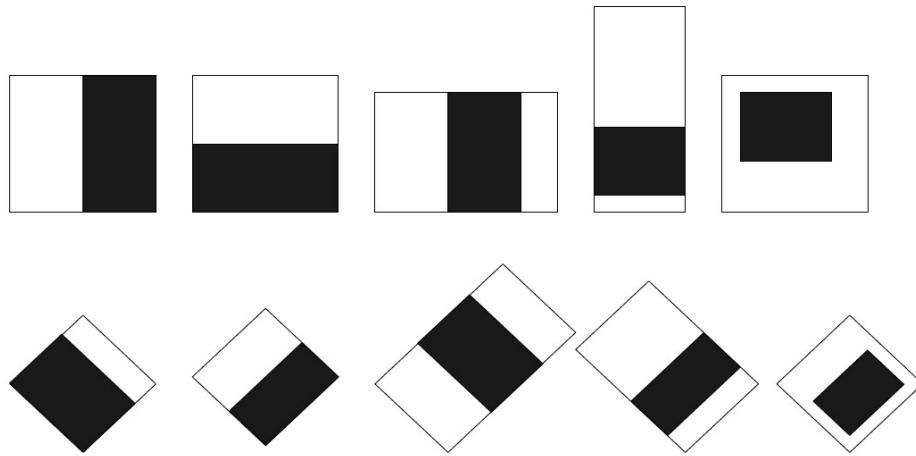


Figure 3.5.1

Examples of regular Haar-like features (top) and tilted Haar-like features (bottom) [63]

## 3.6 Integral Image

An integral image, also known as a summed area table, is an algorithm for fast and efficient calculation of sum of pixel values in a rectangular region of a digital image. This algorithm was first introduced in 1984 but was first prominently used in 2001 by Viola and Jones [62]. The integral image of a given digital image is calculated through a single pass using the equation below where  $i(x, y)$  is the value of the pixel for the given coordinates and  $I(x, y)$  is the corresponding value in the integral image.

$$I(x, y) = i(x, y) + I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1) \quad \text{Equation 3.6.1}$$

Basically, the value of any element in an integral image is just the sum of pixel values above it and to the left of it. Once the integral image of a given image has been calculated, summing up pixel values within any rectangular section of that image can be accomplished in constant time using four look-ups through the equation below where A, B, C, and D are the coordinates of pixels as shown in Figure 3.6.1.

$$\sum i(x, y) = I(C) + I(A) - I(B) - I(D) \quad \text{Equation 3.6.2}$$

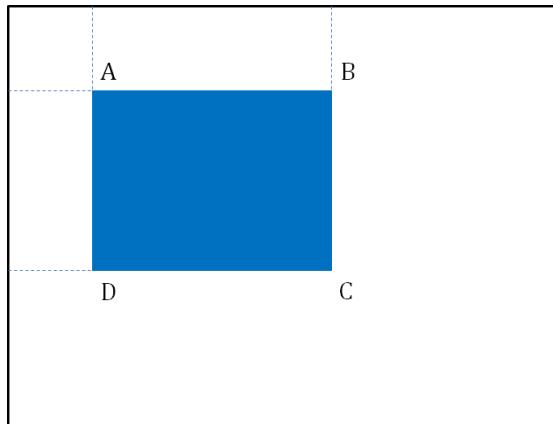


Figure 3.6.1 Calculating the sum of a rectangular section of an image using Integral Image

## 3.7 Genetic Algorithm

Genetic algorithm (GA) is a search heuristic modeled according to the process of natural selection. GAs are most often used in optimization and search problems using techniques or operations inspired by sub-processes involved in natural selection such as selection, crossover, and mutation.

### 3.7.1 Representation

A population of possible solutions to an optimization problem is evolved toward optimal solutions using a number of operations. Each candidate solution is represented by its chromosome which is comprised of a number of genes which can be altered. The basic representation of a chromosome is an array of bits wherein each bit corresponds to a single gene. In this thesis however, instead of an array of bits, a chromosome is represented as an array of floating point numbers.

The evolution process revolves around a population of possible solutions to the problem which comprise a single generation. A solution here is analogous to a chromosome. Over the course of the evolution process, the chromosomes in the



population are either altered or replaced. For each generation, a unique population is produced.

### 3.7.2 Initialization

At the very beginning of the evolution process, a number of possible solutions are randomly generated to form the initial population. The population size is highly dependent on the type of problem. Usually, a population contains hundreds or even thousands of possible solutions. In this thesis however, a population size of 30 proved to be enough to solve the specific task presented in Section 4.5.3.

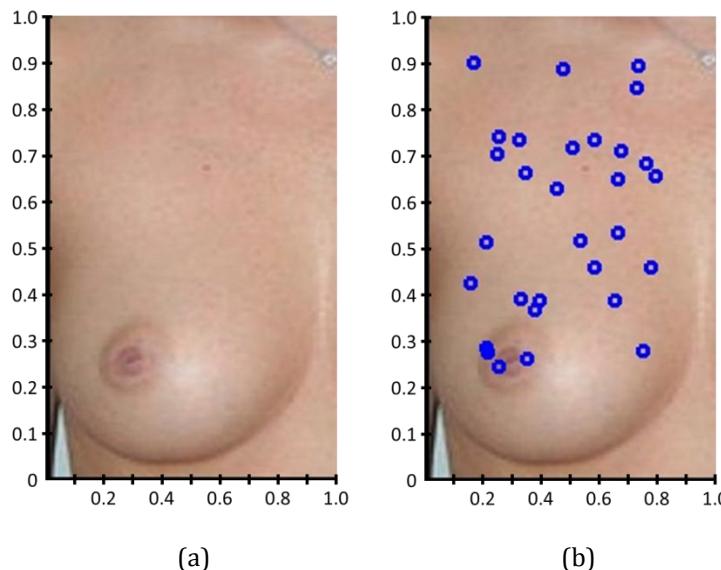


Figure 3.7.1. Sample 2-dimensional search space of the breast area (a) and the initial population with randomly generated solutions [blue circles] (b)

Shown in Figure 3.7.1(a) is a sample problem in this study which can be solved using genetic algorithm. The problem in specific is to locate the nipple within the breast area. In this example, the problem is treated as a 2-dimensional search space which can be solved by evolving a set of candidate solutions characterized by



chromosomes which have genes corresponding to normalized X and Y coordinates within the range of values from 0 to 1. Figure 3.7.1(b) shows the initial population of candidate solutions. Due to their being randomly generated, they are scattered all over the search space.

### 3.7.3 Selection

For each generation, the fitness of each candidate solution is evaluated using a fitness function. The fitness function is formulated in order to solve a specific problem. Candidates are stochastically selected to form the next generation. In the implementation used in this thesis, the probability of selection is based on a cumulative ratio of individual fitness to the total fitness of population. This way, solutions which have higher fitness values are more likely to get selected to survive onto the next generation. The set of selected candidates is referred to as the elite pool.

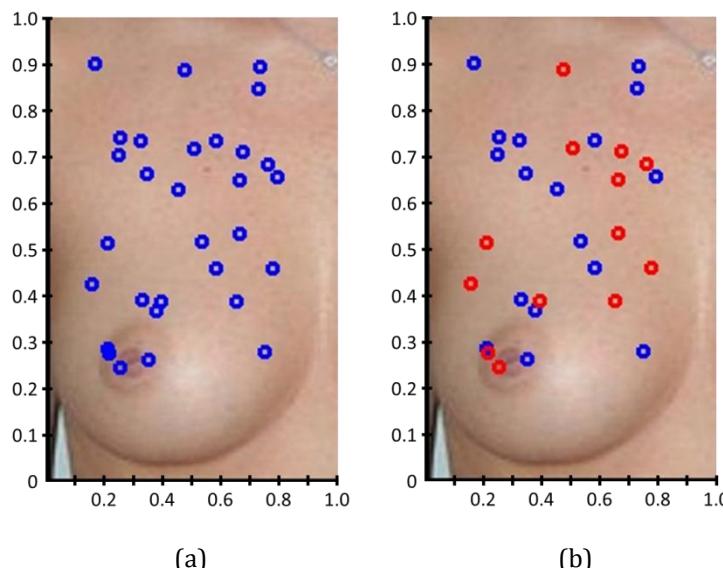


Figure 3.7.2. Randomly generated solutions [blue circles] after the initialization phase (a) and solutions selected to form the elite pool [red circles] after the selection phase (b)



In the sample problem, candidate solutions are randomly selected, with bias to those which have higher fitness values based on a fitness function to be described in Section 4.5.2, to form the elite pool. They are shown as red circles in Figure 3.7.2(b).

### 3.7.4 Crossover

In the crossover stage, pairs of ‘parent’ solutions are selected to produce ‘child’ solutions which contain a mixture of the characteristics of their parents. Generally, genetic algorithms are not limited to using 2 parents only in crossover. Some implementations use three or more parents to produce a single child. In this thesis however, only 2 parents are utilized.

There are a number of crossover techniques used in genetic algorithms. An example is the “one-point crossover” wherein a single crossover point on both parent solutions is selected and data beyond this point are swapped in order to produce the children solutions. Other techniques are the “two-point crossover” and the “uniform crossover”. In this thesis, a variation of the one-point crossover technique is utilized. However, instead of producing two children solutions from this crossover technique, only the first child is taken while the second child is a clone of the parent solution with a higher fitness value. This way, solutions with high fitness values will not be lost over the course of the GA.

In the sample problem, crossover is achieved by having the two crossover children exchange X and Y genes from their parents. They are shown in Figure 3.7.4(b) as green circles.

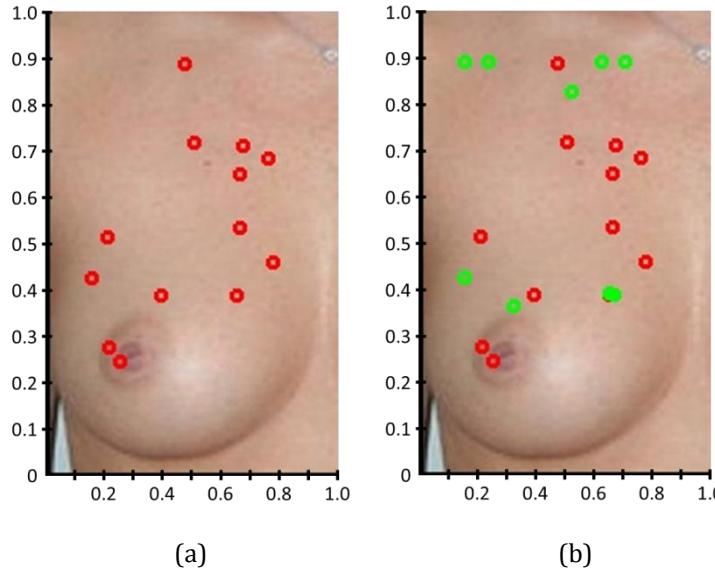


Figure 3.7.3. The elite pool of candidate solutions [red circles] after the selection phase (a) including the crossover children [green circles] after the crossover phase (b)

### 3.7.5 Mutation

To maintain genetic diversity within a population, a mutation stage is included in genetic algorithms. In here, one or more genes in a chromosome are altered by changing their values. Mutation occurs randomly for each gene of each chromosome depending on a user-defined probability of mutation. The resulting set of solutions after this phase constitute the next generation. In this thesis, mutation is applied by adding or subtracting a randomly generated integer to each digit of each gene of each chromosome.

In the sample problem, the probability of mutation is set to 0.3. This relatively high value is used to generate lots of mutated children. This will ensure genetic diversity among the candidate solutions. Mutated children are shown in Figure 3.7.4(b) as yellow circles.

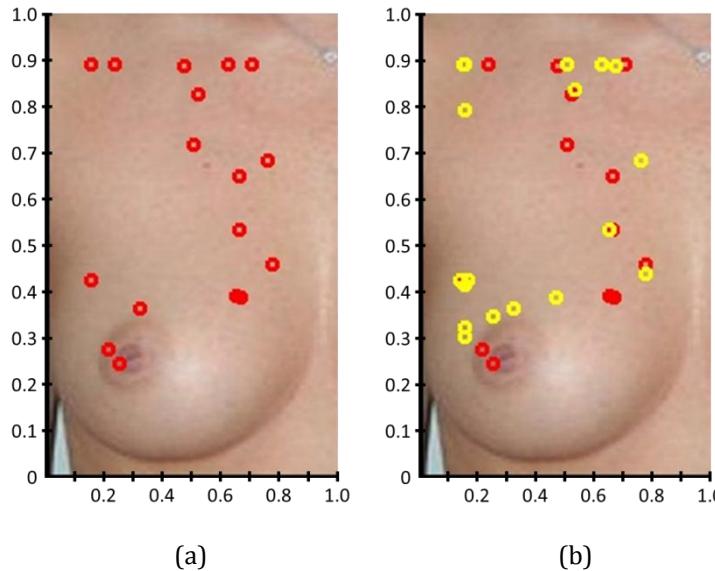


Figure 3.7.4. The current population [red circles] after the crossover phase (a) and the mutated children [yellow circles] after the mutation phase (b)

### 3.7.6 Termination

The evolution process is stopped whenever a termination condition or a combination of termination conditions is satisfied. Some of the most common termination conditions are if a fixed number of generations is reached, the highest ranking solution's fitness has reached a threshold, the overall fitness of the entire population has reached a threshold, etc. In this thesis, only the fixed number of generations is used as a terminating criterion.

In the sample problem as shown in Figure 3.7.5, after 20 generations, most of the candidate solutions have already evolved towards the location of the nipple. This constitutes a successful run of the genetic algorithm.

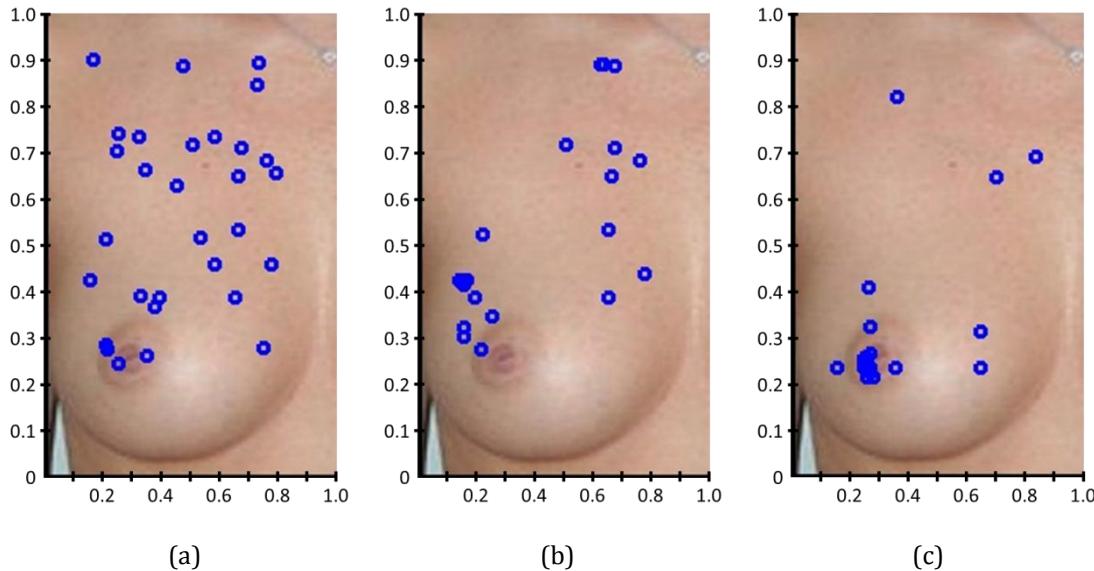


Figure 3.7.5. Evolution of the candidate solutions: initial population (a), population after 5 generations (b), and the final population after 20 generations (c)

The sample problem presented here is treated as a simple 2-dimensional search problem using two parameters which are the X and Y coordinates. Searching the nipple in various input images however is not easily solved this way. This complicated problem requires a more complicated approach such as adding another parameter to use, thereby increasing the dimensionality of the problem. All these, together with the actual fitness function used, are further discussed in Chapter 4.

## 3.8 Optical Flow

Optical flow is the pattern of apparent motion of different elements in a visual scene caused by relative motion between an observer (camera) and the scene [63]. Optical flow methods try to calculate the motion between two successive video frames which are taken at times  $t$  and  $t + \Delta t$ . Motion is calculated either for every



pixel position (dense optical flow) or for a selected set of pixels (sparse optical flow).

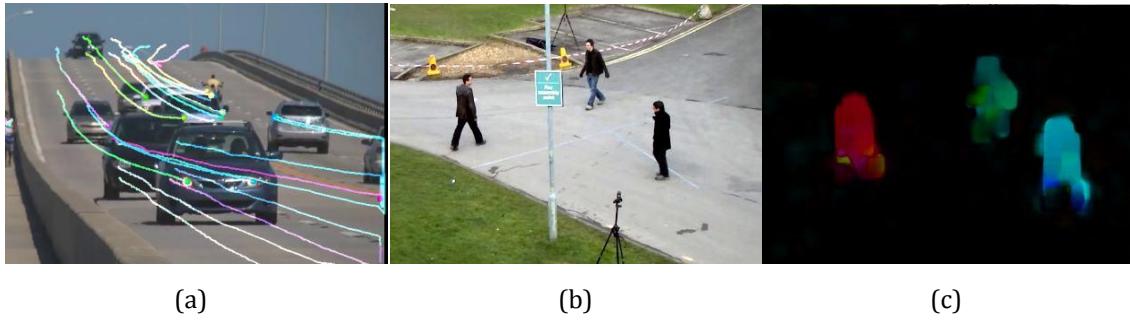


Figure 3.8.1. Optical flow examples: (a) cars on a highway with sparse optical flow results, and (b) pedestrians on a street with (c) dense optical flow results [62]

Assuming that the brightness of pixels is constant for the two successive frames, the following equation can be formulated for a pixel at  $(x, y, t)$  with an intensity  $I$  and displacement of  $\Delta x$ ,  $\Delta y$ , and  $\Delta t$ .

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad \text{Equation 3.8.1}$$

By taking the Taylor series approximation of the right-hand side of the equation, the result is otherwise known as the optical flow equation where  $I_x$  and  $I_y$  are image gradients,  $I_t$  is the gradient along time, while  $V_x$  and  $V_y$  are the  $x$  and  $y$  components of the velocity or optical flow.

$$I_x V_x + I_y V_y = -I_t \quad \text{Equation 3.8.2}$$

The optical flow equation has two unknowns which makes it unsolvable. This is also known as the aperture problem of optical flow algorithms. Different optical flow algorithms address this problem by introducing additional constraints in order to come up with additional sets of equations. This allows for the solution of the aperture problem. One such algorithm is known as the Lucas-Kanade optical flow.



### 3.8.1 Lucas-Kanade Optical Flow

The Lucas-Kanade method is a famous differential method for estimating the optical flow. It holds on the assumption that the flow is essentially constant in a local neighborhood of a specific pixel as shown in the equation below where  $I_x$  and  $I_y$  are image gradients,  $I_t$  is the gradient along time, and  $q_1, q_2$ , and  $q_n$  are pixels inside the local neighborhood.

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\ \vdots &\quad \vdots \\ I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n) \end{aligned} \tag{Equation 3.8.3}$$

This set of equations can be rewritten in matrix form as shown below.

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix} \tag{Equation 3.8.4}$$

$$v = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \tag{Equation 3.8.5}$$

$$b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \tag{Equation 3.8.6}$$

$$Av = b \tag{Equation 3.8.7}$$

The Lucas-Kanade method proceeds on solving these basic optical flow equations for all the pixels in that neighborhood through the least squares criterion as shown below where  $V_x$  and  $V_y$  correspond to the x and y components of the optical flow.

$$v = (A^T A)^{-1} A^T b \tag{Equation 3.8.8}$$



$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum I_x(q_i)^2 & \sum I_x(q_i)I_y(q_i) \\ \sum I_y(q_i)I_x(q_i) & \sum I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_x(q_i)I_t(q_i) \\ -\sum I_y(q_i)I_t(q_i) \end{bmatrix} \quad \text{Equation 3.8.9}$$

### 3.8.2 Kanade-Lucas-Tomasi Feature Tracker

The Lucas-Kanade method uses a sparse feature set, i.e. selected pixels within an image, to compute the optical flow. Kanade and Tomasi further improved this method by using only the features which are suitable for tracking to compute the optical flow [66]. In their proposed improvement, a feature is selected if both of the eigenvalues of the gradient matrix were larger than a predefined threshold. This implementation is also known as the feature extraction technique called KLT tracker.

This improved version explains that the Lucas-Kanade optical flow works best by tracking strong features which are abundant in textured regions. This characteristic makes it suitable in tracking the hand in BSE which is considerably more textured as compared to the rest of its background which is the breast.



Figure 3.8.2. The Lucas-Kanade optical flow/KLT tracker applied on the hand tracking problem



Thus, in this thesis, a modified KLT tracker is implemented by detecting suitable features using Shi-Tomasi corner detection and tracking these features by calculating their optical flow using Lucas-Kanade optical flow. Originally intended for solving the problem of image registration, the modified KLT tracker implemented in this thesis is used as a technique for estimating hand motion or displacement frame per frame. In other words, this modified version is used for continuous hand tracking.



## Chapter 4

### Methodology

#### 4.1 Experiment Setup

The general idea regarding the setup of experiments to be conducted while developing and testing the individual algorithms is summarized in the figure below. The setup is comprised of 4 major components namely, the BSE, input device, processing unit, and feedback mechanism.

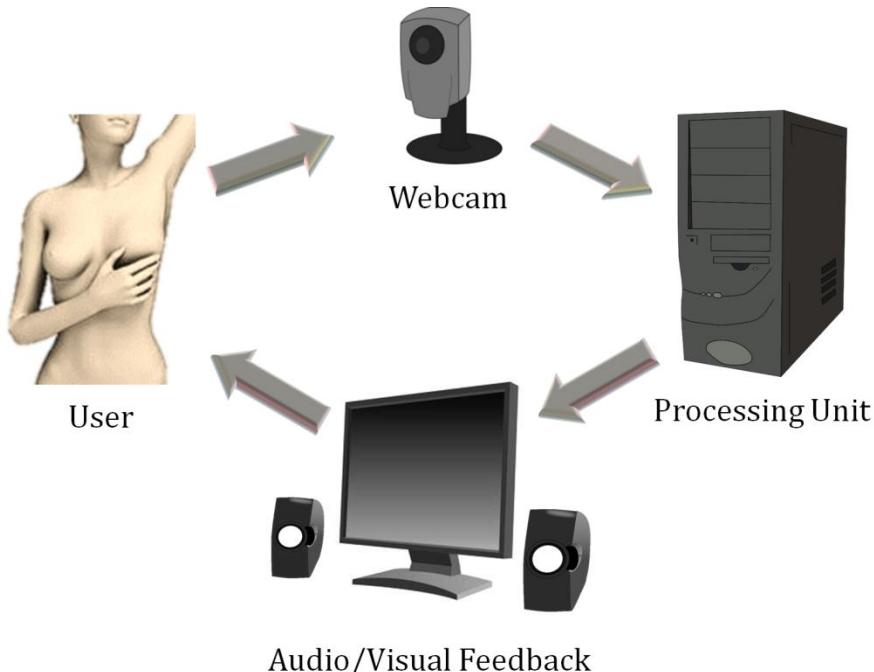


Figure 4.1.1. Block diagram of the experimental setup of the BSE supervision system



The BSE component corresponds to the user performing breast self-examination. The performance is captured using a simple webcam. This is then fed input to the processing unit in which the computer vision algorithms developed process the input to determine the necessary feedback to supply to the user. The feedback is sent to the user via audio and visuals (text and graphics).

## 4.2 Important Assumptions

The individual algorithms and the entire supervision system as a whole will be developed with the assumption that the following conditions are true for the input data, whether pre-recorded or real-time capture:

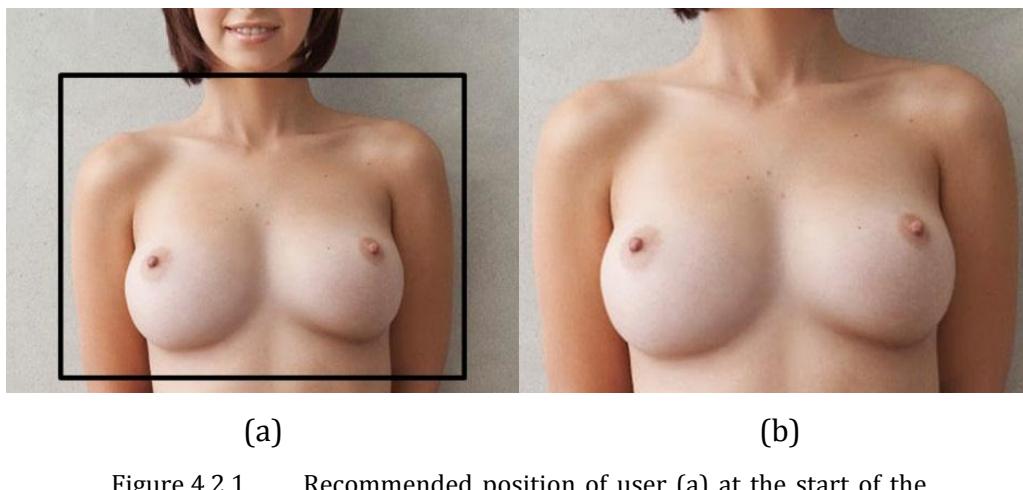


Figure 4.2.1. Recommended position of user (a) at the start of the BSE procedure and the recommended area captured by the webcam (b)

1. The user is positioned in front of the camera as shown in the figure above.
  - a. The user is centered at the center of the camera's capture area
  - b. The user's arms are positioned at the side of her body
  - c. The user is positioned such that the upper boundary of the capture area is just beneath her chin



- d. The user is near enough to the camera such that majority of the capture area is occupied by her body
  - e. The user is distanced enough from the camera such that the entire width of the torso, including the arms, is within the capture area
2. The body of the user remains in this position and is stationary with respect to the camera throughout the BSE procedure except for the hand and arm being used for palpation.
  3. The opposite hand is used for palpating the breasts, e.g. right hand for palpating the left breast and left hand for palpating the right breast.
  4. The user is not wearing any accessories in her hands, e.g. bracelet, watch, etc.
  5. There are no other objects, including the user's hair, in front of the user's torso.

### 4.3 Data Set

The data to be used for the system development phase and testing phase is comprised of three sets. The first set is comprised of 200 images of the female torso downloaded from the internet and extracted from BSE videos. This set will be used for measuring the performance of the breast area detection algorithm and the nipple tracking algorithm. For all images in this data set, ground truth values, comprised of the bounding boxes for the left and right breasts as well as the coordinates of the left and right nipples, were manually annotated.



# De La Salle University



Figure 4.3.1 Sample images of the female torso included in Data Set 1



# De La Salle University

The second data set is comprised of 14 video sequences of women performing BSE. This set is taken from the documented videos of the CHED-PHERNET Breast Cancer Research Group. This data set is used for measuring the performance of the hand tracking algorithm and the palpation detection algorithm. For each video, ground truth coordinates of the hand were manually annotated for each frame.

Video No.	Length (frames)	Video No.	Length (frames)
D2-01	959	D2-08	1,027
D2-02	777	D2-09	461
D2-03	827	D2-10	629
D2-04	750	D2-11	1,330
D2-05	218	D2-12	580
D2-06	206	D2-13	988
D2-07	1,077	D2-14	987

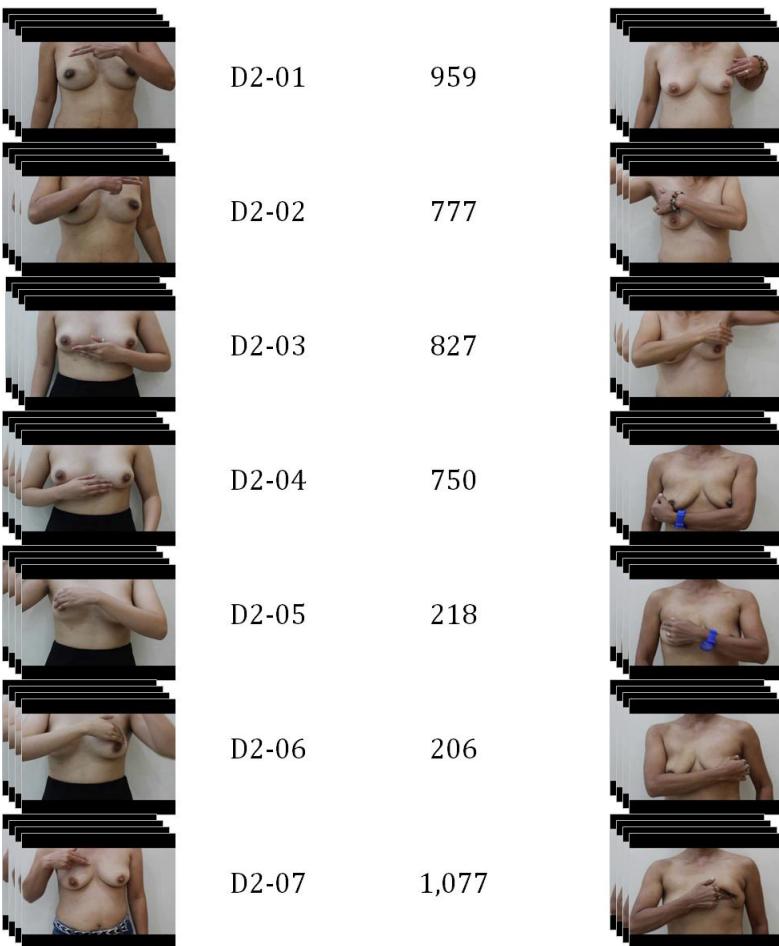


Figure 4.3.2 The 14 video sequences included in Data Set 2



The third data set is comprised of video sequences specifically recorded for this study on a customized mannequin. This data set will also be used for measuring the performance of the palpation detection algorithm. Also included in this data set are complete BSE trials on the same mannequin for evaluating the functional quality of the complete system.

Video No.	Length (frames)	Video No.	Length (frames)
D3-01	293	D3-07	150
D3-02	343	D3-08	150
D3-03	313	D3-09	150
D3-04	334	D3-10	150
D3-05	311	D3-11	150
D3-06	247	D3-12	150

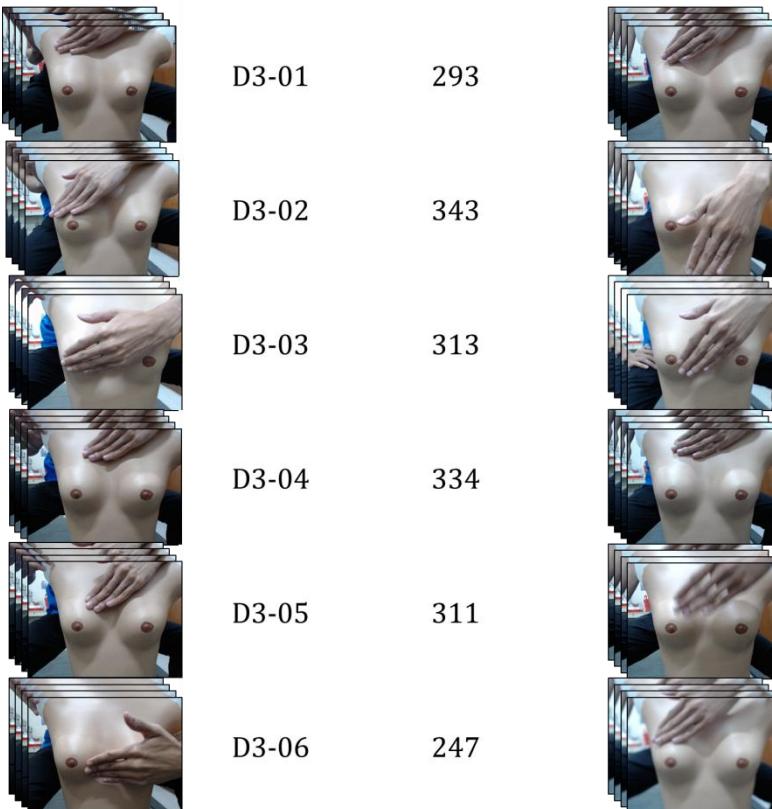
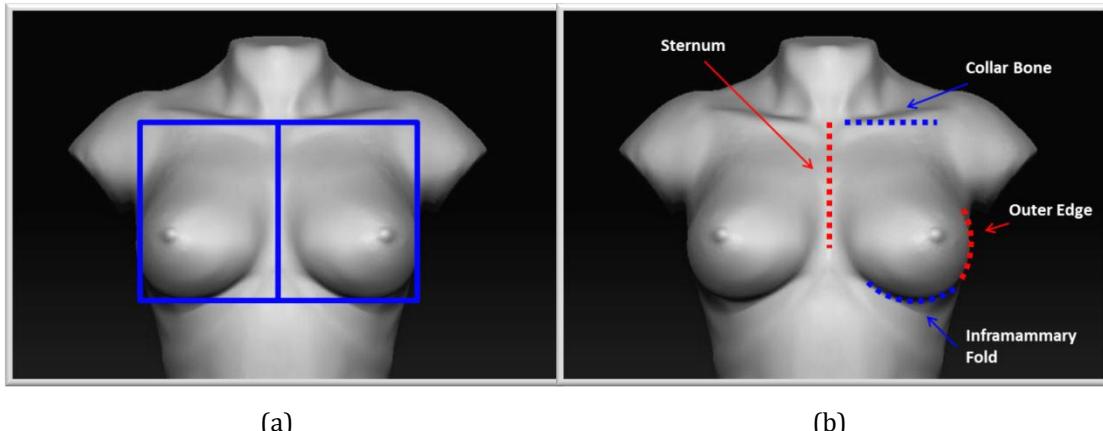


Figure 4.3.3 The 12 video sequences included in Data Set 3



## 4.4 Breast Area Identification

### 4.4.1 Rationale



(a)

(b)

Figure 4.4.1 Recommended area of coverage for BSE for the left and right breasts (a) and visual cues for the automatic delineation (b)

According to [9], a woman must cover the entire breast area from top to bottom and from side to side in performing BSE. Specifically, this spans the area from the collar bone down to the bra line (top of the abdomen) and from the sternum (cleavage) to the armpit (axilla). The algorithm to be presented here will look for 4 specific visual cues as shown in Figure 4.4.1 namely, the collar bone, the inframammary fold, the outer edge of the breast, and the sternum.

### 4.4.2 Framework

The general framework for the breast area identification algorithm, including nipple detection, is to begin the search process within the entire frame and to continuously narrow down the search area until the object of interest is located, i.e. breast area or nipple. Using the assumption that the user/patient is positioned as



shown in Figure 4.4.1, the search process can be effectively done by following the block diagram shown in Figure 4.4.2.

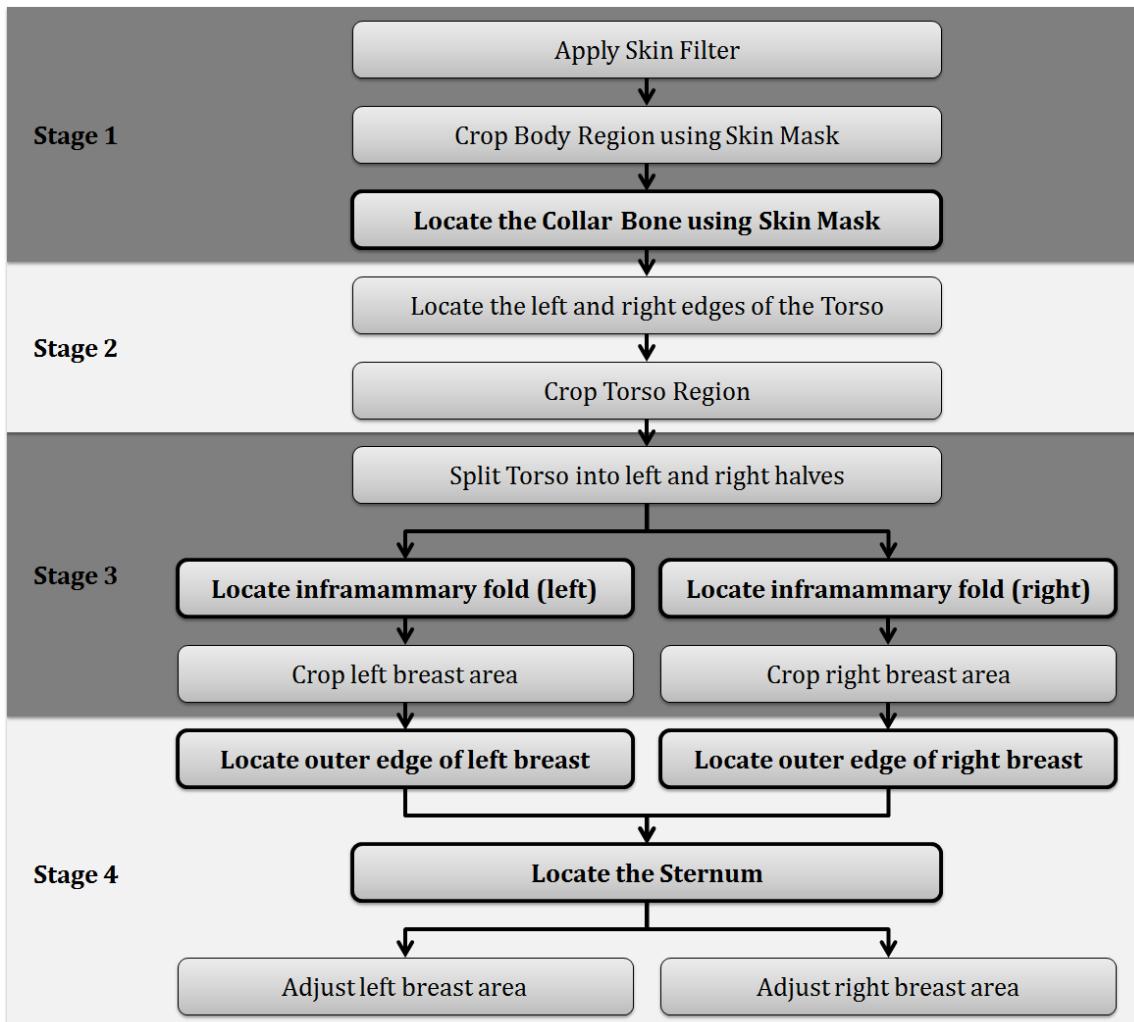


Figure 4.4.2 Block diagram of the Breast Area Identification Framework

### 4.4.3 Stage 1: Skin Segmentation and Locating the Collar Bone

An important preprocessing for the breast area identification algorithm (BAIA) is segmenting the body from the background using skin color information. This is



achieved by using a skin filter. Stage 1 of BAIA includes this part up to the identification of the collar bone location. Stage 1 is summarized in Figure 4.4.3.

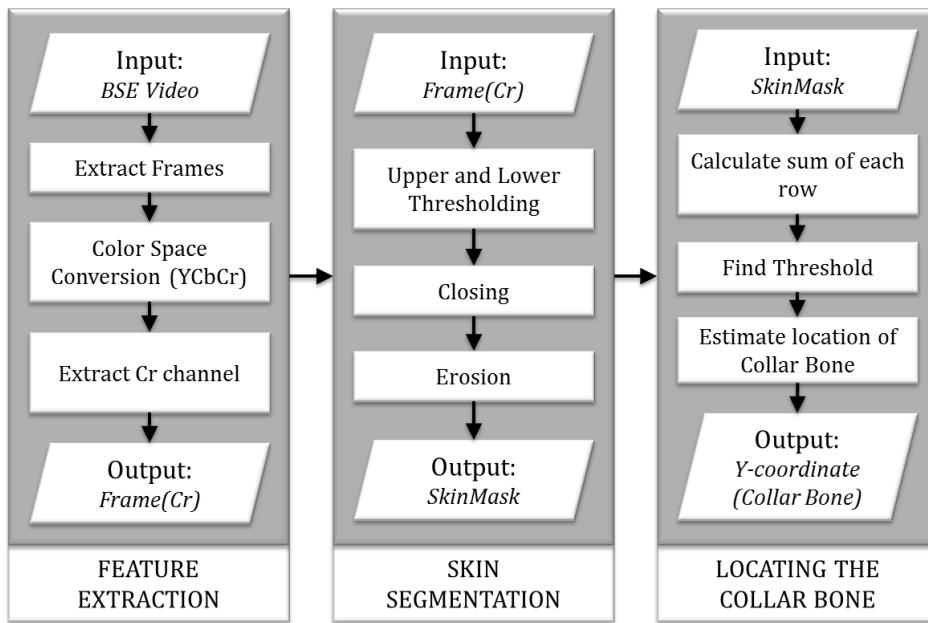


Figure 4.4.3 Block diagram of Stage 1 of the Breast Area Identification Algorithm

Stage 1 starts with converting the original frame into a more suitable color space for skin segmentation. In this study, the YCbCr color space is chosen. The Cr channel (red Chroma component) is extracted and thresholded to determine skin pixels and non-skin pixels. The thresholding process is defined in the relation below where Cr is the pixel value in the Cr channel for the given x and y coordinates and S is the corresponding value in the binary skin mask.

$$\begin{cases} Cr(x, y) < 4, & S = 0 \\ Cr(x, y) > 127, & S = 0 \\ \text{otherwise}, & S = 1 \end{cases} \quad \text{Equation 4.4.1}$$

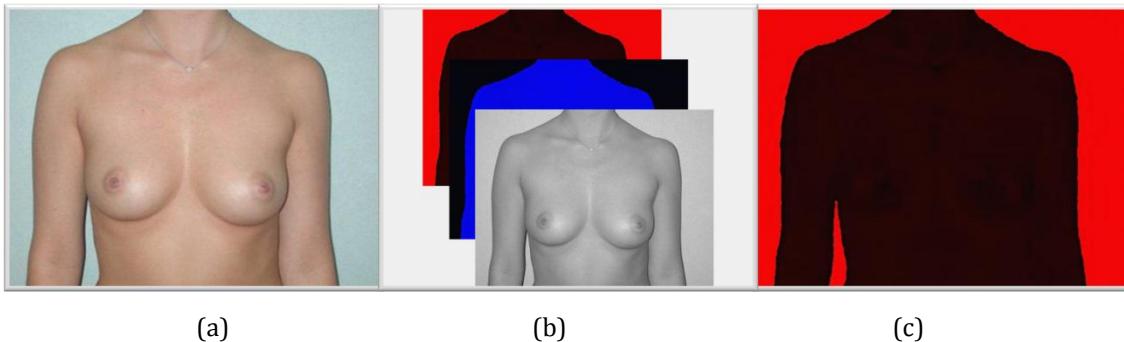


Figure 4.4.4 Feature extraction: input image (a), YCbCr channels (b), extracted Cr channel (c)

Additional morphological operations are performed on the resulting skin mask. First, a closing operator is used to close out small holes in the skin mask. Second, the skin mask is eroded to minimize false positives, i.e. non-skin pixels erroneously classified as skin pixels, along the edges of the body. Figure 4.4.4 shows the skin segmentation process.

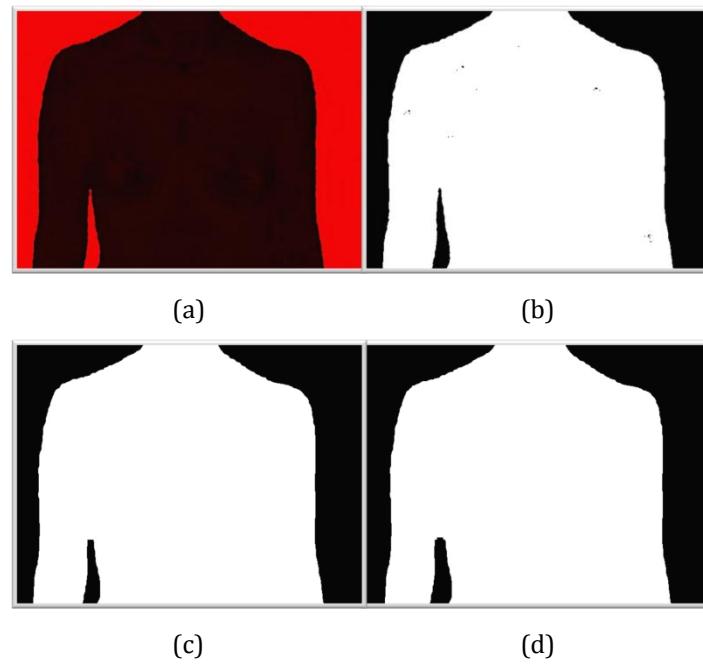


Figure 4.4.5 Skin segmentation process: Cr channel (a), preliminary skin mask (b), skin mask after closing (c), and skin mask after erosion (d)



Using the final skin mask, the location of the collar bone is estimated through ratio and proportion of the body width with respect to the vertical axis. Width measurements are done by counting the number of skin pixels in a row. The body width is defined using a reference width equal to the maximum width of the upper half of the skin region. The collar bone is assumed to be at the vertical level where the skin mask width is 75% of the reference width.

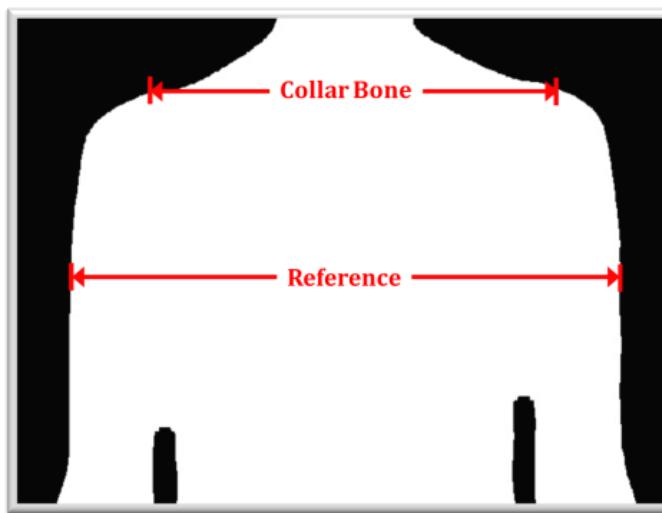


Figure 4.4.6 Illustration of the skin mask reference level and collar bone location



```
void Stage1()
{
    Mat Frame,Y,Cb,Cr,skinmask;

    //Extract frame from webcam
    cap.read(Frame);

    //RGB to YCbCr conversion
    for(int rr=0; rr<480; rr=rr+1)
    {
        for(int cc=0; cc<640; cc=cc+1)
        {
            int b = frame.at<Vec3b>(rr,cc)[0];
            int g = frame.at<Vec3b>(rr,cc)[1];
            int r = frame.at<Vec3b>(rr,cc)[2];
            Y.at<uchar>(rr,cc) = 0.299*r + 0.587*g + 0.11*b;
            Cb.at<uchar>(rr,cc) = b - Y.at<uchar>(rr,cc);
            Cr.at<uchar>(rr,cc) = r - Y.at<uchar>(rr,cc);
        }
    }

    //Upper and lower thresholding
    threshold(Cr,skinmask,127,THRESH_TOZERO_INV);
    threshold(skinmask,skinmask,4,THRESH_BINARY);

    //Closing and Erosion
    morphologyEx(skinmask,skinmask);
    erode(skinmask,skinmask);

    //Calculate summation of each row
    Mat rowsum;
    for(int row=0; row<480; row++)
        for(int col=0; col<640; col++)
            if(skinmask.at<uchar>(row,col) > 0)
                rowsum.at<float>(row)++;

    //Find threshold
    int sumthres = 0;
    for(int row=0; row<480; row++)
        if(rowsum.at<float>(row) > sumthres)
            sumthres=rowsum.at<float>(row);

    //Locate collar bone
    for(int row=100; row>=0; row--)
        if(rowsum.at<float>(row) < sumthres*0.75)
            torso.y = row;
}
```

Figure 4.4.7 Simplified code for Stage 1 of the Breast Area Identification Algorithm



As shown in Figure 4.4.7, the color space conversion from RGB to YCbCr is implemented manually. Upper and lower thresholding is achieved using the built-in *threshold( )* function in OpenCV while closing and erosion are applied using the functions *morphologyEx( )* and *erode( )* respectively. The sum of each row is calculated by passing through every pixel of each row and storing the sums in the matrix *rowsum* with a length equal to the number of rows. The threshold *sumthres* is determined by getting the highest sum in *rowsum*. Finally, the location of the collar bone is determined by comparing each value in *rowsum* with 75% of *sumthres*.

#### 4.4.4 Stage 2: Identifying the Torso Region

After segmenting the body from the background, the next logical step is to determine the region of the body and focus further processing on this region-of-interest (ROI). Stage 2 of the Breast Area Identification Algorithm is summarized in Figure 4.4.8.

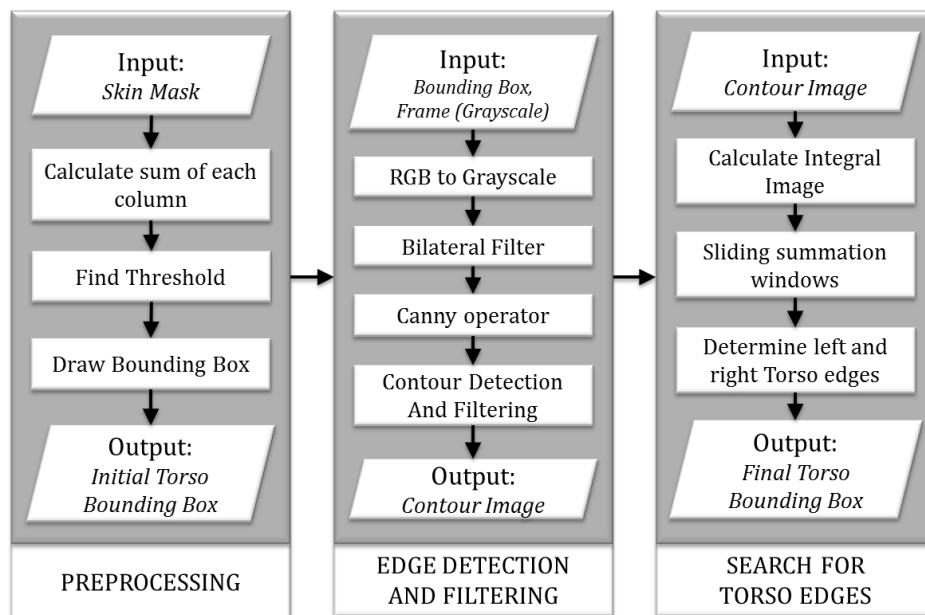


Figure 4.4.8 Block diagram of Stage 2 of the Breast Area Identification Algorithm



A similar method to the one used in locating the collar bone is used to crop the ROI on both left and right sides. The two sides are cropped to remove the outer edges of both arms in preparation for the next sub-stage. Using the resulting bounding box, the original frame is cropped; the resulting image would now contain only the region within the bounding box.

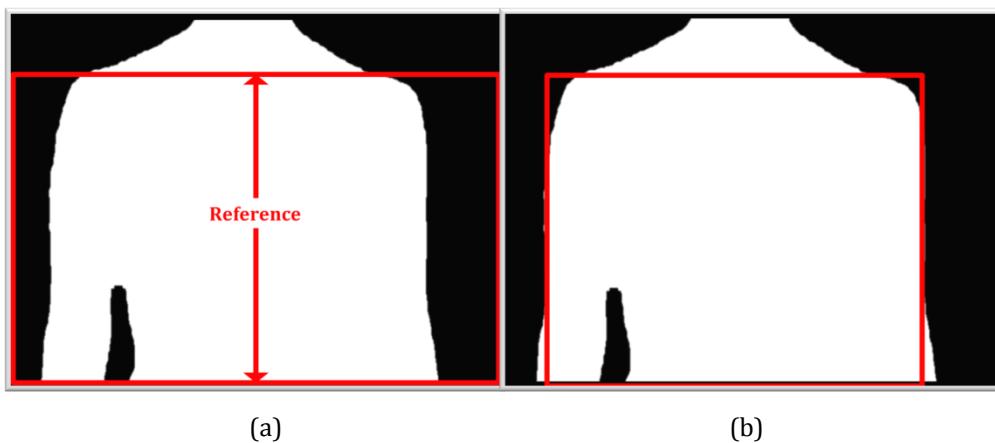


Figure 4.4.9 Preprocessing step: reference column (a) and the initial torso bounding box (b)

Next, this image is converted to grayscale and its edges are detected using the Canny operator. The individual contours in the resulting edge image are detected and evaluated. Contours with significant pixel height are redrawn in a contour image while vertically short contours are disregarded. The result is a contour image which contains possible contours corresponding to the edges of the torso.

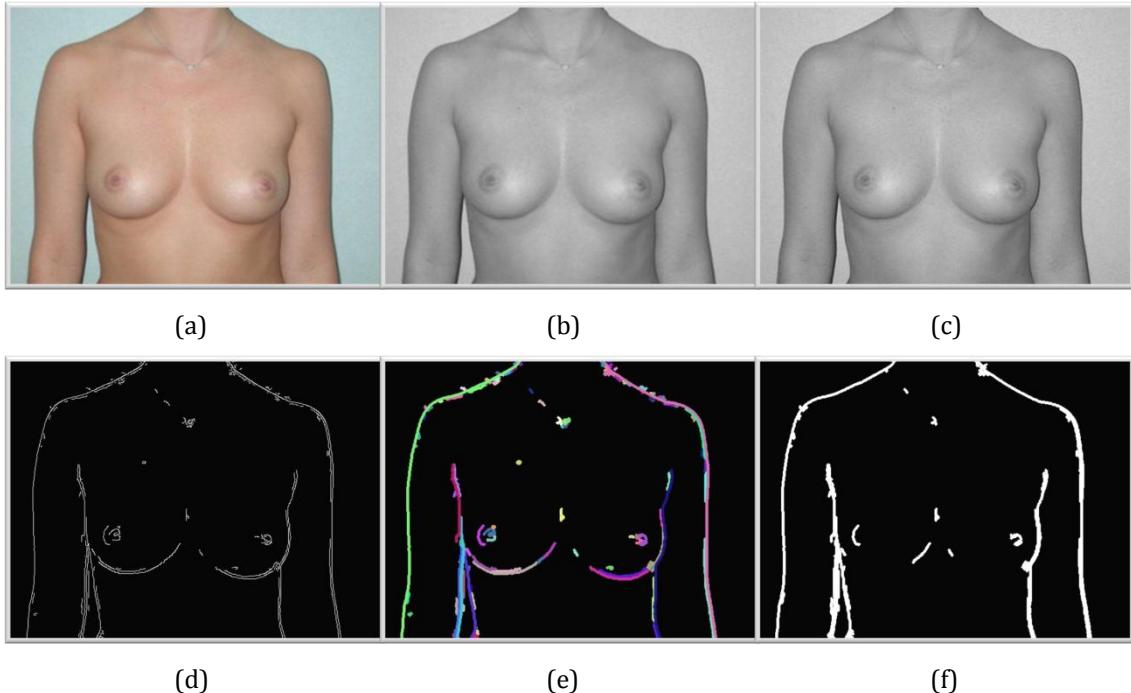


Figure 4.4.10 Edge detection and filtering step: original colored image (a), grayscale image (b), filtered image (c), Canny edge detection output (d), contour detection output (e), contour filtering output (f)

The final step is to locate these left and right torso edges. This is achieved using two sliding summation windows. These windows will take the sum of pixel values within it. The left and right summation windows are moved from the center towards the left and the right respectively. As shown in Figure 4.4.11, the summation windows will be able to find the torso edges by comparing the corresponding sum per window position. Summation of pixel values is done using the contour image's integral image. This is done to facilitate fast and efficient summations. A final bounding box is determined using the results of the sliding summation window method.

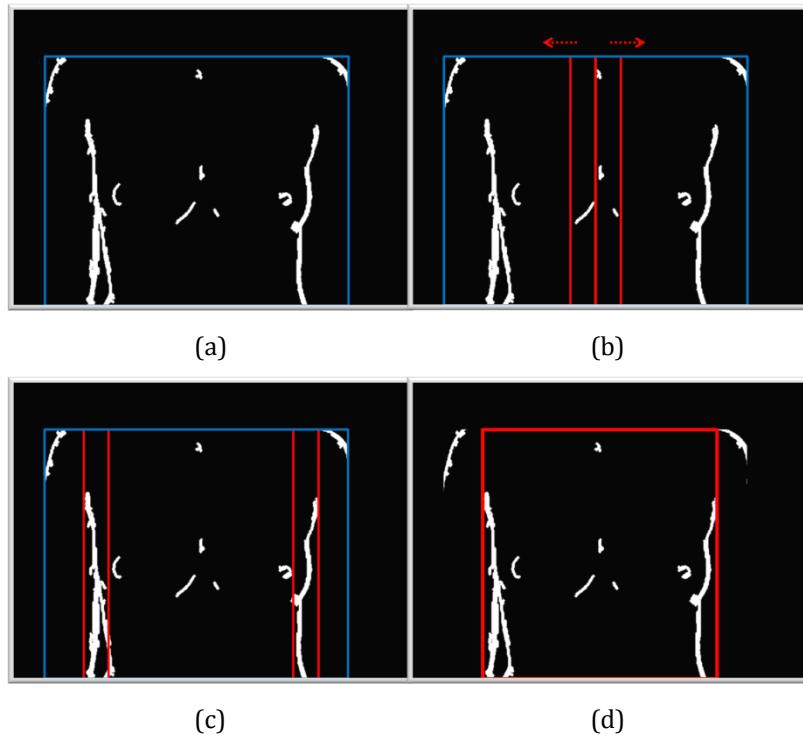


Figure 4.4.11 Search for torso edges: edges within the initial torso bounding box (a), sliding summation windows (b), torso edges located (c), final torso bounding box (d)

Shown in Figure 4.4.12 is a simplification of the code used for Stage 2 of the Breast Area Identification Algorithm. The code for initial cropping of the torso region is virtually identical to the code used to locate the collar bone. The grayscale image is filtered using the OpenCV function *bilateralFilter( )*. Edges are detected using the *Canny( )* function and individual contours are identified using *cvFindContours( )*. Vertically long contours are extracted by comparing the width and height of each contour. These are drawn on a contour map, *temp*. Finally, two summation windows are slid over this from the center outwards to determine the left and right torso boundaries. The final torso bounding box is defined using these two boundaries.



```
void Stage2()
{
    //Calculate summation of each column and find threshold
    for(int col=0; col<640; col++)
        for(int row=torsoROI.y; row<torsoROI.height; row++)
            if(skinmask.at<uchar>(row,col) > 0)
                colsum.at<float>(col)++;
            if(colsum.at<float>(col) > sumthres)
                sumthres=colsum.at<float>(col);

    //Determine left and right boundary
    for(int col=0; col<320; col++)
        if(colsum.at<float>(col) > sumthres*0.5)
            torsoROI.x = col;
    for(int col=640; col>320; col--)
        if(colsum.at<float>(col) > sumthres*0.5)
            torsoROI.width = torsoROI.width-(torsoROI.x+torsoROI.width-col);

    //Detect Edges and Contours
    bilateralFilter(frame_gray,temp);
    Canny(temp,temp);
    cvFindContours(temp, contours2);

    //Filter Contours
    for(int i; i<contours2.size; i++)
        if(contours2[i].height > contours2[i].width)
            cvDrawContours(temp, contours2[i]);

    //Calculate Integral Image
    integral(temp, intImage);

    //Sliding Summation Windows
    for(int i=mid; i>=0; i--)
    {
        sum = summation(Swindow);
        if(sum > sumleft)
        {
            sumleft = sum;
            leftBoundary = Swindow.x;
        }
    }
    for(int i=640; i>=mid; i++)
    {
        sum = summation(Swindow);
        if(sum > sumright)
        {
            sumright = sum;
            rightBoundary = Swindow.x + Swindow.width;
        }
    }

    //Define left and right torso edges
    torsoROI.x = leftBoundary;
    torsoROI.width = rightBoundary - leftBoundary;
}
```

Figure 4.4.12 Simplified code for Stage 2 of the Breast Area Identification Algorithm



#### 4.4.5 Stage 3: Locating the Inframammary Fold

Having identified the torso region, the left and right breast areas can now be properly identified. The first step to this is to split the torso region into two halves. Within these two half torso regions, the left and right breast areas will be identified respectively. Stage 3 of the Breast Area Identification Algorithm is summarized in Figure 4.4.13.

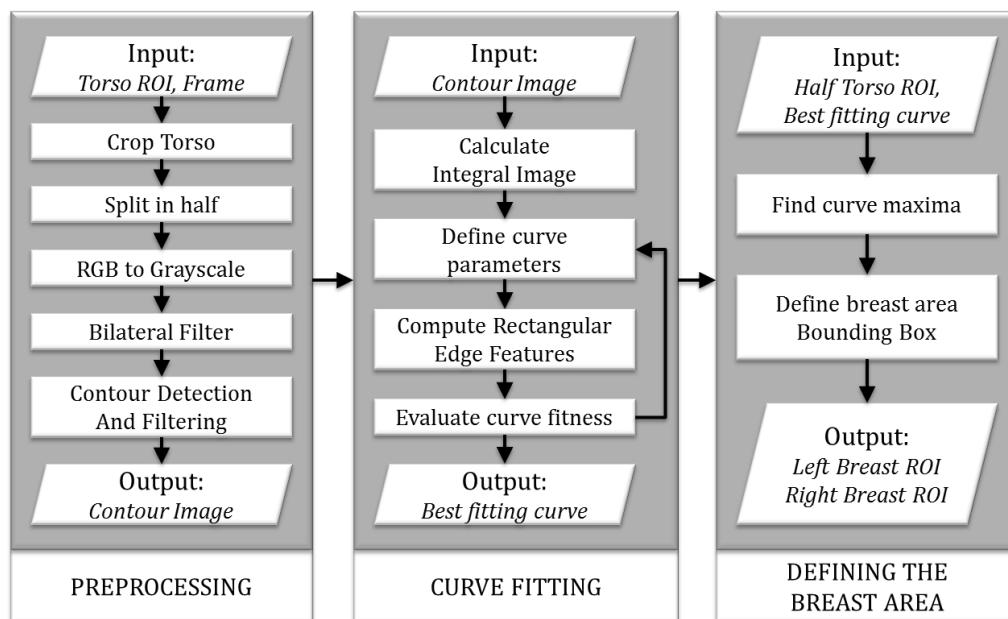
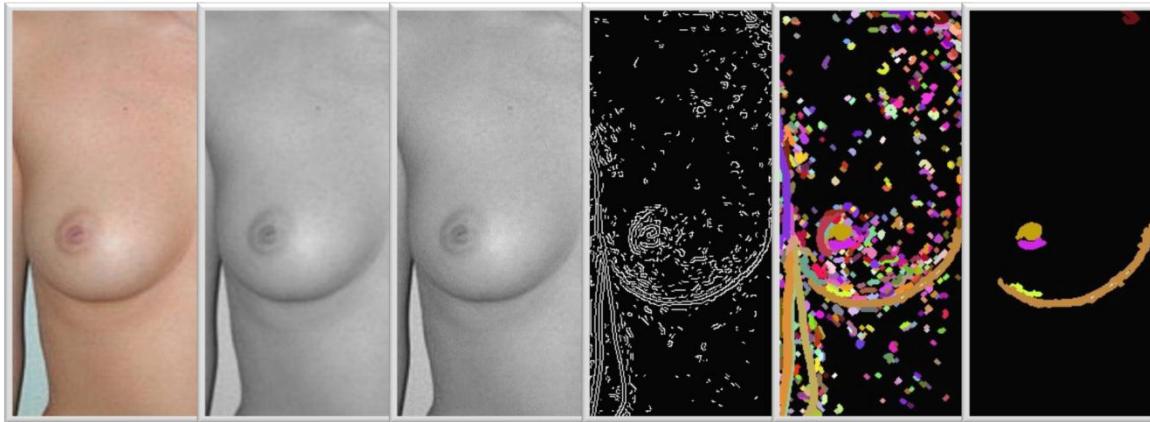


Figure 4.4.13 Block diagram of Stage 3 of the B.A.I. algorithm

Stage 3 will be highly similar with Stage 2. The main differences are in the orientation of the features of interest and the search method used. For the preprocessing, the cropped image will undergo bilateral filtering to strengthen edge features. Canny edge detection and contour detection will be applied after. This time, horizontally long contours are extracted and redrawn in a contour image while horizontally short contours are filtered out.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.4.14 Preprocessing step: cropped half of the torso region (a), grayscale image (b), filtered image (c), Canny edges (d), detected contours (e), filtered contours (f)

To search for the inframammary fold, a curve-fitting method is used. The inframammary fold is modeled as a quadratic curve defined by the function below.

$$Y = AX^2 + BX + C \quad \text{Equation 4.4.2}$$

X and Y correspond to the horizontal and vertical pixel coordinates respectively with respect to the image origin. The function is characterized by three constants A, B, and C to alter the curve's degree of curvature, skewness, and vertical offset respectively. The goal in this step is to find the optimal set of constants which would produce a quadratic curve which fits best with the inframammary fold. A fitness evaluation method is developed to determine the fitness of the quadratic curve.

As shown in Figure 4.4.15(c), the quadratic curve is divided horizontally into 30 sections. For each section, a two-rectangle feature similar to Haar-like features is calculated by getting the difference between the summations of pixel values in the white and gray rectangles respectively. The overall fitness of the quadratic curve is equal to the summation of all 30 features.



As shown in Figure 4.4.15(c), the quadratic curve is divided horizontally into 30 sections. For each section, a two-rectangle feature similar to Haar-like features is calculated by getting the difference between the summations of pixel values in the white and gray rectangles respectively. The overall fitness of the quadratic curve is equal to the summation of all 30 features.

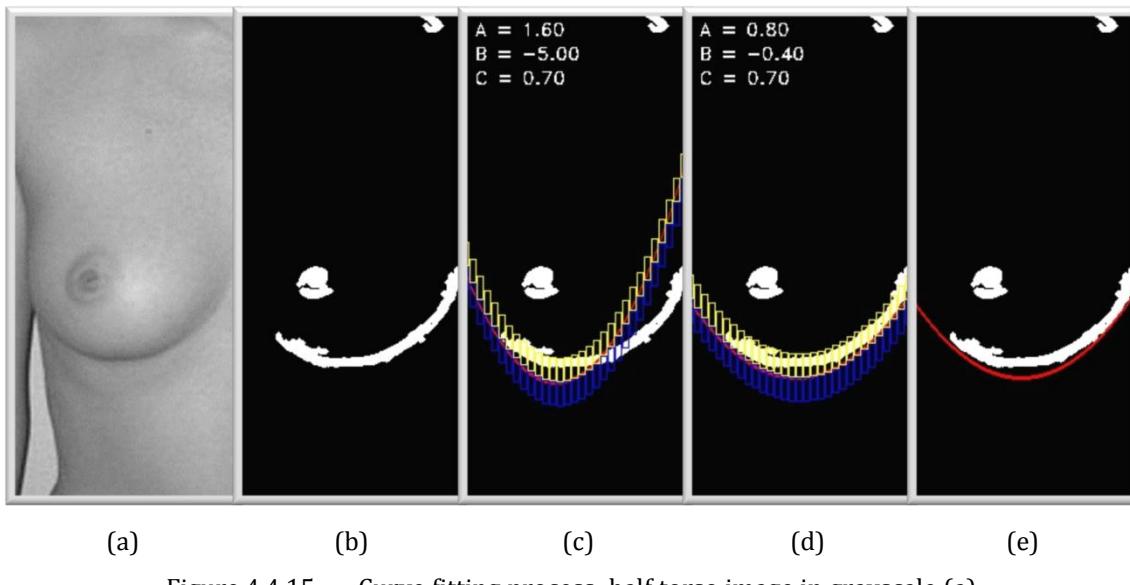
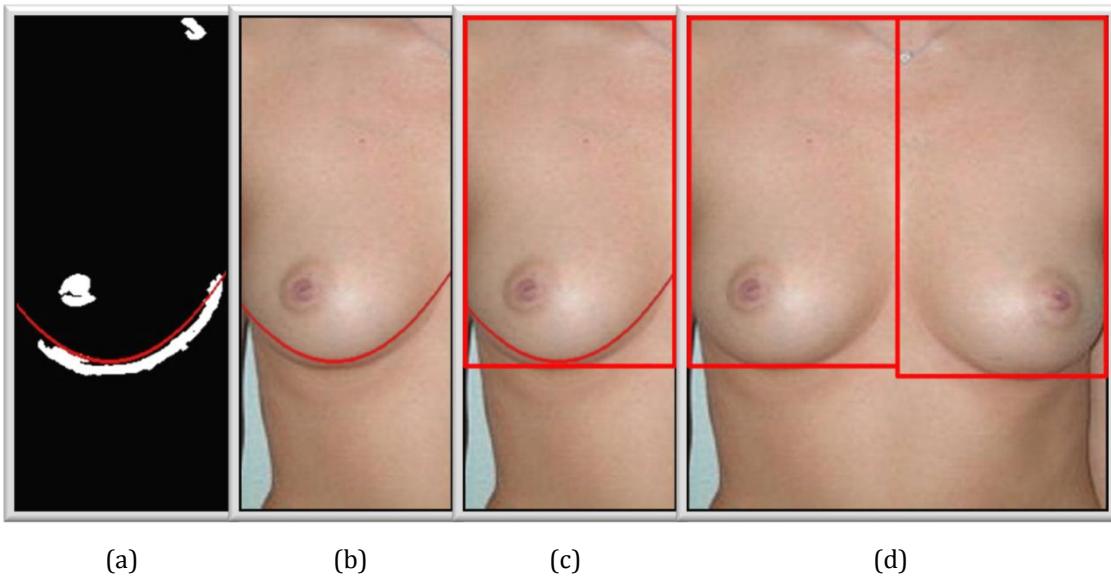


Figure 4.4.15 Curve fitting process: half torso image in grayscale (a), contours detected and filtered (b), curve fitness evaluation (c), best-fitting parameters (d), and best-fitting curve (e)

As the breast area is defined using a simple rectangular bounding box, the most important information from this best-fit curve is its local maxima (lowest visual point in the curve). After determining the local maxima of the curve, the breast area bounding box is now properly defined.



(a)

(b)

(c)

(d)

Figure 4.4.16 Defining the breast area: results of the previous step  
(a), best-fitting curve superimposed on the original image (b), breast area bounding box (c), left and right  
breast areas (d)



```
void Stage3()
{
    //Preprocessing
    cvtColor(halfTorso,halfTorso,CV_BGR2GRAY);
    bilateralFilter(halfTorso,halfTorso);

    //Edge/Contour Detection and Filtering
    Canny(halfTorso,halfTorso);
    cvFindContours(halfTorso, contours);
    for(int i; i<contours.size; i++)
        if(contours[i].height > contours[i].width)
            cvDrawContours(halfTorso, contours[i]);

    //Calculate Integral Image
    integral(halfTorso, intImage);

    //Curve Fitting
    for(C=0.7; C<=0.8; C=C+0.02)
        for(B=-5; B<=5; B=B+0.2)
            for(A=0.4; A<=2; A=A+0.4)
                curveFitness(intImage,breast,fitness,A,B,C);

                if(fitness > maxFit)
                    maxFit = fitness;
                    AA = A;
                    BB = B;
                    CC = C;

    //Defining the Breast Area
    for(int col=0; col<breast.width; col=col++)
        y = (float)(AA*(col-breast.width/2)*(col-breast.width/2)/(-100) + BB*(col-
breast.width/2)/10 + CC*breast.height) + 20;

        if(y>yMax)
            yMax = y;
}
```

Figure 4.4.17 Simplified code for Stage 3 of the Breast Area Identification Algorithm

The preprocessing step of Stage 3 uses the previously used OpenCV functions *cvtColor( )*, *bilateralFilter( )*, *Canny( )*, and *cvFindContours( )*. For the curve-fitting step, a linear search for the best set of parameters *A*, *B*, and *C* is performed. For each set of parameters, a fitness function *curveFitness( )* is evaluated. This function was manually formulated as described previously in this section. The resulting value is noted for each iteration and if exceeds the current best fitness value, the optimal parameters *AA*, *BB*, and *CC* are changed. At the end of this linear search, *AA*, *BB*, and



$CC$  contains the optimal combination of  $A$ ,  $B$ , and  $C$  which represents the best-fitting curve. Finally, the curve maximum is determined by evaluating the curve equation.

#### 4.4.6 Stage 4: Locating the Outer Breast Edges and the Sternum

At the end of Stage 3, a preliminary breast area has already been defined. Stage 4 serves as a refinement stage to more accurately define the left and right breast areas. The outer boundary of the breast area was determined as a result of defining the torso region in Stage 2. However, the outer boundary can be better defined using the contours of the breasts themselves.

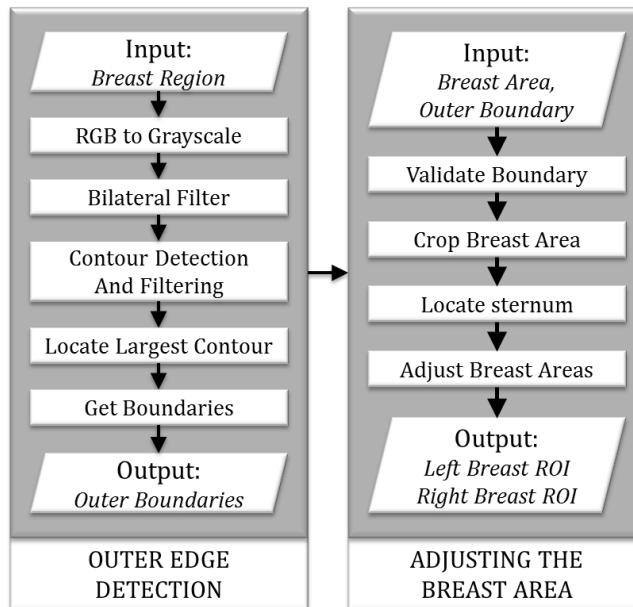


Figure 4.4.18 Block diagram of Stage 4 of the Breast Area Identification Algorithm

Similar to Stages 2 and 3, the preprocessing for Stage 4 is comprised of bilateral filtering, Canny edge detection, contour detection, and contour filtering. The task here is to detect the vertically longest contour which corresponds to the outer edge



of the breast. The outermost boundary of the bounding box of this contour is selected as the outer boundary of the breast area.

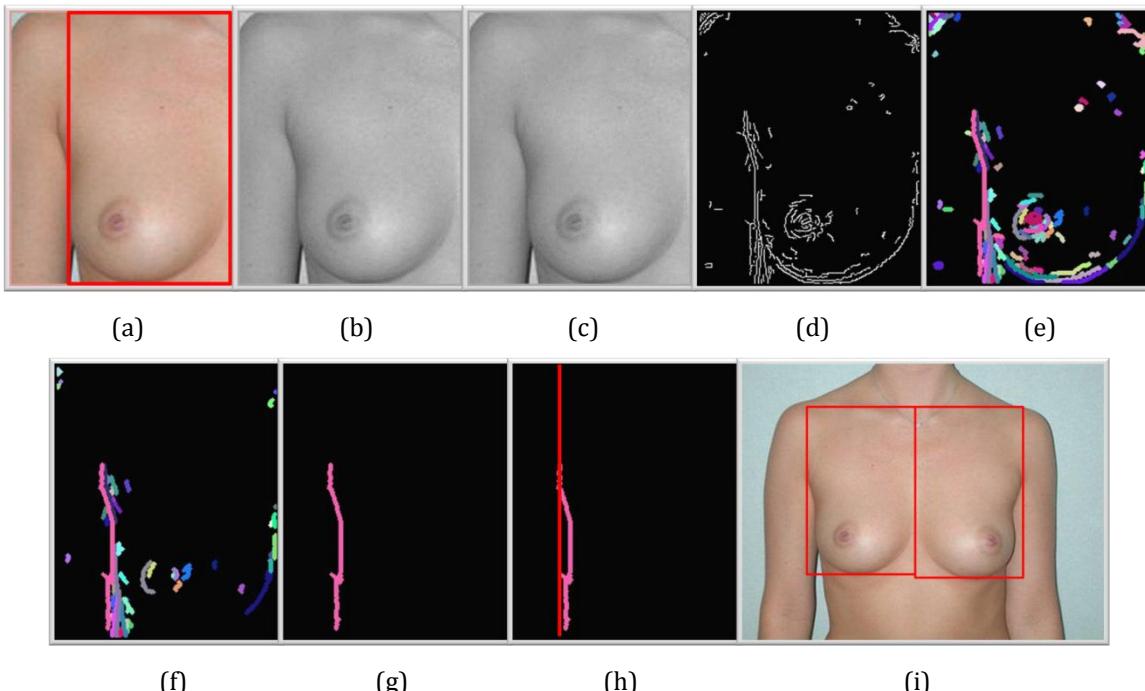


Figure 4.4.19 Stage 4 process: extended breast region (a), grayscale image (b), filtered image (c), Canny edges (d), detected contours (e), filtered contours (f), longest contour (g), outer boundary (h), adjusted left and right breast area bounding boxes (i)

Having adjusted the outer boundaries of the left and right breast areas, the location of the sternum or the inner boundaries of the breasts has to be adjusted as well. It is assumed that the left and right breast areas have equal width therefore the sternum is assumed to be at the center of the left and right breast areas.



```
void Stage4()
{
    //Preprocessing
    cvtColor(leftBreast,leftBreast,CV_BGR2GRAY);
    bilateralFilter(leftBreast,leftBreast);

    //Edge/Contour Detection and Filtering
    Canny(leftBreast,leftBreast);
    cvFindContours(leftBreast, contours);

    //Locate largest contour and Get boundaries
    for(int i; i<contours.size; i++)
        if(contours[i].height > 1.5*contours[i].width)
            if(contours[i].height > maxheight)
            {
                maxheight = contours[i].height;
                if(side == "left")
                    p1.x = breast.x + testROI.x;
                else if(side == "right")
                    p2.x = breast.x + testROI.x + testROI.width;
            }

    //Repeat everything for the right breast

    //Locate sternum and Adjust breast areas
    leftBreast.width = (rightBreast.x+rightBreast.width - leftBreast.x)/2;
    rightBreast.x = leftBreast.x + leftBreast.width;
    rightBreast.width = leftBreast.width;

}
```

Figure 4.4.20 Simplified code for Stage 4 of the Breast Area Identification Algorithm

Again, familiar functions such as `cvtColor( )`, `bilateralFilter( )`, `Canny( )`, and `cvFindContours( )` were used for the preprocessing steps. Then, the largest contour is identified by comparing the height of each contour to a threshold which is being updated for each iteration. The left or right boundary of this largest contour defines the outer boundary of either the left and right breast. All of these steps are performed for both the left and right breasts. Finally, the sternum is located by adjusting the bounding boxes for the left and right breasts.



## 4.5 Breast Area Division

### 4.5.1 Rationale

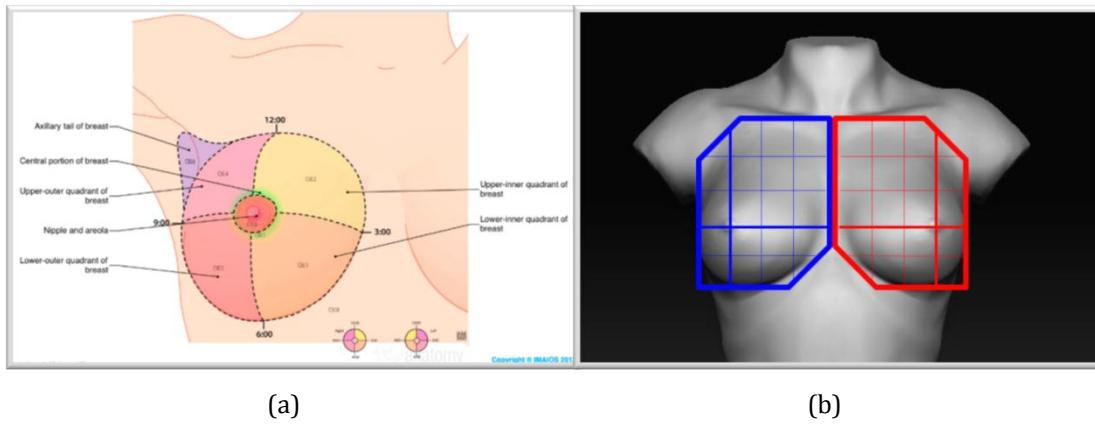


Figure 4.5.1. Diagram of the breast quadrant system showing the five major regions: nipple-areola complex, upper outer quadrant, upper inner quadrant, lower outer quadrant, lower inner quadrant, and axillary tail [66] (a) and the proposed breast area division into smaller blocks (b)

A common way of how doctors examine the breast is by using a quadrant system. In this manner, each breast is divided into four quadrants, namely, the upper outer, upper inner, lower outer, and lower inner quadrants; the quadrant system is centered on the nipple. This quadrant system adheres to the 10<sup>th</sup> revision of the International Classification of Diseases (ICD-10) [67]. As per the recommendation provided by the consulting medical doctor in [8], a quadrant system will be a more suitable way of dividing the breast area into smaller regions so that information regarding the location of the detected mass or abnormality can be easily understood and transmitted from patient to doctor and from doctor to doctor.

Though the quadrant system is ideal for communicating findings to doctors, this method of breast area division is not exactly ideal in the context of maximizing the



area thoroughness of BSE. The reason for this is because the four quadrants have very different areas with the two upper quadrants significantly larger than the two lower quadrants. Asking the user to palpate each quadrant is insufficient. To conform to the recommended BSE method of using coin-sized circular motions per region of palpation, each quadrant will be further subdivided into smaller blocks. The number of blocks per quadrant depends on the size of the quadrant.

### 4.5.2 Locating the Nipple

Two prerequisites to dividing the breast area into 4 quadrants are the breast area boundaries and the coordinates of the nipple centroid. Having defined previously the breast area, the remaining task is to locate the nipple.

To search for the nipple, a search window of variable size and location is used. The size and coordinates of this search window is varied and for each variation, a fitness function is evaluated. The fitness of the search window is derived from the center-surround Haar-like feature as shown in Figure 4.5.2.

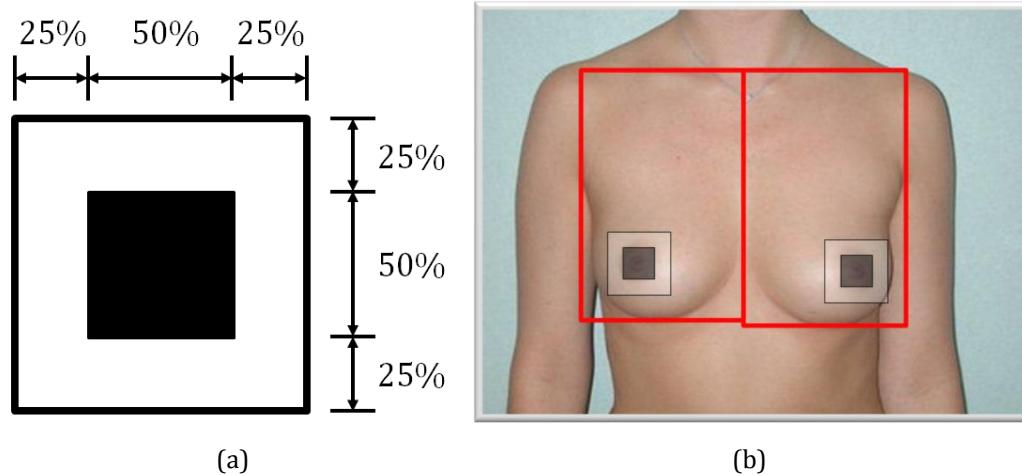


Figure 4.5.2    Center-surround Haar-like feature (a) and sample nipple tracking results (square) within the breast area (red box) (b)



The relative dimensions of the two-rectangle search window are shown in Figure 4.5.2(a). Its parameters will be varied in order to locate the nipple such as shown in Figure 4.5.2(b). To do this, a fitness function to be evaluated for each set of parameters. The goal here is to optimize the fitness function and find out the best set of parameters. The fitness function is loosely based on the calculation of a center-surround Haar-like feature where: HF is the numerical value of the feature, LR corresponds to the light region, and DR corresponds to the dark region.

$$HF = \sum pixel_{LR} - \sum pixel_{DR} \quad \text{Equation 4.5.1}$$

However, since the proposed fitness function is to evaluate features of different sizes, calculations must be uniform for all possible sizes to allow uniform comparison of fitness values. Instead of raw summation of pixels, the average of pixels per region is used to calculate the numerical value of the feature. Additionally, these values are normalized by dividing them with the maximum possible pixel value of 255. The resulting equation for calculating the numerical value of the feature is shown in the equation below where: HF is the normalized numerical value of the feature,  $LR_{ave}$  is the average of pixel values in the light region, and  $DR_{ave}$  is the average of pixel values in the dark region.

$$HF = (LR_{ave} - DR_{ave})/255 \quad \text{Equation 4.5.2}$$

Additionally, a size bias is included in the fitness function in order to avoid results wherein the best feature is either too big or too small. Through observation, it was determined that the average nipple size is around 30% of the breast area width. Since the size of the search window/feature is expressed as a percentage of the width of the breast area in decimal form, the size bias takes on the form shown in the equation below. SB is the size bias and  $W_{size}$  is the size of the search window.

$$SB = |0.3 - W_{size}| \quad \text{Equation 4.5.3}$$



Last, an offset value is included in the fitness function in order to prevent negative values. This is necessary because negative fitness values are not allowed in the optimization algorithm to be used which will be presented in the next section. The final form of the fitness function is shown below where C is equal to the offset value.

$$Fitness = (LR_{ave} - DR_{ave})/255 - |0.3 - Wsize| + C \quad \text{Equation 4.5.4}$$

### 4.5.3 Genetic Algorithm

In this study, genetic algorithm is used as the search heuristic for finding the optimal set of parameters for the nipple search window. This step is summarized in Figure 4.5.3 (a).

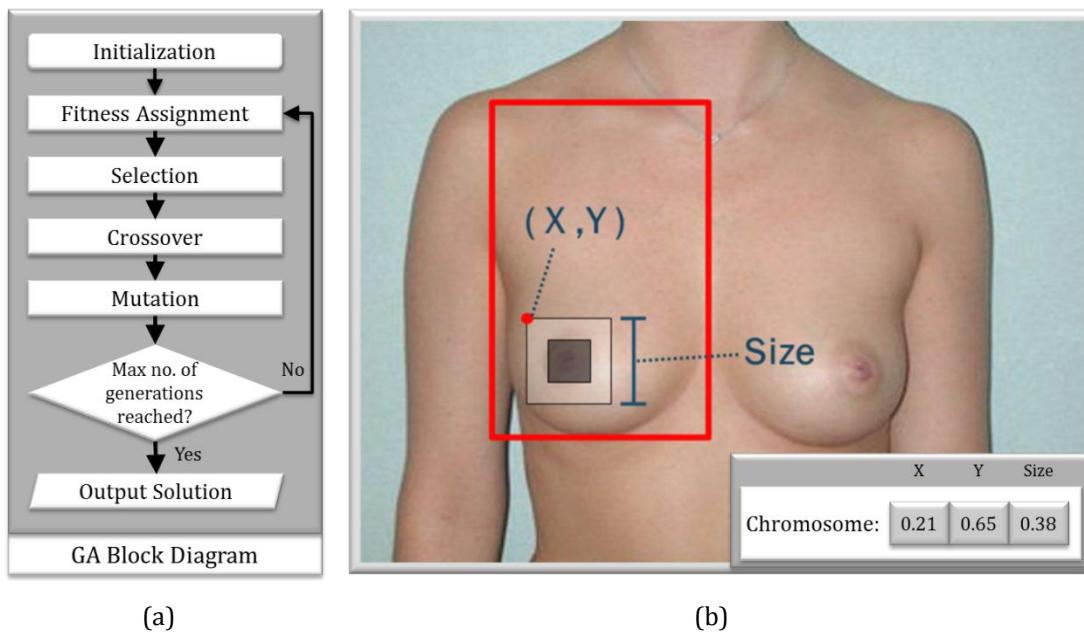


Figure 4.5.3

Block diagram of Genetic Algorithm (a) and structure of a chromosome (b)

The method used here follows the typical structure of genetic algorithm which is comprised of a number of phases namely: initialization, fitness assignment,



selection, crossover, and mutation. For the initialization phase, a population of 20 chromosomes is randomly generated. Each chromosome is composed of 3 genes: an X gene, a Y gene, and a Size gene. All three genes are expressed in terms of percentage of the breast area width in decimal form, i.e. 0.2, 0.51, 0.75, etc. For the fitness assignment phase, all 20 chromosomes' fitness values are evaluated using the method and fitness function explained in Section 4.5.2. For the selection phase, the chromosomes are arranged based on their fitness values in descending order. Each chromosome is randomly selected to form the elite pool. Their probability of selection depends on their fitness value relative to the fitness value of the best chromosome. Chromosomes which did not survive into the elite pool are replaced by the best chromosome.

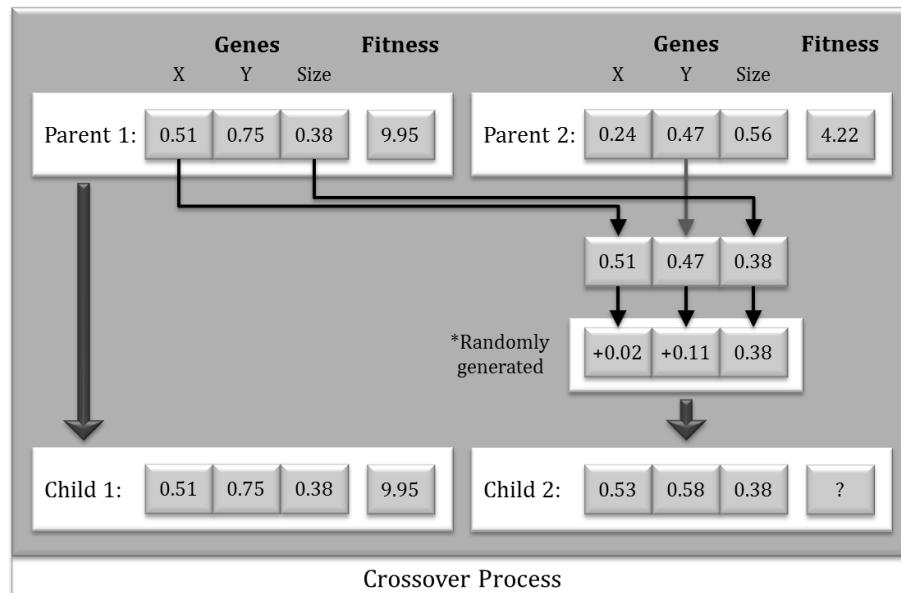


Figure 4.5.4 Block diagram of the crossover phase used in the genetic algorithm

For the crossover phase, chromosomes in the elite pool are randomly selected to undergo crossover based on a predefined crossover probability. As shown in Figure



4.5.3 (b), the first among the two offspring is a clone of the parent chromosome with the better fitness value. The second child on the other hand takes its X gene and Size gene from the better parent and its Y gene from the other parent. Additionally, the three genes of the second child are slightly mutated by adding randomly generated floating point numbers. The two offspring replace their parents in the elite pool.

For the mutation phase, each of the 6 digits after the decimal point of each gene of each chromosome is randomly mutated based on a predefined mutation probability. Each digit, if selected, is to be mutated by adding or subtracting a randomly generated digit. Since this process involves a high probability of introducing invalid chromosomes, mutated chromosomes are validated first and if they are considered invalid, the mutations are disregarded and the chromosome is left unchanged.

After the mutation phase, the chromosomes in the elite pool are carried over to the next generation. This next generation will repeat the cycle starting from fitness assignment up to mutation, to be repeated over and over again until at least one of the following conditions is satisfied.

1. The three chromosomes with the highest fitness values are identical and their fitness value is equal to or greater than a predefined threshold
2. The predefined maximum number of generations has been reached

When the cycle is terminated, the genes of the best chromosome are selected as the X and Y coordinates and window size of the nipple search window.



#### 4.5.4 Quadrants and Blocks

The breast area is divided primarily into 4 quadrants as shown in Figure 4.5.5. These are the upper inner quadrant (blue), the upper outer quadrant (green), the lower outer quadrant (red), and the lower inner quadrant (yellow).

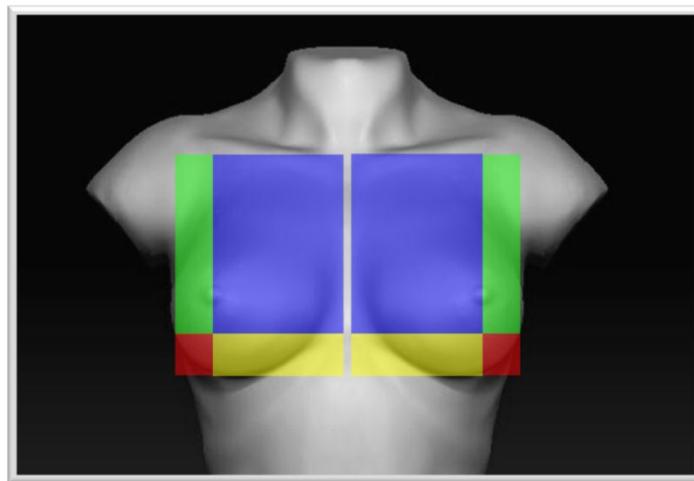


Figure 4.5.5      Sample breast areas divided primarily into four quadrants each

Each quadrant is further divided into smaller blocks arranged in rows and columns. The number of rows per quadrant is dependent on the ratio of the quadrant height to the breast height. On the other hand, the number of columns per quadrant is dependent on the ratio of the quadrant width to the breast width. Shown in Figure 4.5.6 is an example of a possible breast area division.

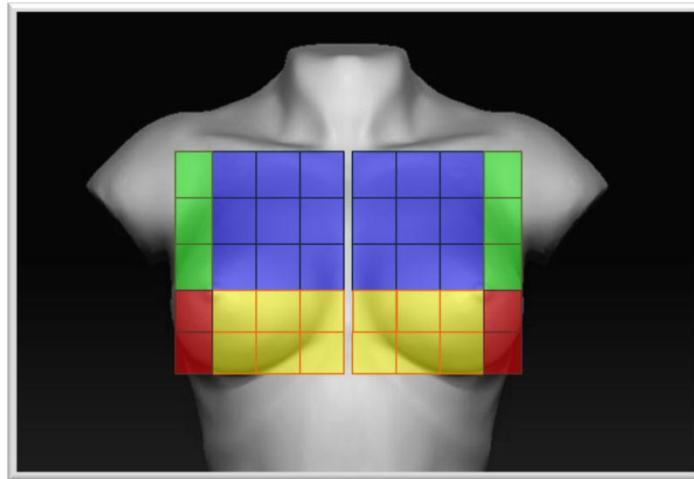


Figure 4.5.6 Sample breast areas divided into quadrants and smaller blocks

### 4.5.5 Implementation

The genetic algorithm used in this part is implemented manually using only the basic structures of the C++ language. The only special OpenCV-specific function used is the *integral()* function used to calculate the integral image for faster evaluation of the fitness function. The actual code is a direct translation of the processes explained in Section 3.7. Specific values of the parameters are as follows. The population size is set at 30, maximum number of generations at 20, crossover probability at 0.5 and mutation probability at 0.3.



## 4.6 Hand Tracking

### 4.6.1 Rationale

After successfully delineating and characterizing the breast area, the next important step is to monitor the palpation to be conducted by the patient/user. An important prerequisite to this is to be able to detect and track the hand of the user. There have been a number of proposed algorithms to solve this task as presented in Chapter 2. In this study however, an entirely different approach is used to solve the color similarity problem.

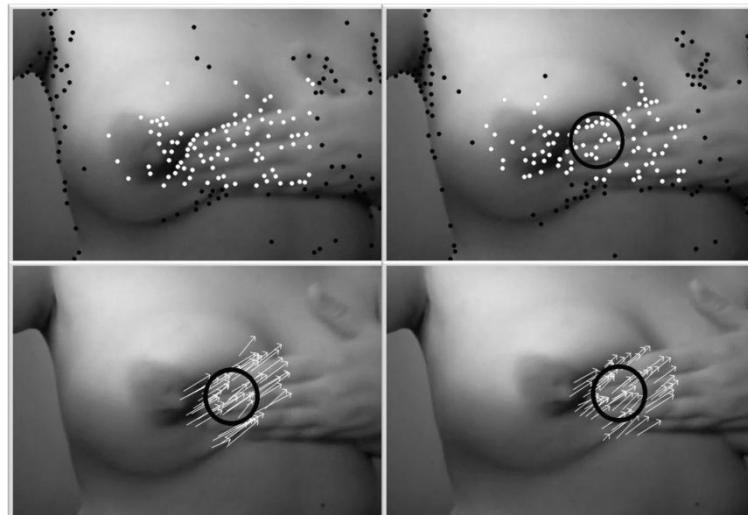


Figure 4.6.1     Sample results of the proposed hand tracking methodology

Instead of using color features or hand segmentation, corner features will be used to detect the hand and locate the finger pads. Subsequently, optical flow is used to independently track these corner features in order to track the hand during palpation.



## 4.6.2 Initialization

The first step to tracking the hand, specifically the finger pads, is to detect its presence. Shown in Figure 4.6.2 is the step-by-step process of the simple algorithm for detecting the presence of the hand up to the estimation of the finger pad location.

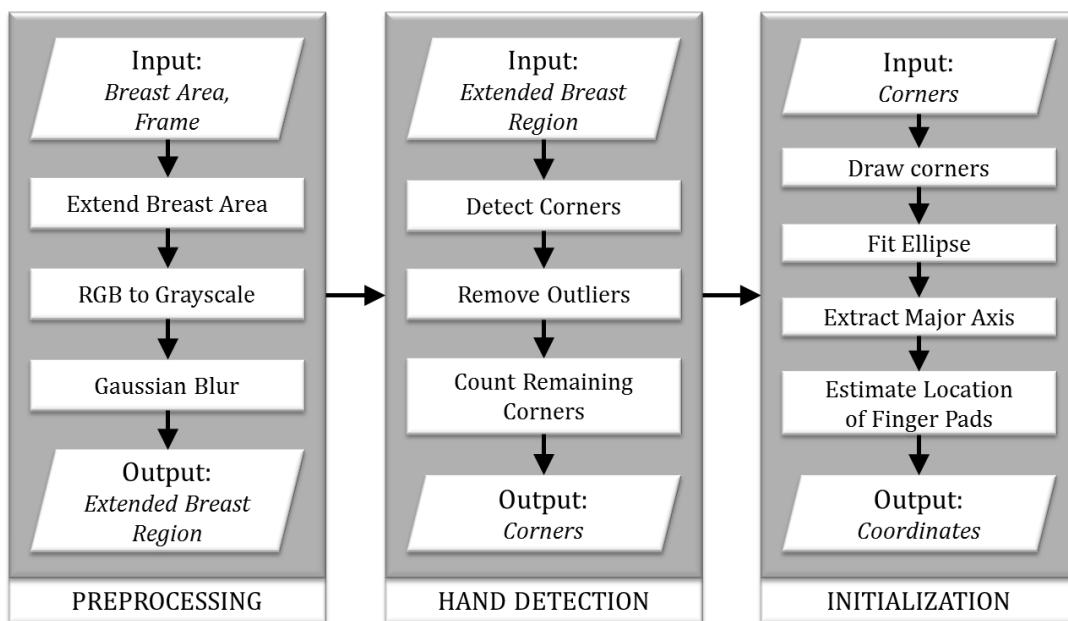


Figure 4.6.2 Block diagram of the Initialization Stage of the Hand Tracking Algorithm

For the preprocessing step, the boundaries of the breast area defined previously are extended temporarily. This extended breast region is extracted as the hand will be detected and tracked within its boundaries. This region is converted to grayscale and subjected to Gaussian blur to remove noise.

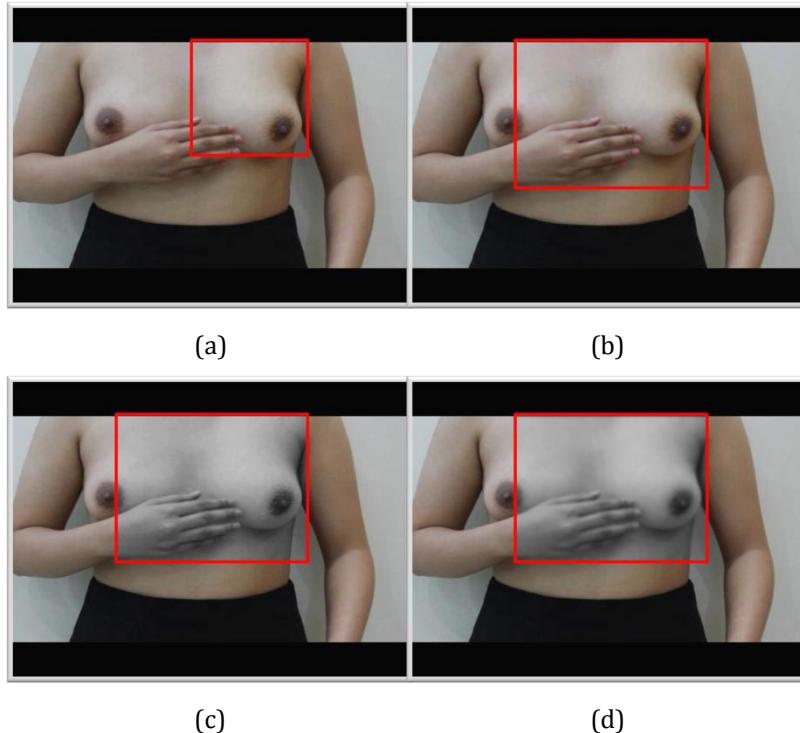
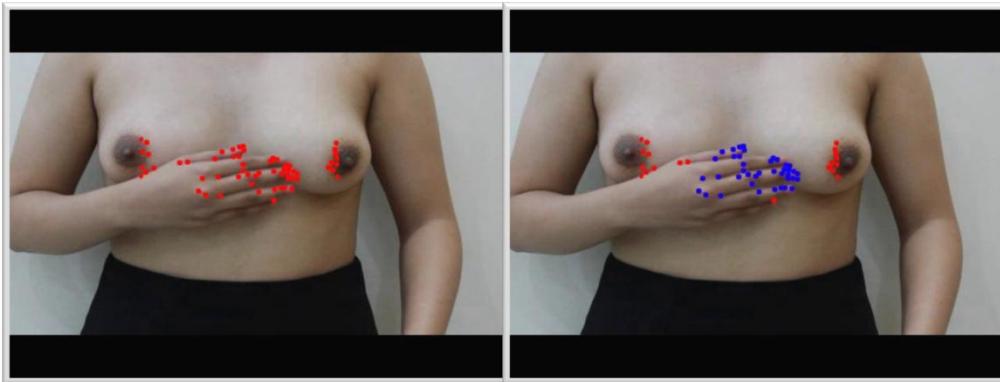


Figure 4.6.3 Preprocessing step: original breast region (a), extended breast region (b), ROI in grayscale (c), and ROI after applying Gaussian blur (d)

Shi-Tomasi corner detection is used to detect strong corner features within the extended breast region. Outlying corners are eliminated using a filtering method. First, the centroid and the standard deviations on both horizontal and vertical axes of the feature set are determined. If the horizontal or vertical distance of each feature from the centroid exceeds the horizontal or vertical standard deviation respectively, that corner feature is eliminated.



(a)

(b)

Figure 4.6.4 Corner detection results (a) and outlier elimination results (b) showing detected corners (red) and selected corners (blue)

The number of remaining features is compared to a predefined threshold to determine if the hand is present or not. If too few features remained, it implies that the features are mostly scattered which indicates that the hand is not present. In such case, the detection step must be repeated until the hand is detected. Otherwise, it implies that the hand is present and most features detected are located along the area of the hand.

The remaining features are drawn on a feature map and an ellipse that fits the feature set best is calculated based on the function developed in [68]. The location of the finger pads is estimated as lying along the major axis of the best fitting ellipse, with a distance of 25% of the length of the major axis from either one of the two endpoints. If the user is using her left hand, the right endpoint of the major axis is selected. Otherwise, the left endpoint is selected.

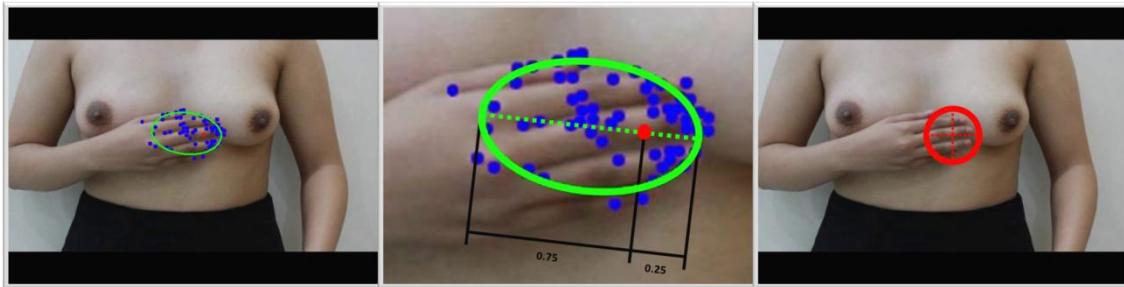


Figure 4.6.5      Sample initialization result showing detected corners (blue), best-fitting ellipse (green) and estimated finger pad location (red)

```
void initialization()
{
    //Preprocessing
    cvtColor(frame2(ROI), breastarea, CV_BGR2GRAY);
    breastarea = GaussianBlur(breastArea);

    //Detect corners
    goodFeaturesToTrack(next, corners, maxCorners);

    //Remove outliers
    AveStdDev(corners, xmean, ymean, xstd, ystd);
    for (int i=0; i<corners.size(); i++)
        if (abs(corners[i].x-xmean)>xstd || abs(corners[i].y-ymean)>ystd)
            corners.erase(corners.begin()+i);           i--;

    //Fit Ellipse
    RotatedRect hand;
    hand = fitEllipse(corners);

    //Extract ellipse properties
    Point2f vertices[4];
    hand.points(vertices);
    ellipse( Results, hand, Scalar(0,255,0), 2, 8 );

    //Estimate Location of Finger Pads
    handloc = Point((vertices[0].x+vertices[3].x)/2,(vertices[0].y+vertices[3].y)/2);
}
```

Figure 4.6.6      Simplified code for the initialization stage of the Hand Tracking Algorithm

For the preprocessing step, the cropped breast image is converted to grayscale and filtered using the OpenCV functions *cvtColor( )* and *GaussianBlur( )*. Corner features are detected using the Shi-Tomasi corner detection function of OpenCV, *goodFeaturesToTrack( )*. Outliers are removed by comparing the deviations in the X



and Y axis of each feature from the centroid to the corresponding standard deviations. The centroid and standard deviations are calculated first using *AveStdDev( )*. The best fitting ellipse to the remaining feature set is determined using the *fitEllipse( )* function of OpenCV. The exact coordinates of the finger pads are extracted using the exact procedure previously described in this section.

### 4.6.3 Tracking

After detecting the hand, the next step is to track its location in the subsequent frames. In this part, a unique sparse optical flow-based method is used. Sparse optical flow processes only a set of pixels from the whole image. In this case, the corners detected previously in the hand detection stage are used to calculate the optical flow and estimate the hand motion. The step by step process is shown below.

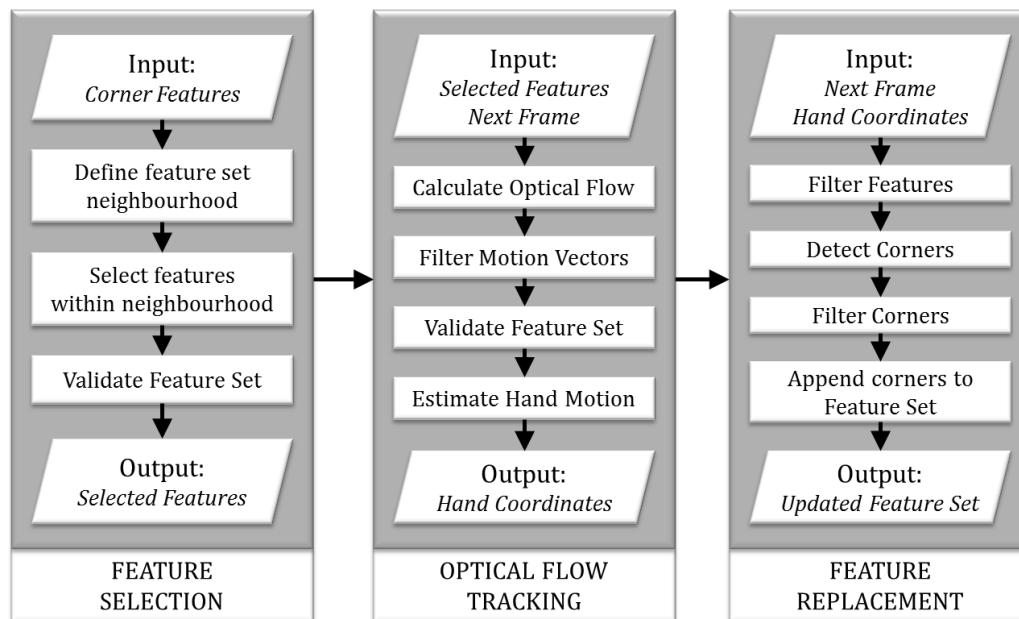
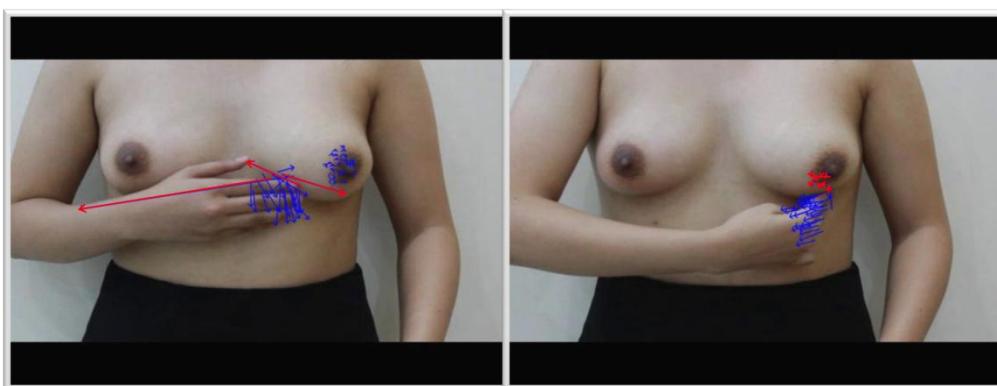


Figure 4.6.7 Block diagram of the tracking stage of the Hand Tracking Algorithm



Before calculating the optical flow using the feature set extracted from the hand detection stage, they are further screened first. This is done by defining a neighborhood of features-of-interest. This is defined using a circular section of the image whose radius is equal to 25% of the breast region. All features outside of this neighborhood are removed from the feature set. If very few features remained after this step, then the algorithm lost track of the hand and would need to undergo the initialization stage again.

The next step is to track the location of each feature within the next frame. This is achieved using Lucas-Kanade optical flow. The goal of this step is to estimate the motion of the hand using information from the motions of the individual corner features of the feature set. The next few steps after calculating the optical flow is to filter out the tracked features so that only reliable ones will remain from which the hand motion/displacement can be estimated from. First is to remove very large displacement vectors as shown in Figure 4.6.8 (a). These large motion vectors correspond to incorrectly tracked features. Next is to remove very small motion vectors as shown in Figure 4.6.8 (c). These small motion vectors most likely correspond to features which do not belong to the hand.



(a)

(b)

Figure 4.6.8      Example of large motion vectors {red arrows} (a) and very small motion vectors {red arrows} (b)



Ideally at this point, only the motion vectors which correctly characterize the hand displacement remain. The displacement of the hand between the previous and current frames is estimated by calculating the average displacement for all of the remaining motion vectors.

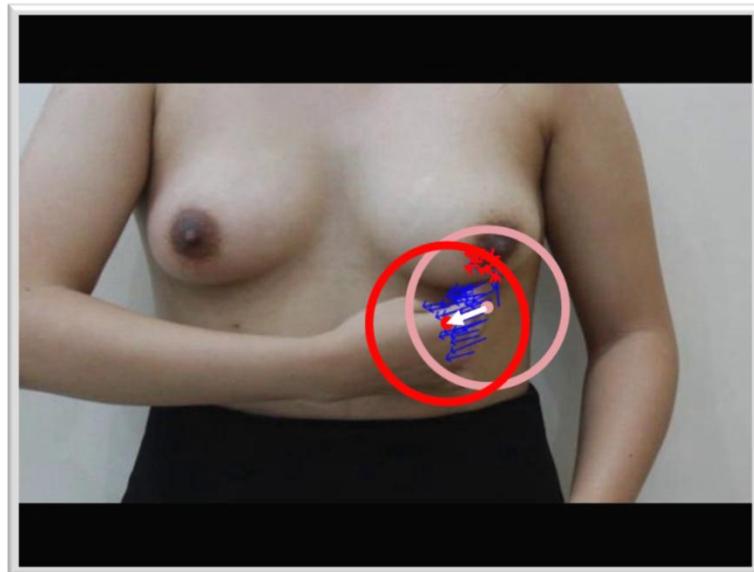


Figure 4.6.9 Hand motion estimation: motion vectors of the feature set (blue arrows) and the average motion vector (white arrow)



```
void opticalFlowTracking()
{
    //Eliminate corners not within neighborhood of previous solution
    for (int i=0; i<corners.size(); i++)
        if (sqrt(pow(prevloc.x-corners[i].x,2) + pow(prevloc.y-corners[i].y,2)) > sRadius)
            corners.erase(corners.begin()+i); i--;

    //Calculate Optical Flow
    calcOpticalFlowPyrLK(prev, next, corners, corners_b);

    //Filter Motion Vectors
    for (int i=0; i<corners.size(); i++)
    {
        float dev = (float)sqrt(pow((corners[i].x - corners_b[i].x),2) + pow((corners[i].y - corners_b[i].y),2));
        if (dev > maxDev || dev < minDev)
            corners.erase();
    }

    //Estimate Hand Motion
    AveStdDev(corners, xmean, ymean, xstd, ystd);
    Point temp1 = Point(xmean, ymean);
    AveStdDev(corners_b, xmean, ymean, xstd, ystd);
    Point temp2 = Point(xmean, ymean);

    handloc.x = prevloc.x+(temp2.x-temp1.x);
    handloc.y = prevloc.y+(temp2.y-temp1.y);
}
```

Figure 4.6.10 Simplified code for the optical flow tracking step of the Hand Tracking algorithm

The neighborhood of features-of-interest is defined in the implementation by a pre-defined value,  $sRadius$ . By comparing the distance of each feature from  $sRadius$ , features outside this neighborhood are eliminated. Next, these features are tracked using the optical flow function of OpenCV,  $calcOpticalFlowPyrLK()$ . Next, very large and very small motion vectors are eliminated by comparing the magnitude of each vector to two values:  $minDev$  and  $maxDev$ . Overall hand displacement is estimated as the average displacement of the remaining vectors using the  $AveStdDev()$  function. This displacement is added to the previous hand location,  $prevloc$ , to come up with the current hand location,  $handloc$ .



On the way to tracking the hand, the feature set undergoes several filtering steps. Thus, lots of features are being removed. Also, these features may tend to flock very close with each other and thus would not sufficiently represent the hand. Therefore, in order to keep the tracking continuous, the feature set must be updated and replenished accordingly. First, features which are very close to other features are removed. Then, additional features, detected using the same method using in the initialization stage, are automatically appended. This feature set is fed input to the next iteration of the tracking stage. The tracking stage is repeated over and over again until the algorithm loses track of the hand, in which case the initialization stage must be repeated.

```
void featureReplacement()
{
    /*Remove features too close to other features
    for (int i=0; i<features.size(); i++)
        featuremap.at<uchar>(features[i].y,features[i].x) = 255;

    for (int i=0; i<features.size(); i++)
    {
        circle(featuremask,Point(features[i].x,features[i].y),10,Scalar(255),-1,8,0);
        bitwise_and(featuremap,featuremask,maskedfeatures);
        Scalar s = sum(maskedfeatures)[0]/255;
        if(s.val[0] > 1)
            features.erase(features.begin()+i);      i--;
    }

    //Detect additional features
    goodFeaturesToTrack(next,corners,maxCorners);

    //Append additional features
    if(features.size() < maxCorners)
        features.push_back(corners[i]);
}
```

Figure 4.6.11 Simplified code for the Feature Replacement step of the Hand Tracking algorithm

Features which are too close to other features are removed through the following method. First, all of the remaining features are drawn on the matrix *featuremap*. Second, a circle, centered at the current feature-of-interest and having a radius of 10 pixels, is drawn on the matrix *featuremask*. The intersection of the two



matrices is computed using the function `bitwise_and()`. Technically, if the current feature-of-interest is nowhere near other features, the only intersection between the two matrices is the feature-of-interest itself. Otherwise, they may be more intersections. In such case, the feature-of-interest is eliminated. Additional corner features are detected using the `goodFeaturesToTrack()` function again. These are appended to the current feature set.

## 4.7 Palpation Detection

### 4.7.1 Rationale

Using the output of the hand tracking stage, lots of information can be extracted from it which can be used to detect palpations being done by the patient. In this algorithm, hand movements must satisfy four criteria in order for them to be identified as correct palpation. These are the following:

1. Shape and repetitiveness
2. Proper speed of palpation
3. Correct palpation location
4. Proper size of palpation

In the developed algorithm, compliance to these criteria can be determined using the X and Y coordinates of the tracked hand within subsequent frames. These are treated as two separate discrete-time signals.

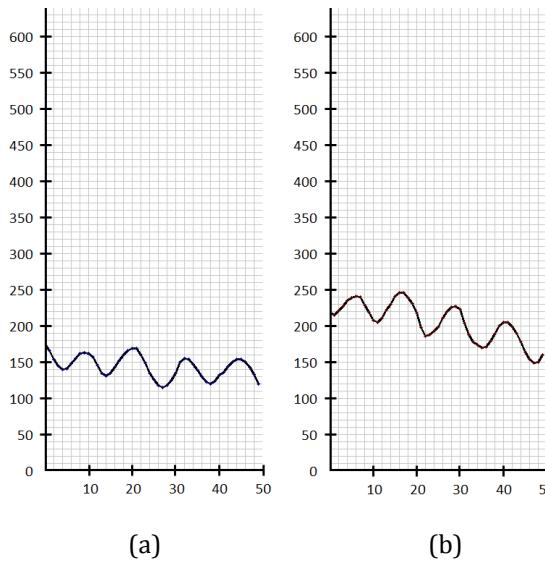


Figure 4.7.1     Sample X signal (a) and Y signal (b)

### 4.7.2 Signal Sampling

To detect palpations, the X and Y signals are continuously sampled using two discrete-sized buffers. For each frame, the sampled signal contained in the buffer is analyzed whether it satisfies the necessary criteria. Each buffer has a length of 50 samples. This is approximately equivalent to a 2-second long sample as the system has a frame rate of 25 fps. These buffers are continuously updated as long as the system maintains proper hand tracking. Once the system loses track of the hand, the buffers are cleared.

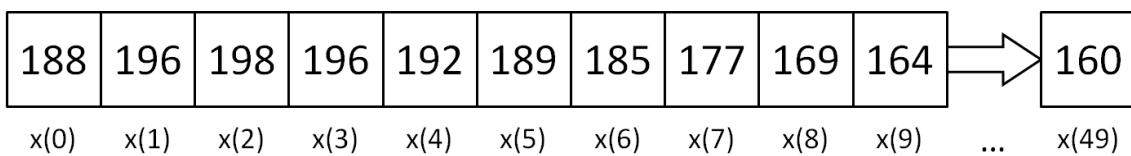


Figure 4.7.2     Sample X buffer with values for the waveform shown in Figure 4.7.1 (a)



### 4.7.3 Time Series Regression Analysis

The X and Y signals are also treated as time series data in which linear regression analysis is performed. The idea here is that the hand movements are perfectly repetitive, thus the sampled data points must be symmetrical with respect to the horizontal axis. Using linear regression analysis, ideally the slope of the linear model must be equal to zero due to the horizontal symmetry.

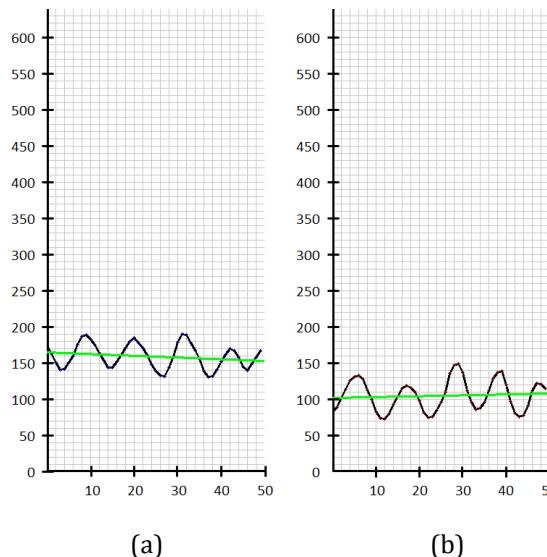


Figure 4.7.3 Sample X signal (a) and Y signal (b) during palpation with their respective linear regression models (green line)

The linear regression model is described by the equation of a line shown below where Z corresponds to the value of the signal at time t, A is the slope of the line, and B is the Z-intercept.

$$Z = At + B \quad \text{Equation 4.7.1}$$

The goal here is to find the equation of a line, specifically the constants A and B, which best fits the series of data points. To do this, the linear least squares approach



is used in which the following equation is used where t corresponds to the time index and Z is the signal value at the specified time index.

$$\begin{bmatrix} \sum_{i=1}^m t_i^2 & \sum_{i=1}^m t_i \\ \sum_{i=1}^m t_i & \sum_{i=1}^m 1 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m Z_i t_i \\ \sum_{i=1}^m Z_i \end{bmatrix}$$

Equation 4.7.2

For each frame, the signal buffers are being updated. Hence, for each frame, a linear regression model is calculated. Using the linear regression model for each frame, the sample data contained in the buffers can be tested whether they satisfy the 4 criteria for correct palpation.

```
void regression(Mat &buffer, float &A, float &B)
{
    float g1,g2,g3,g4,h1,h2;

    g1 = 40425;
    g2 = 1225;
    g3 = g2;
    g4 = 50;

    h1 = 0;
    h2 = 0;
    for(int count=0; count<50; count++)
    {
        h1 = h1 + count*buffer.at<float>(count);
        h2 = h2 + buffer.at<float>(count);
    }

    float r1, r2, r3, r4;

    r1 = (float)(h1/g1);
    r2 = (float)(h2/g3);
    r3 = (float)(g2/g1);
    r4 = (float)(g4/g3);

    B = (float)((r1-r2)/(r3-r4));
    A = (float)(r1 - B*r3);

    return;
}
```

Figure 4.7.4     Code snippet for the Linear Regession Analysis in the Palpation Detection Algorithm



#### 4.7.4 Criteria for Evaluation

For the first criteria, the hand must perform a number of circular motions repetitively within a specific area of the breast. Ideally, the X and Y signals must be perfect sinusoids. However, since maintaining perfectly circular motions at a perfectly constant speed is almost impossible, the X and Y signals only somewhat resemble sinusoids. In this algorithm, the main concern regarding criteria #1 is the oscillation in each signal. To satisfy criteria #1, the sampled signal within the buffer must be oscillating with respect to a fixed level such as that the slope of the linear regression model is within the allowable values of  $\pm 0.2$ . As shown in Figure 4.7.5, unlike random hand motion, palpation motion is characterized by a repetitive wave and its linear regression model has a slope near to zero.

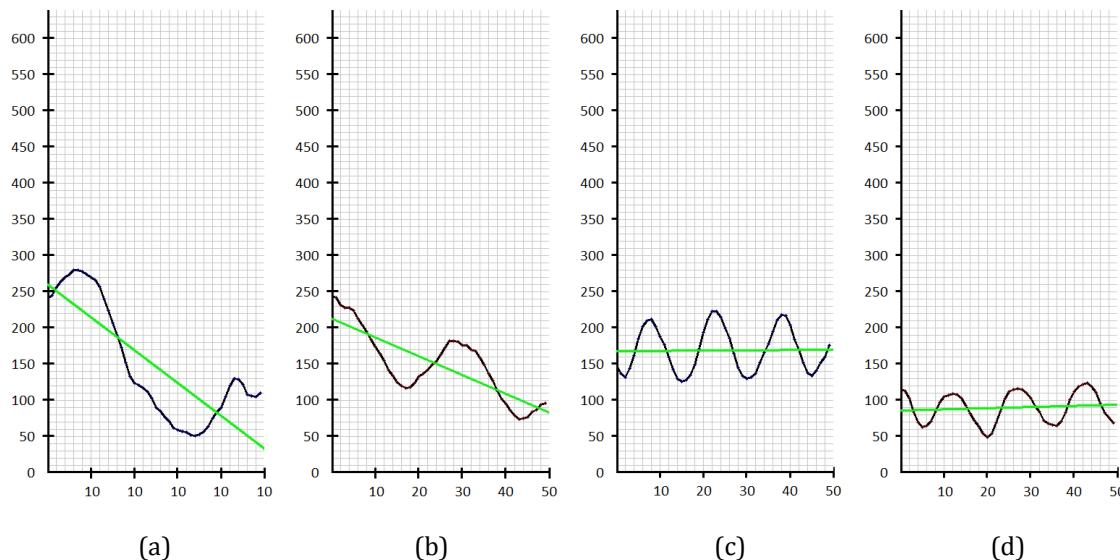


Figure 4.7.5

Sample X signal (a) and Y signal (b) for non-palpation motion and sample X signal (c) and Y signal (d) for palpation motion with their respective linear regression models (green line)



```
float slopeThresh = 0.3;  
  
regression(xBuffer,xA,xB);  
regression(yBuffer,yA,yB);  
  
if(xA < slopeThresh && xA > -slopeThresh)  
    palpCriteria[0][0] = true;  
  
if(yA < slopeThresh && yA > -slopeThresh)  
    palpCriteria[1][0] = true;
```

Figure 4.7.6    Code snippet for the evaluation of the first criterion for correct palpation

In the implementation of the system, the first criterion is checked as follows. The linear regression model for both X and Y signals inside *xBuffer* and *yBuffer* are computed using the *regression( )* function. The slopes *xA* and *xB* of the two linear models are compared to a pre-defined threshold *slopeThresh* to determine whether the currently buffered hand motion satisfies the first criterion. In the final implementation, *slopeThresh* is set to 0.3.

Second, the patient must perform the palpation at a proper speed. In this implementation, the recommended speed of palpation is around 1.5 to 2 cycles per second. Thus the patient must be able to perform at least 3 complete cycles of palpation within the 2-second margin recorded by the X and Y buffers. By counting the number of complete cycles recorded in the buffers, the algorithm can estimate the speed of palpation. If the patient is palpating slower than the recommended speed, palpation will not be detected. If the patient is palpating very fast, i.e. more than 4 complete cycles, the detected palpation is deemed invalid and the system will notify the user to slow down.

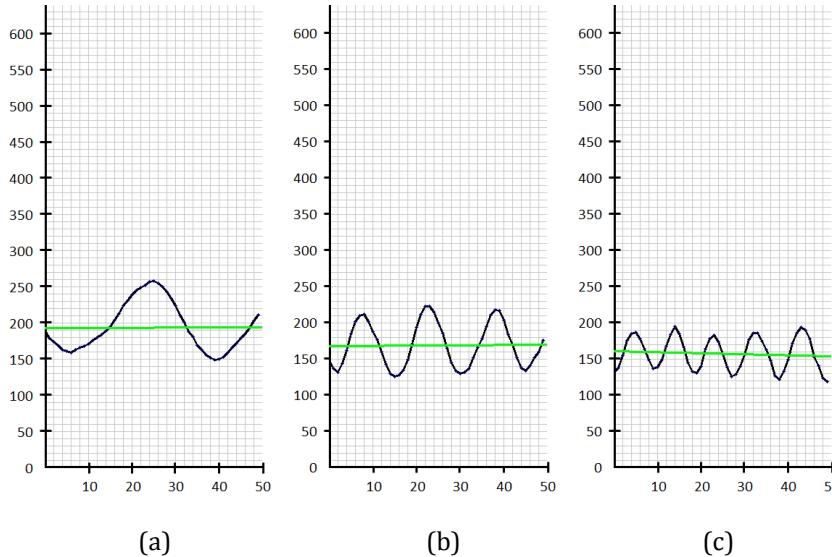


Figure 4.7.7 Sample X signals showing palpation at different speeds: too slow (a), ideal (b), and too fast (c)

```
int cycleThresh = 3;
int cycleMax = 4;

//Count cycles
int xCycles = 0;
int xMid = (xA*25+xB);

for(int count=0; count<49; count++)
    if(xBuffer.at<float>(count) < xMid)
        if(xBuffer.at<float>(count+1) > xMid)
            xCycles++;

if(xCycles >= cycleThresh)
    palpCriteria[0][1] = true;

if(xCycles <= cycleMax)
    palpCriteria[0][2] = true;
```

Figure 4.7.8 Code snippet for the evaluation of the second criterion for correct palpation

In the system implementation, the number of cycles is determined by counting the number of horizontal axis crossings within the signal contained in the buffer. The horizontal axis is set at the median value,  $xMid$ . The number of cycles,  $xCycles$ , is



compared to two values: *cycleThresh* and *cycleMax*. These are the minimum and maximum number of allowable cycles to limit the speed of repetition to within 1.5 to 2 cycles per second. The same method is also applied to the Y signal.

Third, the user must be performing palpation on the location specified by the system at a specific point in the BSE procedure. In the integrated system, palpation locations are defined using blocks by the system. To determine if the palpations are performed on the proper locations, the centroid of the hand motion is used. This is calculated by determining the center of mass of the sampled signal. The center of mass of the X and Y signals within the two buffers must be within the region defined by the block as shown in Figure 4.7.9.

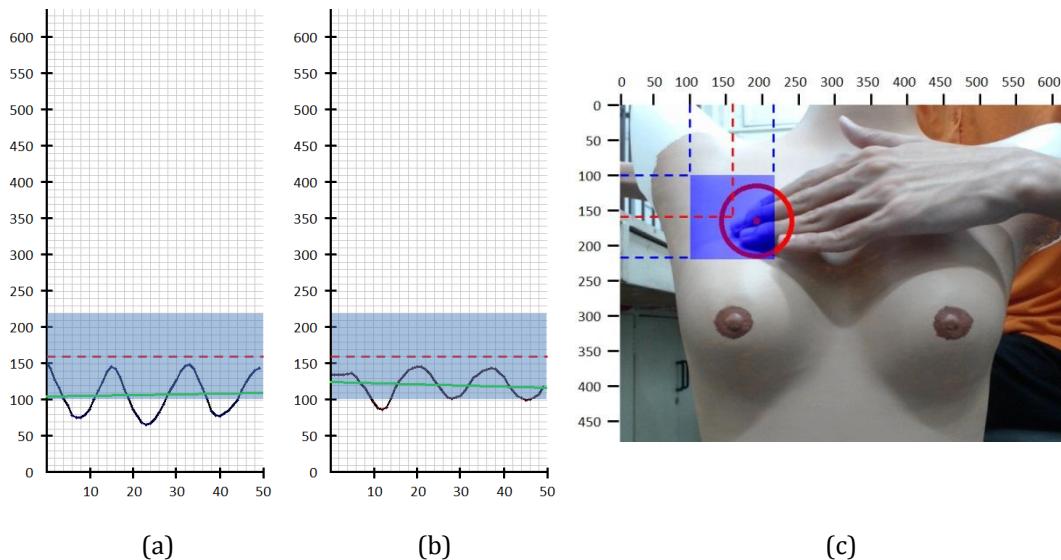


Figure 4.7.9

Sample X signal (a) and Y signal (b) for palpation motion performed within the specified block (blue region) (c)

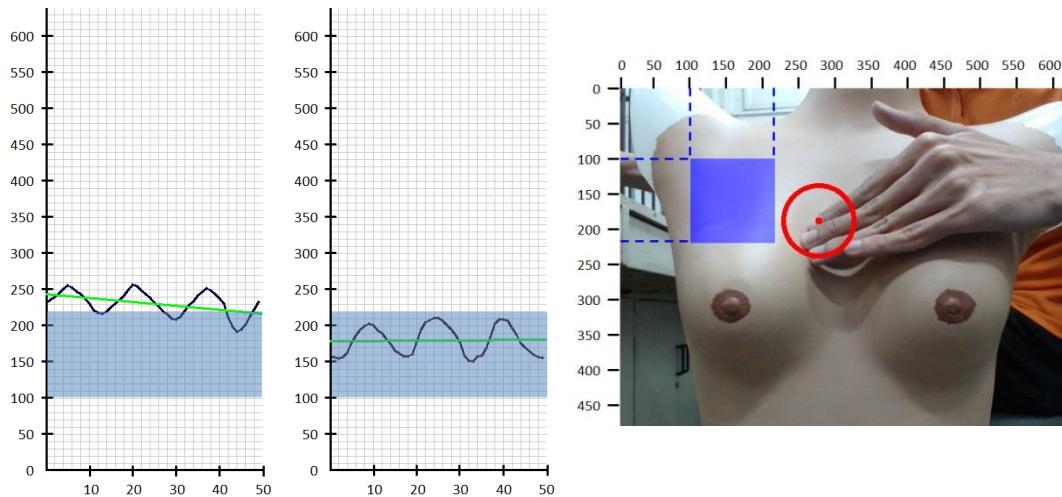


Figure 4.7.10

Sample X signal (a) and Y signal (b) for palpation motion performed outside the specified block (blue region) (c); The Y signal lies within the allowable region, the X signal however does not.

The center of gravity of the buffered hand motion is defined by the median values  $xMid$  and  $yMid$ . To qualify for proper palpation,  $xMid$  and  $yMid$  of the current signal must be within the bounding box of the current block. This is implemented using a simple if-else statement.

```
int xMid = (xA*25+xB);
int yMid = (yA*25+yB);

if(xMid > currentBlock.x && xMid < currentBlock.x+currentBlock.width)
    palpCriteria[0][3] = true;

if(yMid > currentBlock.y && yMid < currentBlock.y+currentBlock.height)
    palpCriteria[1][3] = true;
```

Figure 4.7.11 Code snippet for the evaluation of the third criterion for correct palpation

Last, palpations must be done in small coin-sized circular motions. In this implementation, palpations must at least be of the same size with the region of interest, i.e. blocks previously defined in the breast area division algorithm. The



radius of the circular motion during palpation is represented by the amplitude peaks in the X and Y signals as shown in Figure 4.7.12.

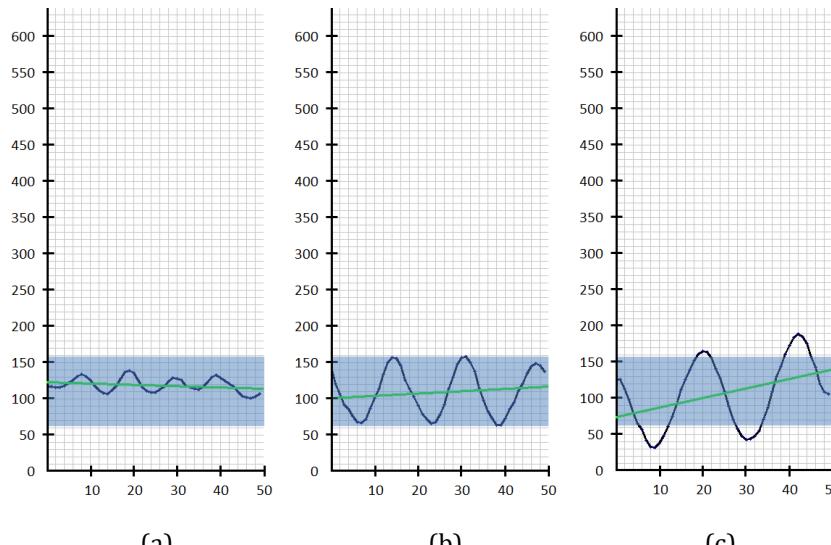


Figure 4.7.12 Sample X signals showing hand motions of various sizes: too small (a), ideal (b), and too large (c)

In here, the size of the circular motions is estimated by calculating the signal energy of the sampled X and Y signals. The energy of the signal within each buffer is calculated using the equation below where  $z$  is the normalized value and  $t$  is the time index.

$$E_z = \sum_{t=0}^{49} |z(t)|^2 \quad \text{Equation 4.7.3}$$

Ideal palpation is characterized by a sinusoidal signal with a frequency of 1.5 Hz. Using this assumption, the total signal energy for an ideal palpation is described by the equation below where  $A$  corresponds to the width or height of the block.

$$E_z = A \times 25^2 \quad \text{Equation 4.7.4}$$



The equation above is a function of the width or height of the block. Hence, for each of the X and Y signals, a different threshold is used. The allowable range of signal energy is within 30% to 70% of the computed threshold.

```
float minimumSize = (float)(0.3*25*25*currentBlock.width);
float maximumSize = (float)(0.7*25*25*currentBlock.width); //Count cycles

float palpSize = 0;
for(int count=0; count<49; count++)
    palpSize = palpSize + pow(xBufferTemp.at<float>(count),2);

if(palpSize > minimumSize)
    palpCriteria[0][4] = true;

if(palpSize < maximumSize)
    palpCriteria[0][5] = true;
```

Figure 4.7.13 Code snippet for the evaluation of the last criterion for correct palpation

In the system implementation, the size of palpation, *palpSize*, is calculated using Equation 4.7.3. To qualify as proper palpation, *palpSize* is compared to two values: *minimumSize* and *maximumSize*. The lower threshold, *minimumSize*, is calculated as 30% of the threshold value calculated using Equation 4.7.4. The upper threshold, *maximumSize*, is calculated as 70% of this value.

## 4.7.5 Detecting Palpation

It was mentioned above that a correct palpation must satisfy the four criteria explained in the previous section. Each criterion applies to both the X and Y signals. Therefore, a correct palpation must satisfy a total of 8 conditions. However, the proposed experimental setup is limited to only what the webcam can see. The breast surface is far from being planar. Therefore, palpation motions are not exactly circular to the point of view of the camera. As shown in Figure 4.7.14, the breast area is roughly divided into 4 classes distinguished by 4 different colors.

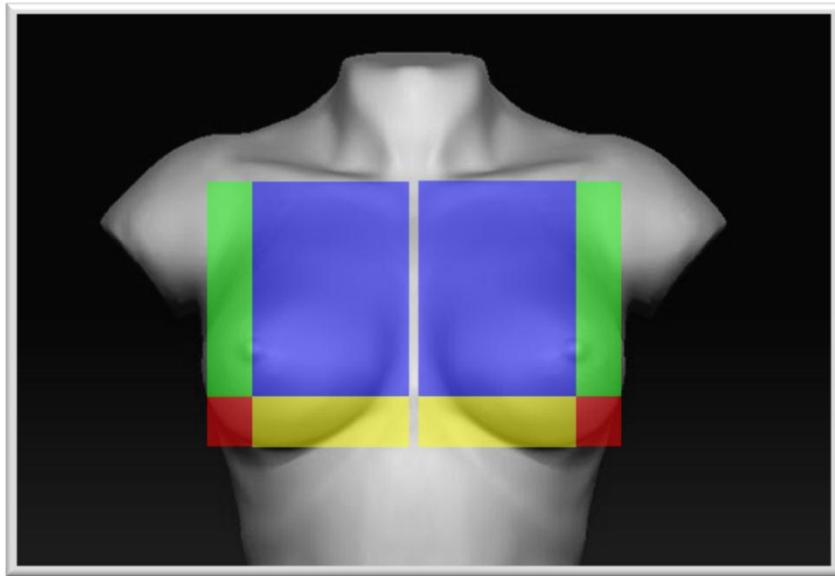


Figure 4.7.14 Sample X signals showing non-palpation motion (a) and palpation motion (b)

The blue region, or the inner breast regions, is the most planar to the camera's point of view. Therefore, all 8 conditions can be employed to detect palpations. The green region, or the outer breast regions, is characterized by breast surfaces which are tilted about the vertical axis. Oscillations in the X-axis will have small amplitudes. Therefore, the size criteria on the X signal can be disregarded. On the opposite hand, the size criteria on the Y signal can be disregarded for the yellow region, or lower breast regions. Last, for the red region, the size criteria for both X and Y signals can be ignored.

## 4.8 System Integration

Using the individual algorithms presented in this chapter, a complete BSE instruction and supervision system is developed. All four algorithms are integrated into one complete system. Additionally, a graphical user interface (GUI) is created in



order to facilitate ease of use for the user together with real-time instructions and corrective feedback in the form of textual, graphical, and audio cues.

### 4.8.1 System Overview

The complete system can be visualized using the block diagram below. It is comprised of four main parts namely: Initialization, Monitoring & Evaluation, Feedback, and Hardware.

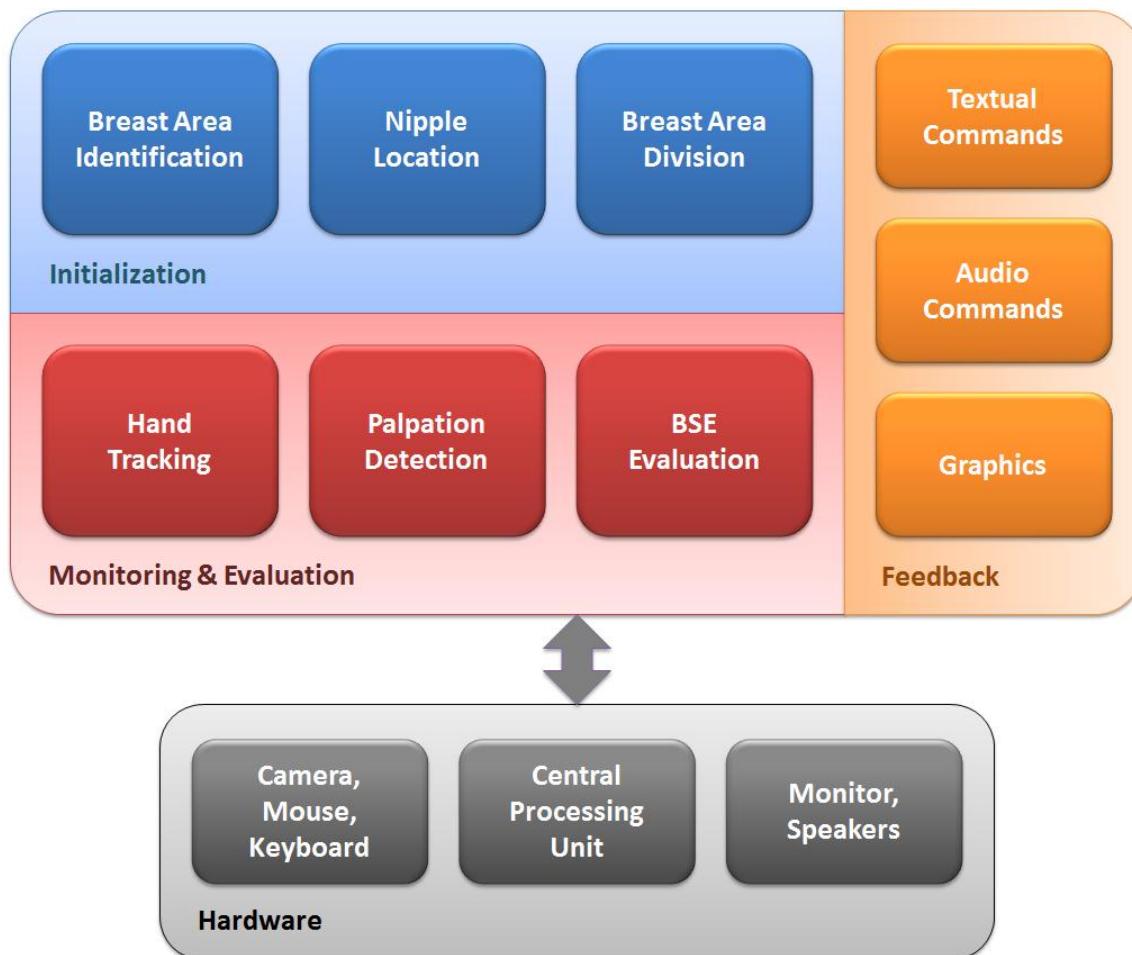


Figure 4.8.1 General Block Diagram of the BSE Instruction and Supervision System



The hardware part has already been explained in Section 4.1. The Initialization and Monitoring & Evaluation parts comprise the intelligence component of the system. They consist of the individual computer vision algorithms presented in this chapter. The feedback part is comprised of additional features such as text, audio, and graphics to provide the user instructions, corrective feedback, and updates regarding the breast self-exam.

### 4.8.2 Process Overview

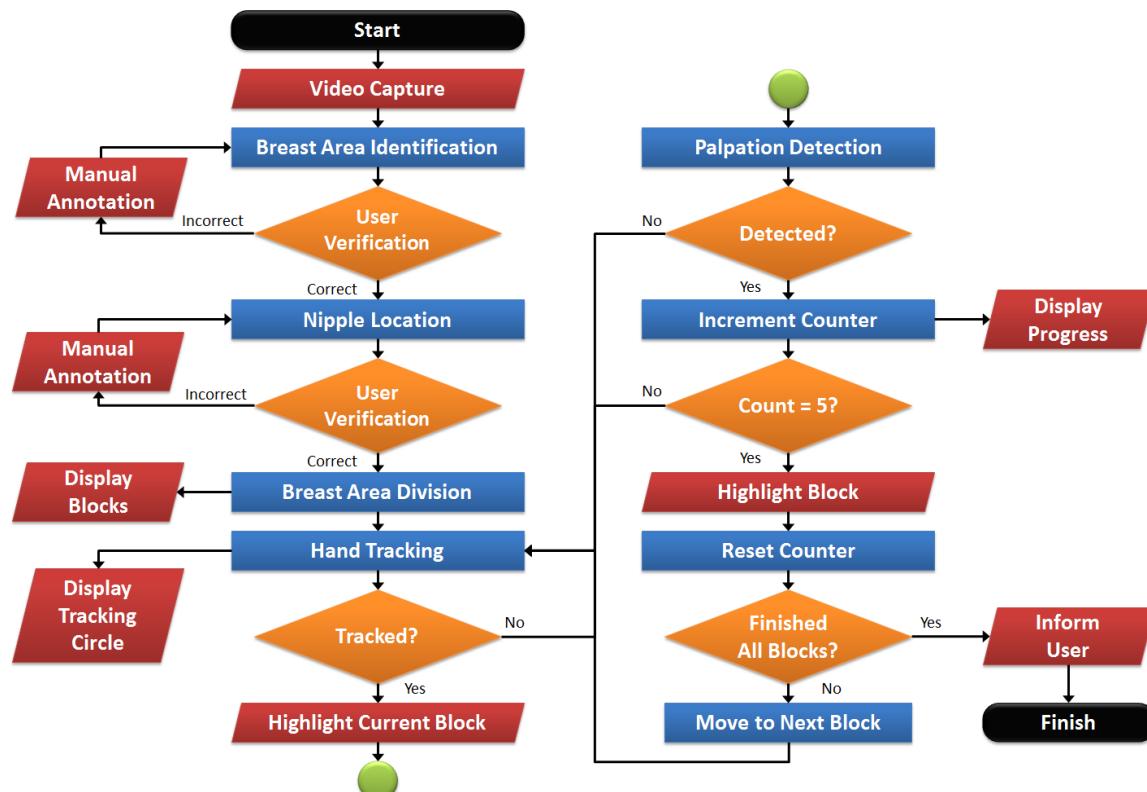


Figure 4.8.2 Flowchart of the complete BSE Instruction and Supervision System

A complete BSE trial using the integrated system follows the flowchart shown in Figure 5.4.4. The process starts with identifying the breast area using the algorithm



presented in Section 4.5. The user is asked to verify if the breast area was identified correctly. If not, the user shall manually annotate the correct breast area. Next, the nipples will be automatically located using the algorithm described in Section 4.5. Again, the user is asked to verify the results and if the algorithm failed to correctly locate the nipple, the user shall manually pinpoint the nipple location. Next, the system will divide the breast area accordingly into quadrants and blocks.

After characterizing the breast area, the system will continuously detect and track the hand of the user in real time. In parallel with this, the system will monitor and detect possible palpation motions. Detected palpation motions are further evaluated and if they satisfy the criteria for correct palpation enumerated in Section 4.7.1, the palpation counter is incremented. If the counter reaches the threshold, i.e. 5, the block is deemed as completely palpated and highlighted graphically to inform the user. The user is asked to palpate the next block after this. This process is repeated until all blocks have been sufficiently palpated. In which case, the BSE has already been completed.



### 4.8.3 Graphical User Interface

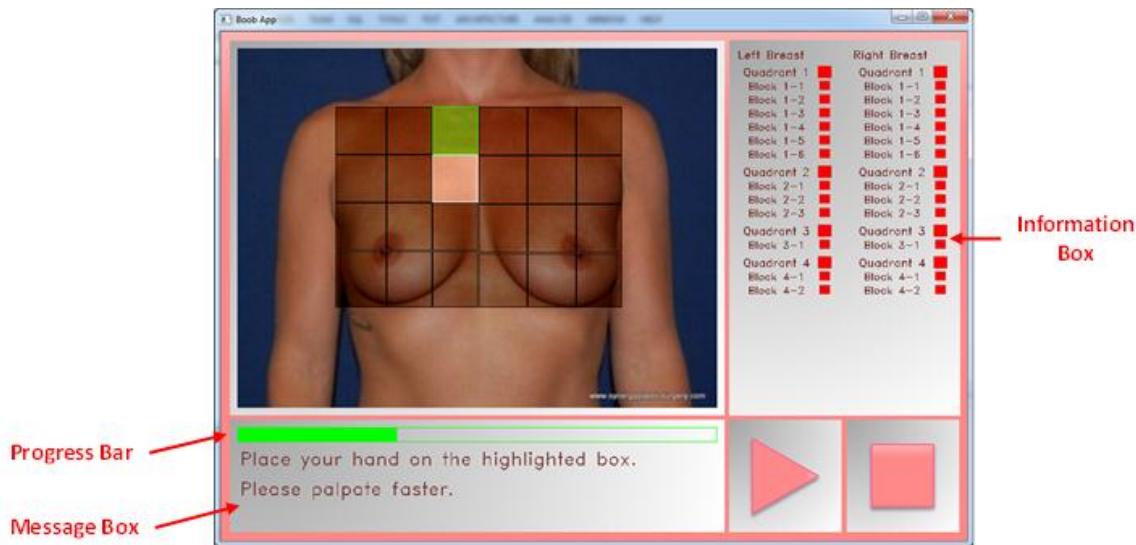


Figure 4.8.3      Graphical User Interface (GUI) of the integrated system

As shown in the figure above, the graphical user interface is comprised of a couple of buttons for user control, a message box, an information box, a progress bar, and a webcam feed. The webcam feed is continuously updated. The breast area division through blocks is drawn on the webcam feed. The current block is highlighted as a white block. The progress bar shows the percentage of completeness of palpation for the current block. If a block has already been completed, it is highlighted as green. The message box shows instructions and corrective feedback to the user. The information box shows additional information such as the current status of the BSE, etc.



## Chapter 5

### Results and Discussion

#### 5.1 Performance Measurement

To ensure that the complete system is working accordingly, the individual algorithms will be tested separately and their performance measured using a specific metric. Performance measurements to be conducted in this study are divided into three categories. The first category is concerned with the correctness of the tracked coordinates. The second category is concerned with the correctness of the tracked area. The third category is concerned with binary classification performance. A summary of the experiments to be conducted including the data sets and performance metrics used is shown at the end of this section.

##### 5.1.1 Tracking Errors

In the context of tracking, there are three basic types of errors:

1. Deviation: the distance between the tracking result and the ground truth
2. False positive: the tracker identifies a target which is not a target
3. False negative: the tracker fails to identify and locate the target

##### 5.1.2 F-score

F-score is a measure of accuracy which considers both precision and recall of the tracking algorithm [67]. Precision, also called positive predictive value, is the fraction of positive tracking results which are correct. Recall, also called sensitivity,



is the fraction of correct results which are positive. In using the F-score as a measure of tracking performance, tracking results must first be classified as either true positive, true negative, false positive, or false negative. The calculation of precision, recall and ultimately the F-score is done using the formulas below where  $N_{tp}$  is the number of true positives,  $N_{fp}$  is the number of false positives, and  $N_{fn}$  is the number of false negatives.

$$precision = \frac{N_{tp}}{N_{tp} + N_{fp}} \quad \text{Equation 5.1.1}$$

$$recall = \frac{N_{tp}}{N_{tp} + N_{fn}} \quad \text{Equation 5.1.2}$$

$$F = 2 \times \frac{precision \times recall}{precision + recall} \quad \text{Equation 5.1.3}$$

### 5.1.3 Deviation

Deviation is also used as a tracking accuracy measure as shown in [68]. Deviation is the error of the tracking location expressed in pixels. In this context, it is a measure of the mean deviation for all frames. Deviation is computed using the formula below where  $d(T^i, GT^i)$  refers to the pixel distance error while  $M_s$  refers to the set of frames where the target is correctly tracked.

$$deviation = 1 - \frac{\sum_{i \in M_s} d(T^i, GT^i)}{|M_s|} \quad \text{Equation 5.1.4}$$

### 5.1.4 F1-score

F1-score is an area-based variant of the F-score [69]. It is very similar to the F-score except for different measures of precision and recall which are now more concerned with area rather than just a binary classification of either true or false. The F1-score provides insight regarding the average area covered by the tracked



bounding box,  $T^i$ , and the ground truth bounding box,  $GT^i$  [70]. The F1-score is calculated using the formulas below.

$$p^i = |T^i \cap GT^i| / |T^i| \quad \text{Equation 5.1.5}$$

$$r^i = |T^i \cap GT^i| / |GT^i| \quad \text{Equation 5.1.6}$$

$$F1 = \frac{1}{N} \sum_i 2 \times \frac{p^i \times r^i}{p^i + r^i} \quad \text{Equation 5.1.7}$$

## 5.1.5 Binary Classification Metrics

Binary classification is the task of classifying input data into either one of two groups, commonly either positive or negative. In this study, five measures of binary classification will be used. The formulas for calculating these metrics are shown below [71].

$$TPR = \frac{\sum \text{True positive}}{\sum \text{Actual positive}} \quad \text{Equation 5.1.8}$$

$$TNR = \frac{\sum \text{True negative}}{\sum \text{Actual negative}} \quad \text{Equation 5.1.9}$$

$$PPV = \frac{\sum \text{True positive}}{\sum \text{Predicted positive}} \quad \text{Equation 5.1.10}$$

$$NPV = \frac{\sum \text{True negative}}{\sum \text{Predicted negative}} \quad \text{Equation 5.1.11}$$

$$ACC = \frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total}} \quad \text{Equation 5.1.12}$$

Sensitivity or true positive rate (TPR) is the ratio of correctly classified positives to the total number of actual positives. Specificity or true negative rate (TNR) is the ratio of correctly classified negatives to the total number of actual negatives. Positive predictive value (PPV) is the ratio of correctly classified positives to the total number of predicted positives. Negative predictive value (NPV) is the ratio of



correctly classified negatives to the total number of predicted negatives. Last, accuracy (ACC) is the ratio of the total number of correct classifications to the total number of classifications performed.

## 5.2 Breast Area Identification

For this part of the study, Data Set 1, which is comprised of 200 images of the female torso, is used. For each image, the ground truth values, i.e. bounding box of left and right breasts, were manually annotated. And for this part of the study, the developed algorithm's performance is compared to the past algorithm developed by E. Mohammadi Nejad in his master's thesis [38]. The performance metric used here is the area-based F1-score. For each input image, the F1-score calculation covers the areas of both the left and right breasts.

### 5.2.1 Results

The performance of the old and new algorithms is summarized in the following table and figures.

Table 5.2.1 Performance of Old and New Breast Area Identification Algorithms

	Old Algorithm	New Algorithm
$p_i$	0.894	0.935
$r_i$	0.428	0.973
F1-score	0.576	0.953

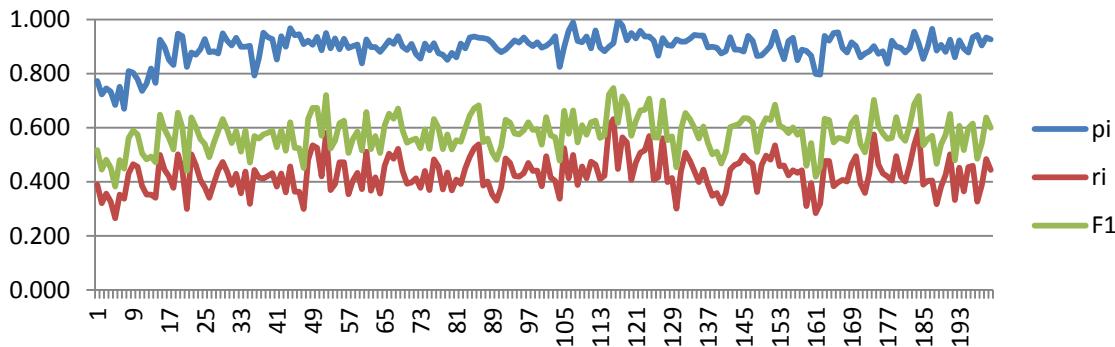


Figure 5.2.1 Performance results for the 200 test images using the Old Algorithm

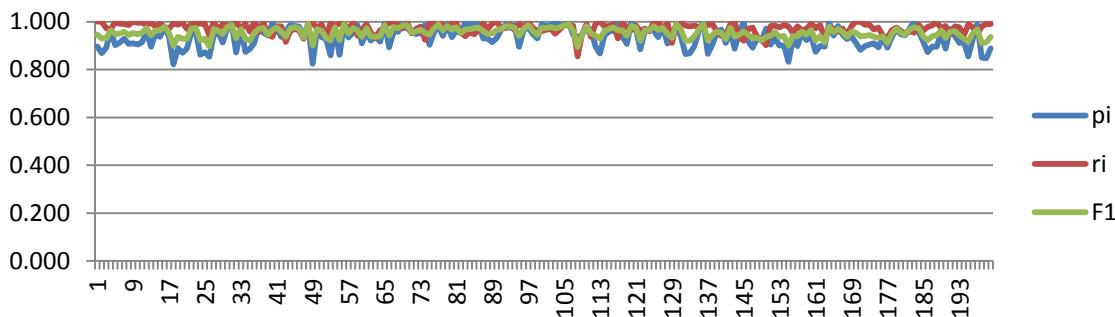


Figure 5.2.2 Performance results for the 200 test images using the New Algorithm

## 5.2.2 Discussion

The new algorithm developed in this study boasts a greatly improved F1-score of approximately 0.953 over the old algorithm's F1-score of only 0.576. The F1-score is specifically selected as the performance metric to emphasize the only important task of the Breast Area Identification Algorithm, which is to correctly identify the boundaries of the breast areas.

The F1-score is comprised of two components:  $p_i$  and  $r_i$ . The  $p_i$  component corresponds to the ratio of the correctly identified area to the total area of the tracked bounding box. On the other hand, the  $r_i$  component corresponds to the ratio



of the correctly identified area to the total area of the ground truth bounding box. As seen in Figure 5.2.1, the  $p_i$  scores of the Old Algorithm are relatively high while its  $r_i$  scores are very low. A high  $p_i$  score means that the resulting bounding box of the old algorithm mostly covers true breast area. However, its low  $r_i$  score means that it is not able to cover much of the overall true breast area. Hence, the Old Algorithm's overall F1-score of 0.576 is very low to be considered effective in identifying the entire breast area.

On the other hand, as seen in Figure 5.2.2, the New Algorithm's  $p_i$  and  $r_i$  scores are both consistently high with average values of 0.935 and 0.973 respectively. This means that the resulting bounding box for each test sample includes mostly true breast area and at the same time covers most of the entire true breast area. Hence, the New Algorithm's overall F1-score of 0.953 proves that it is very effective in identifying the entire breast area.

The significant difference in performance between the old and new algorithms lies in the respective methodologies used. Instead of using assumptions with no factual basis, the new algorithm used actual visual features which truly characterize the breast boundaries. Hence, this justifies the new algorithm's importance. Shown in Figure 5.2.3 and Figure 5.2.4 are sample results for the two algorithms.

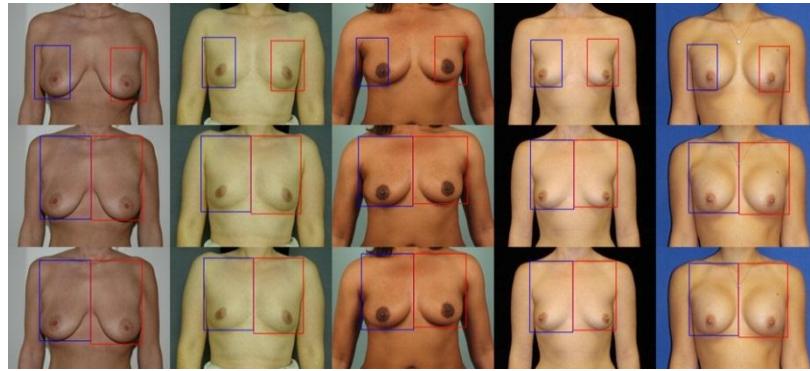


Figure 5.2.3 Sample images from Data Set 1 showing the results of the Old Algorithm (top), results of the New Algorithm (middle), and ground truth (bottom)

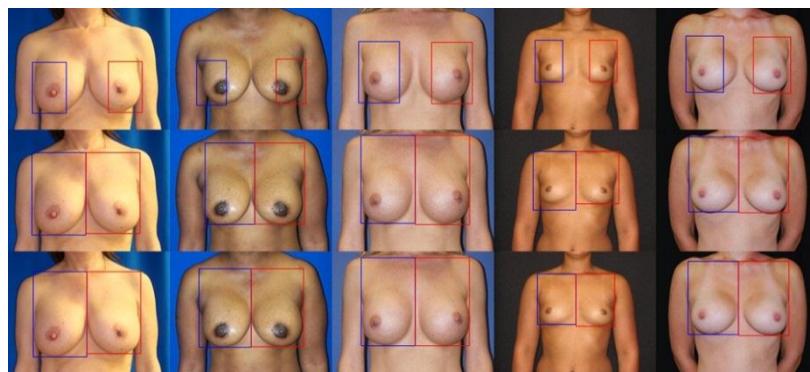


Figure 5.2.4 Sample images from Data Set 1 showing the results of the Old Algorithm (top), results of the New Algorithm (middle), and ground truth (bottom)

### 5.3 Nipple Location and Breast Area Division

For this part of the study, Data Set 1, which is comprised of 200 images of the female torso, is also used. For each image, the ground truth values, i.e. bounding box of left and right nipples, were manually annotated. The performance metrics used here are the F-score and deviation.



### 5.3.1 Results

For the calculation of the F-score, the tracking results are evaluated based on the overlap between the tracked bounding box and the ground truth bounding box. Tracking results are categorized as either a true positive or a false positive. A successful track is when the tracking result satisfies the Pascal criterion defined in the equation below.

$$|T^i \cap GT^i| / |T^i \cup GT^i| \geq 0.5 \quad \text{Equation 5.3.1}$$

Table 5.3.1 Performance of the Nipple Tracking Algorithm

	Left Nipple	Right Nipple	Overall
precision	0.910	0.875	0.893
recall	1.000	1.000	1.000
F1-score	0.953	0.933	0.943
Deviation	9.088	10.441	9.764

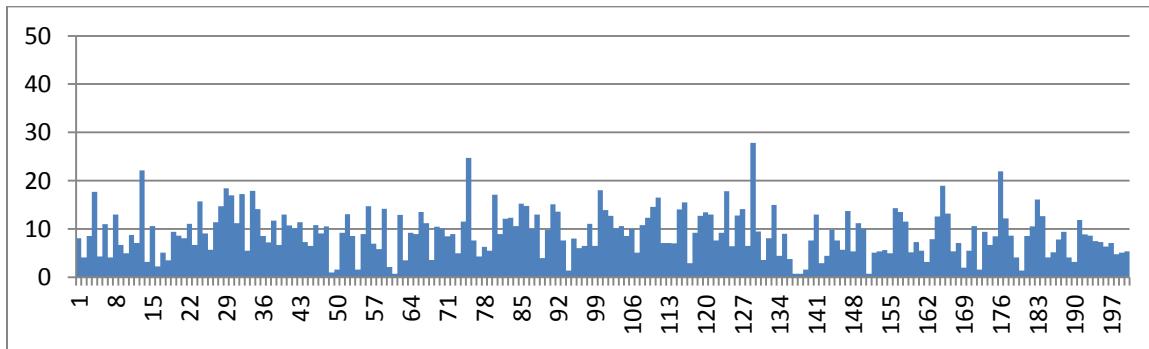


Figure 5.3.1 Left Nipple deviations for the 200 test images

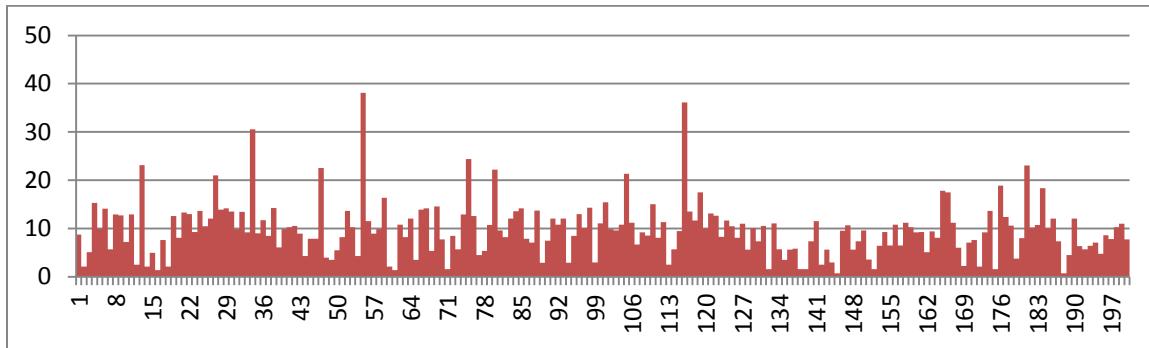


Figure 5.3.2 Right Nipple deviations for the 200 test images

### 5.3.2 Discussion

The nipple location algorithm boasts an overall F-score of 0.943 which proves that it is capable of efficiently locating the left and right nipples. It also yielded an average deviation of 9.764 pixels. As seen in Figure 5.3.1 and Figure 5.3.2, nipple centroid deviations from the ground truth centroid are kept under 20 pixels except from a few spikes which indicate failed tracking.

The centroids of the left and right nipples are used to divide the left and right breasts respectively into four quadrants. Shown in Figure 5.3.3 are samples of the Nipple Location results and in Figure 5.3.4 are samples of the Breast Area Division results.

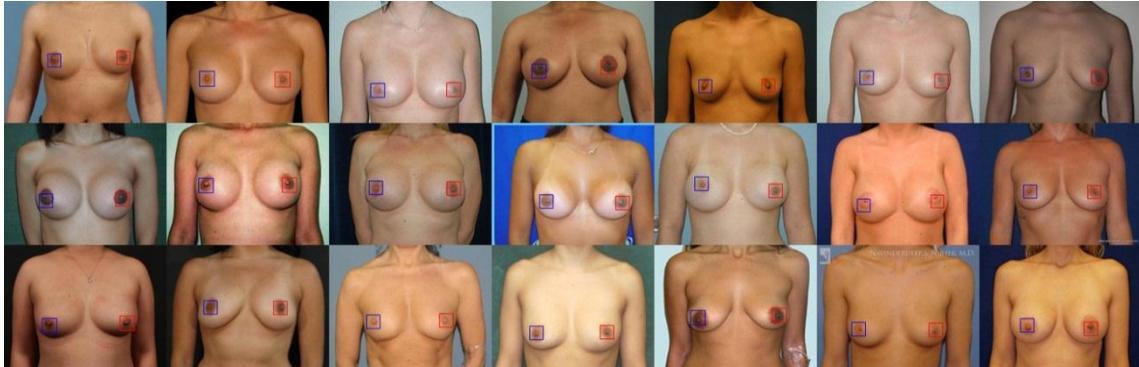


Figure 5.3.3 Sample images from Data Set 1 showing the tracked left and right nipples using the Nipple Location Algorithm

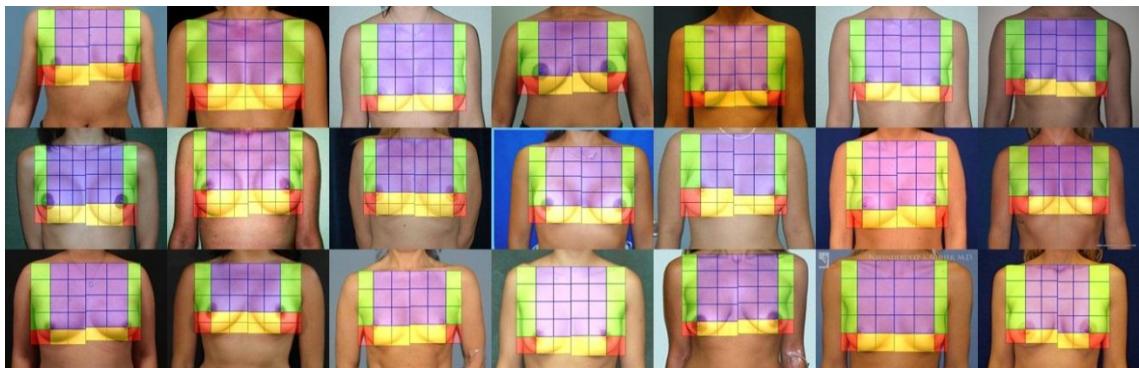


Figure 5.3.4 Sample images from Data Set 1 showing the breast area divided into 4 quadrants and further into smaller blocks using the Breast Area Division Algorithm

## 5.4 Hand Tracking

For this part of the study, Data Set 2, which is comprised of 14 video snippets of women palpating their breasts, is used. For each video, the ground truth values, i.e. coordinates of the finger pads per frame, were manually annotated. And for this part of the study, the developed algorithm's performance is compared to the past algorithm developed by Hu et al [36]. The performance metrics used here are the F-score and deviation.



## 5.4.1 Results

The tracking results for the 14 videos in Data Set 2 using Hu et al's algorithm is shown in the table below.

Table 5.4.1 Performance of Hu et al's Hand Tracking Algorithm

	Precision	Recall	F-score	Deviation
D2-01.avi	0.6151	0.6457	0.6300	49.88
D2-02.avi	0.8015	0.6587	0.7231	32.17
D2-03.avi	0.8742	0.7133	0.7856	26.86
D2-04.avi	0.7424	0.6325	0.6831	34.36
D2-05.avi	0.9883	0.7824	0.8734	19.17
D2-06.avi	0.8600	0.6973	0.7701	27.83
D2-07.avi	0.7922	0.6170	0.6983	35.46
D2-08.avi	0.6629	0.5866	0.6224	37.20
D2-09.avi	0.9915	0.7620	0.8617	13.72
D2-10.avi	0.9413	0.6364	0.7594	21.05
D2-11.avi	0.8863	0.7096	0.7882	26.59
D2-12.avi	0.3763	0.4474	0.4088	48.68
D2-13.avi	0.8896	0.6979	0.7810	28.17
D2-14.avi	0.8343	0.6575	0.7354	30.42
<b>Overall</b>	<b>0.7935</b>	<b>0.6602</b>	<b>0.7208</b>	<b>30.64</b>

The tracking results for the 14 videos in Data Set 2 using the newly developed algorithm is shown in the following table.



Table 5.4.2 Performance of the newly developed Hand Tracking Algorithm

	Precision	Recall	F-score	Deviation
D2-01.avi	0.9990	0.9948	0.9969	14.96
D2-02.avi	0.9456	0.8945	0.9193	31.50
D2-03.avi	0.8489	0.7903	0.8186	36.21
D2-04.avi	0.8621	0.8634	0.8628	36.69
D2-05.avi	1.0000	0.9954	0.9977	16.60
D2-06.avi	1.0000	0.9709	0.9852	33.11
D2-07.avi	0.9538	0.8790	0.9149	28.50
D2-08.avi	0.9721	0.9051	0.9374	24.05
D2-09.avi	1.0000	0.9241	0.9605	7.80
D2-10.avi	1.0000	0.9968	0.9984	17.46
D2-11.avi	0.9648	0.9145	0.9390	27.54
D2-12.avi	1.0000	0.9931	0.9965	19.82
D2-13.avi	1.0000	0.9879	0.9939	22.61
D2-14.avi	0.9958	0.9573	0.9761	22.85
<b>Overall</b>	<b>0.9653</b>	<b>0.9277</b>	<b>0.9461</b>	<b>24.33</b>

A comparison of Hu et al's algorithm and the new algorithm in terms of their precision, recall, and F-score are shown in the following figures.

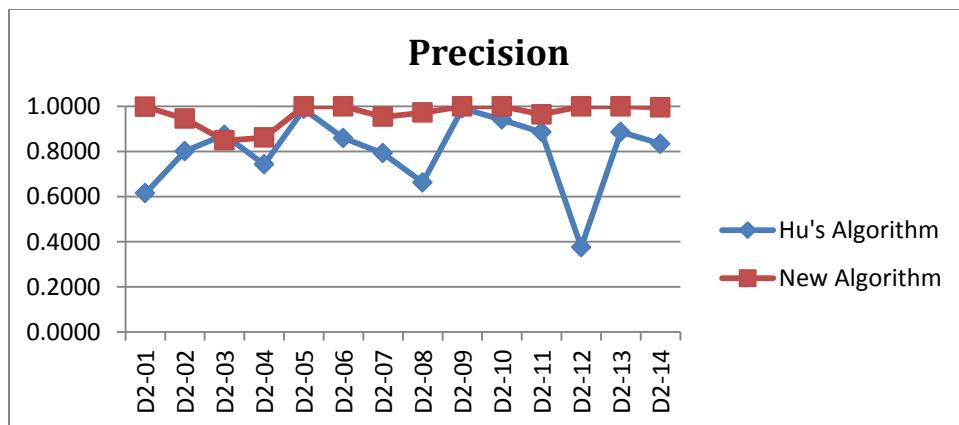


Figure 5.4.1 Precision values for the 14 videos in Data Set 2 using Hu's algorithm (blue) and the new algorithm (red)

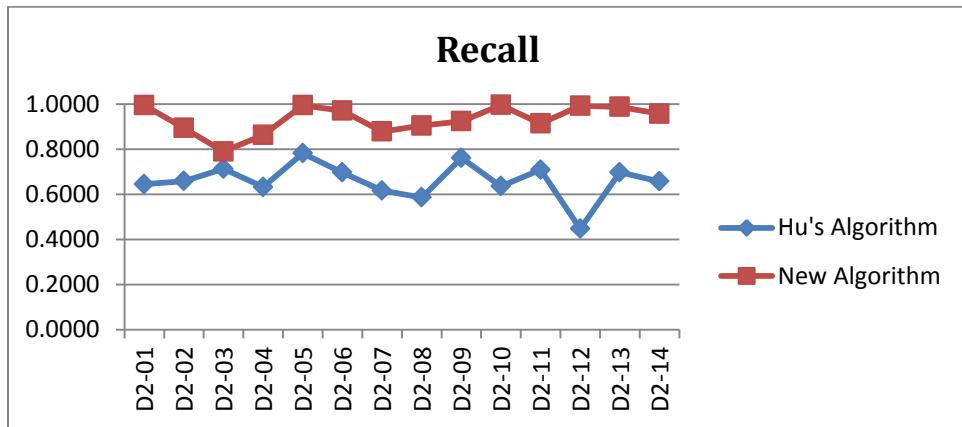


Figure 5.4.2 Recall values for the 14 videos in Data Set 2 using Hu's algorithm (blue) and the new algorithm (red)

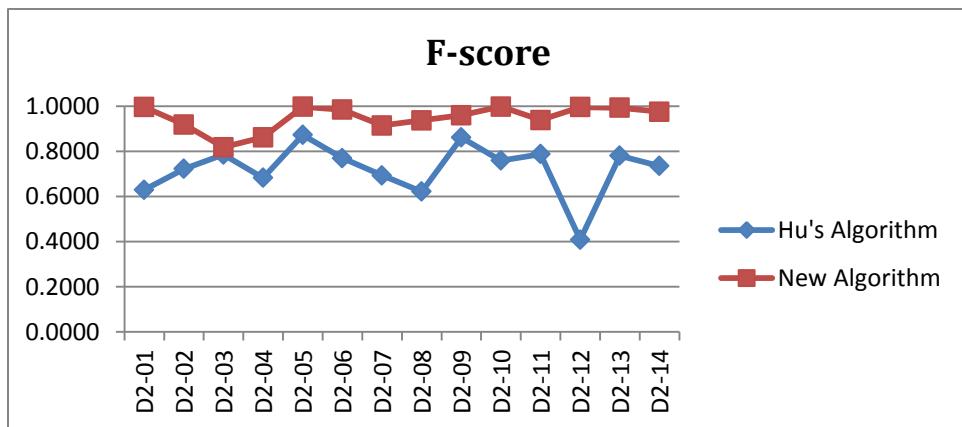


Figure 5.4.3 F-scores for the 14 videos in Data Set 2 using Hu's algorithm (blue) and the new algorithm (red)

## 5.4.2 Discussion

The new algorithm boasts an improved F-score of 0.9461 over the old algorithm's F-score of 0.7208. As seen in Figure 5.4.1, the precision of the old algorithm is relatively low compared to the new algorithm's precision. This is due to the old algorithm being dependent on hand motion for it to successfully track the hand. This results to erroneous tracking if there is no significant hand motion.



As seen in Figure 5.4.2, the recall values of the old algorithm are constantly lower than the recall of the new algorithm by a significant level. This is due to large amounts of false negatives, i.e. failed tracking, when using the old algorithm. False positives in the old algorithm occur when the hand is not moving at all. Unlike the old algorithm, the new algorithm is capable of tracking the hand even if it is not moving.

Due to the difference in precision and recall values between the old and new algorithms, the F-score of the old algorithm is of course much lower than the F-score of the new algorithm. This proves that feature tracking, which was used in the new algorithm, is a better method for tracking the hand compared to motion segmentation, which was used in the old algorithm.

The deviation ratings of the two algorithms, on the other hand are not significantly different. The deviation of the new algorithm is just a little better at 24.33 pixels compared to the deviation of the old algorithm at 30.64 pixels. Nevertheless, the new algorithm proved to be better.

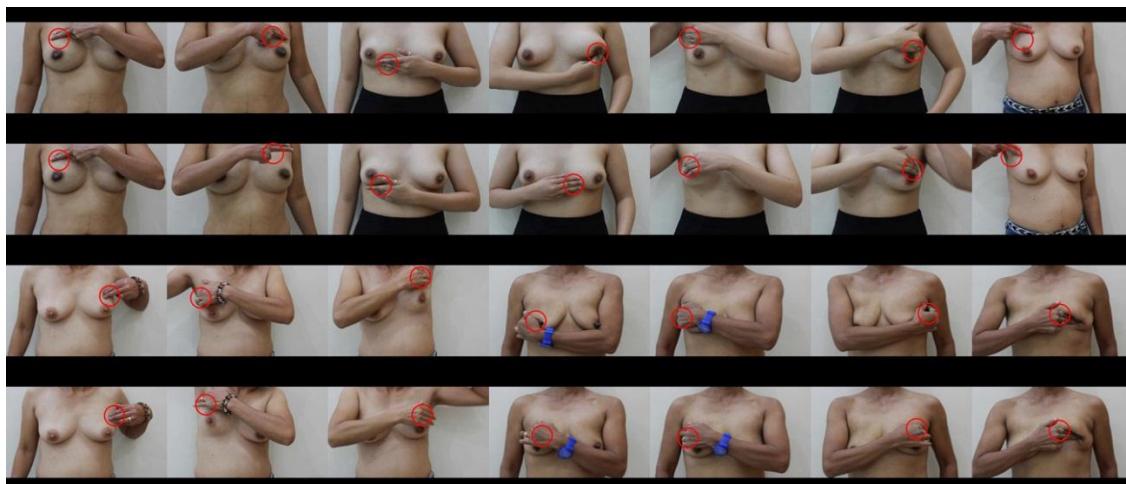


Figure 5.4.4      Sample frames from the 14 videos in Data Set 2 showing the tracking results (red circle) using the new algorithm



## 5.5 Palpation Detection

### 5.5.1 Part 1: Evaluation of the New Algorithm

For this part of the study, selected videos from Data Set 2 were used. These are D2-01, D2-02, D2-11, D2-12, D2-13, and D2-14. These are the only videos in which the palpation technique used conforms to the one described in this study. Specifically, D2-01 and D2-02 used repetitive horizontal back and forth motion while D2-11 up to D2-14 used the recommended circular motion. For these 6 videos, only the 1<sup>st</sup> (repetitiveness) and 2<sup>nd</sup> (speed) criteria for correct palpation were used to detect palpations. The other criteria, i.e. location and size, were not used because the palpation methods used in the videos were beyond the control of this study.

Since the palpation detection algorithm is dependent on the hand tracking algorithm, experiments were conducted on the entire length of each video. Hand trajectories were manually annotated as either positive or negative. Specific hand trajectories which were classified are the following: static hand, hand position changes, and palpation motions. The performance metrics used in this experiment are binary classification metrics such as TPR, TNR, PPV, NPV, and ACC.

### 5.5.2 Part 1: Results and Discussion

The results of experiments conducted on the 6 videos using the Palpation Detection Algorithm are summarized in the following tables.



Table 5.5.1 Binary classification results of the Palpation Detection Algorithm

	True Positives	True Negatives	False Positives	False Negatives
D2-01.avi	8	12	0	4
D2-02.avi	11	11	0	0
D2-11.avi	19	17	2	1
D2-12.avi	7	6	0	0
D2-13.avi	14	13	2	0
D2-14.avi	15	14	1	1
<b>Total</b>	<b>74</b>	<b>73</b>	<b>5</b>	<b>6</b>

Table 5.5.2 Confusion matrix for the Palpation Detection Algorithm

		Predicted	
		Positive	Negative
Actual	Positive	74	6
	Negative	5	73

The performance of the Palpation Detection Algorithm measured using binary classification metrics is summarized in the following table.

Table 5.5.3 Performance of the Palpation Detection Algorithm

<b>True Positive Rate (TPR)</b>	<b>0.925</b>
<b>True Negative Rate (TNR)</b>	<b>0.936</b>
<b>Positive Predictive Value (PPV)</b>	<b>0.937</b>
<b>Negative Predictive Value (NPV)</b>	<b>0.924</b>
<b>Accuracy (ACC)</b>	<b>0.930</b>

The newly developed Palpation Detection algorithm yields a very high accuracy of approximately 0.93. Using videos comprised of around 80 positive palpations motions and 78 non-palpation motions, the developed algorithm scored a PPV of 93.7% and a TPR of 92.5%. This means that most of the positive detections are real



palpation motions and that the algorithm was able to detect almost all positive palpation motions. Also, the algorithm scored an NPV of 92.4% and a TNR of 93.6%. This means that most of the motions which were not classified as palpation were truly non-palpation motions and that the algorithm was able to classify almost all non-palpation motions correctly. These ratings prove the efficiency of the developed palpation detection algorithm.

### 5.5.3 Part 2: Comparative Evaluation of the Old and New Algorithms

Direct comparison between the palpation detection algorithm developed previously by E. Mohammadi Nejad [38] and the new algorithm presented in this study is not possible. Despite being measured using similar metrics, performance results of the old algorithm presented in [38] and performance results of the new algorithm presented in Section 5.5.2 cannot be used as a reliable basis for comparison of the two algorithms. This is due to the differences between the implementations of the two algorithms as well as the data set used in their corresponding experiments.

In order to compare the performance of each algorithm, a standard procedure must be established first. In the following experiments, the standard procedure is characterized as follows. First, the same data set is used for both algorithms. This is comprised of 12 videos which cover a complete BSE over the left breast of a mannequin. Second, since both algorithms are binary classifiers, they must classify elements of similar units. In their original implementations, the old algorithm classifies each video frame as positive or negative. On the other hand, the new algorithm classifies entire hand motions as positive or negative. To cater to the old algorithm's requirements, the results of the new algorithm will be interpreted in



terms of video frames. Third, the performance metrics to be used for both algorithms will be the following: TPR, TNR, PPV, NPV, and ACC.

#### 5.5.4 Part 2: Results and Discussion

The results of experiments conducted on the 12 videos using the old and new algorithms are summarized in the following tables.

Table 5.5.4 Binary classification results of the Old and New Palpation Detection Algorithms

	True Positives	True Negatives	False Positives	False Negatives
Old Algorithm w/o filter (T= 50k)	746	768	506	721
Old Algorithm w/o filter (T= 75k)	476	861	413	991
Old Algorithm w/o filter (T= 100k)	306	922	352	1161
Old Algorithm w/o filter (T= 125k)	178	964	310	1289
Old Algorithm w/ filter (T= 50k)	989	511	763	478
Old Algorithm w/ filter (T= 75k)	914	861	413	991
Old Algorithm w/ filter (T= 100k)	909	656	618	558
Old Algorithm w/ filter (T= 125k)	760	711	563	707
New Algorithm	1194	1264	10	273



Table 5.5.5 Performance of the Old and New Palpation Detection Algorithms

	TPR	TNR	PPV	NPV	ACC
Old Algorithm w/o filter (T= 50k)	0.509	0.603	0.596	0.516	0.552
Old Algorithm w/o filter (T= 75k)	0.324	0.676	0.535	0.465	0.488
Old Algorithm w/o filter (T= 100k)	0.209	0.724	0.465	0.443	0.448
Old Algorithm w/o filter (T= 125k)	0.121	0.757	0.365	0.428	0.417
Old Algorithm w/ filter (T= 50k)	0.674	0.401	0.564	0.517	0.547
Old Algorithm w/ filter (T= 75k)	0.623	0.426	0.556	0.495	0.532
Old Algorithm w/ filter (T= 100k)	0.620	0.515	0.595	0.540	0.571
Old Algorithm w/ filter (T= 125k)	0.518	0.558	0.574	0.501	0.537
New Algorithm	0.814	0.992	0.992	0.822	0.897

The old algorithm was run over the data set in 8 iterations with varying threshold values and with or without the noise filter described in the paper for its motion detection part. Based from these results, the best version of the old algorithm is that which uses the noise filter with a threshold of 100k with a best accuracy of 57.1%. Increasing the threshold of the algorithm decreases its sensitivity to detecting motions characterized by its decreasing true positive and false positive rates. In turn, its true negative and false negative rates increased. This however ultimately results to a decreasing trend in its overall accuracy.

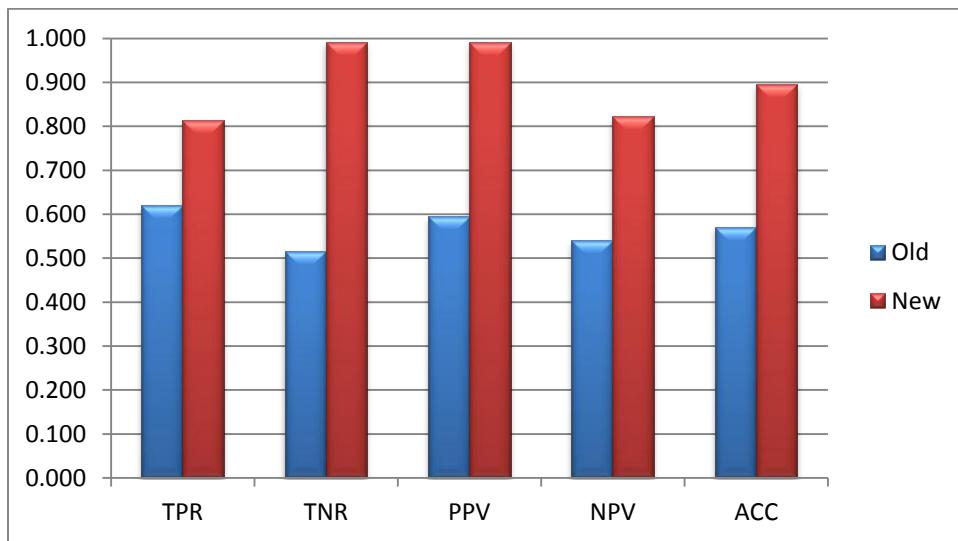


Figure 5.5.1      Performance of the Old Algorithm with threshold=50k and New Algorithm

Shown in Figure 5.5.1 is a graphical comparison of the best performance of the Old Algorithm and the performance of the New Algorithm. The new algorithm has significantly higher true positive rate and true negative rate. This means that the new algorithm is much better in correctly classifying hand motions as either palpation or non-palpation motions. The new algorithm also has higher positive and negative predictive values (PPV and NPV). This means that the new algorithm is able to correctly classify most of the actual positives and negatives. All these contribute to a higher accuracy for the new algorithm at 89.3% as compared to the best accuracy of the old algorithm at 57.1%.



## 5.6 Integrated System

To test the complete system, 5 complete BSE trials performed on a mannequin were run using the system. These are documented by taking a video capture of the system throughout each trial. Some video frames are shown in the figure below.

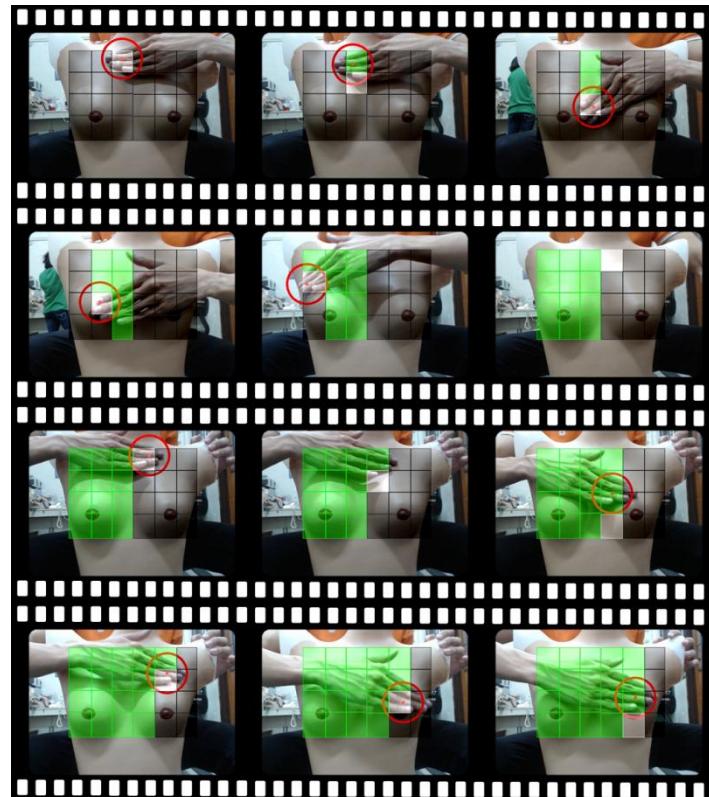


Figure 5.6.1 Sample frames from the BSE trials run using the complete system



## Chapter 6

### Conclusion and Recommendations

#### 6.1 Summary

In this thesis, a complete BSE instruction and supervision system was developed. In line with this, a number of computer vision algorithms essential to such a system were developed, tested, and validated. First, a breast area identification algorithm was developed to address the need for automatically identifying the entire area for breast self-examination. Second, a nipple location and breast area division algorithm was developed to address the need to divide the breast areas into quadrants and smaller blocks in which the user will be instructed to palpate sequentially. Third, a hand tracking algorithm was developed to address the need to monitor the hand motions. Fourth, a palpation detection algorithm was developed to address the need to automatically know if the user is performing palpation and evaluate whether she is doing it correctly. Last, the individual algorithms were integrated into a complete system together with features such as a graphical user interface and audiovisual feedback in order to address the need for a fully-functional BSE guidance system.

The developed algorithms were tested and their performances were evaluated using metrics specified in Section 5.1. Additionally, their performances were compared to existing related algorithms. Performance results are as follows: the breast area identification algorithm attained an F1-score of 95.30%, the nipple locating algorithm yielded an F-score of 94.30%, the hand tracking algorithm had an F-score of 94.61%, and the palpation detection algorithm had an accuracy of 93%.



The developed algorithms proved to be superior both quantitatively through these performance results and qualitatively through the additional features and functionality. All of the specified objectives were satisfactorily met.

## 6.2 Conclusion

One of the most significant contributions of this thesis is that it aligned the implementation of a computer vision-based BSE guidance system with a number of medical standards. First, the breast area defined here conforms to the standard breast exam area recommended by the International Agency for Research on Cancer (IARC). Second, the breast area was divided in this study in a manner which caters to the standard quadrant system used by physicians in relaying information about location of lumps, etc. Third, the system puts emphasis on the standard BSE procedure of using coin-sized circular motions and using the vertical strip method. The system divides the breast area into appropriately-sized blocks and defines palpation as circular and repetitive hand motions. Also, the system asks the user to palpate one block after another wherein the order of blocks to be palpated conforms to the vertical strip pattern.

Another significant contribution of this thesis is its implementation of a complete system with expanded intelligence. First, the system along with its numerous subsystems was created in a manner in which calibration is not necessary. An ideal system is that which needs minimal human intervention. Thus, the developed system in this thesis is intelligent enough to automatically adjust necessary parameters in its subsystems and is robust enough to handle a wide variety of possible input conditions. Second, the system sports a number of automated processes which makes the BSE procedure easier for its users. Some of these automated processes are the identification of the breast area, division of



breast area into quadrants and blocks, continuous hand detection and tracking, etc. Last and most importantly, the developed system is capable of providing corrective feedback to the user. Unlike the previous studies, the developed system in this thesis transcends mere detection and tracking of objects and events. The system now capable of understanding what the user is doing and evaluating whether the user is doing it correctly or not. Unlike the prototype system in the previous thesis which only detects palpation, the new system is now capable of knowing what the user is doing correctly and incorrectly. The system is able to inform the user what she is doing incorrectly and what she should do to correct it. This constitutes the “corrective feedback” which is the main purpose of using computer vision in BSE training.

In conclusion, this thesis produced a more comprehensive implementation of a BSE instruction and supervision system using computer vision. Using the works of BIOCORE and Mohammadi Nejad as basis for the functional requirements of such system, more robust and applicable algorithms have been developed and integrated together to come up with a complete system. Additionally, this thesis finally implements the corrective feedback function which has not been implemented in the previous studies. This is a significant step forward towards the deployment of a BSE guidance system which will be helpful in training women with breast self-examination.

### 6.3 Recommendations for Future Work

In this study, the user is instructed to not move her body throughout the BSE procedure. A possible improvement to this system is to integrate a body tracking algorithm in order to compensate for the small lateral movements of the body during the BSE procedure and to regulate the lateral movements of the user.



Also in this study, the palpation detection algorithm is limited to a 2-dimensional perspective of the camera feed. A possible improvement to this algorithm is to include a 3-dimensional reconstruction or at least an estimation of the camera feed in order to estimate the orientation of the surface which the user is palpating. This way, the shape and size of palpation motions can be better evaluated for all areas within the breast.

Last, an important part of BSE which was not addressed in this study is determining the palpation pressure levels. Future work should address this area in order to fully characterize and evaluate breast self-exams. Ultimately, this will enable a comprehensive training of women in performing BSE.



## Appendix A

### Medical Doctor's Comments

The quadrant examination used in this study is to simulate the actual examination by doctors on a female patient where the breast is divided into quadrants. This makes the localization more accurate. One recommendation I can give is to further subdivide the breast like a clock face; for example, the right outer quadrant at 2 o'clock location. The numbering is similar to that of a watch face. This may be good for future studies to include the exact location within the quadrant by clock numbers.

Another recommendation is that the program should also teach patients the correct depth of palpation on her skin. This is important because lumps can be located deep beneath the breast of a patient. The female patient must be able to cover the entire breast including this.

This computer guided examination may help patients even if the doctor is not present. Filipino women are known to be very conservative and shy even in examining their own breasts. So this program may help them overcome this especially if this is a self-guided type of exam where a doctor need not be present.

**Jose Ma. C. Avila, M.D.**

Fellow and Former President

Philippine Society of Pathologists, Inc.



## References

- [1] F. Bray, P. McCarron, and D.M. Parkin, "The changing global patterns of female breast cancer incidence and mortality," *Breast Cancer Research*, pp. 229-239, 2004.
- [2] World Health Organization, "Health profile: Philippines," World Health Organization, 2010.
- [3] A. Nover et al., "Modern breast cancer detection: A technological review," *International Journal of Biomedical Imaging*, pp. 1-15, 2009.
- [4] T. Reyes, "Newer ways to diagnose breast cancer," *The Philippine Star*, October 2008.
- [5] S.N. Ghazali et al., "Non-practice of breast self examination and marital status are associated with delayed presentation with breast cancer," *Asian Pacific Journal of Cancer Prevention*, vol. 14, no. 2, pp. 1141-1145, 2013.
- [6] J.P. Kosters and P.C. Gotzsche, "Regular self-examination or clinical examination for early detection of breast cancer," *The cochrane database of systematic reviews*, vol. 1, no. 2, 2003.
- [7] A.B. Miller and C.J. Baines, "The role of clinical breast examination and breast self-examination," *Preventive*, vol. 53, no. 3, pp. 118-120, September 2011.
- [8] E. Mohamaddi Nejad, *Design and development of the computer vision algorithm for a real-time breast self-examination*. Manila: De La Salle University, 2014, Unpublished master's thesis, De La Salle University, Manila.
- [9] IARC Screening Group. (2006, January) Breast digital atlas - Breast self-Examination (BSE). [Online].  
<http://screening.iarc.fr/breastselfexamination.php>



- [10] J. Sariego, "Breast cancer in the young patient," *The American Surgeon*, vol. 76, no. 12, pp. 1397-1400, December 2010.
- [11] J.G. Reeder and V.G. Vogel, "Breast cancer prevention," *Cancer Treatment and Research*, vol. 141, pp. 149-164, 2008.
- [12] J.D. Yager, "Estrogen carcinogenesis in breast cancer," *The New England journal of medicine*, vol. 354, no. 3, pp. 270-282, January 2006.
- [13] E. Santoro, DeSoto M., and J.H. Lee, "Hormone therapy and menopause," *National research center for women and families*, February 2009.
- [14] A. Haim and B.A. Portnov, *Light pollution as a new risk factor for human breast and prostate cancers*, 8th ed.: Springer, 2013.
- [15] The Merck Manual, "Breast Disorders: Breast Cancer," Merck Manual Professional, 2003.
- [16] American Cancer Society, "Cancer facts and figures 2007," American Cancer Society, 2007.
- [17] M. Watson, "Assessment of suspected cancer," *InnovAiT*, vol. 1, no. 2, pp. 94-107, 2008.
- [18] P. Carney et al., "Individual and combined effects of age, breast density, and hormone replacement therapy use on the accuracy of screening mammography," *Annals of Internal Medicine*, vol. 138, no. 3, pp. 168-175, February 2003.
- [19] K. Armstrong, E. Moye, S. Williams, J. Berlin, and E. Reynolds, "Screening mammography in women 40 to 49 years of age: A systematic review for the American College of Physicians," *Annals of Internal Medicine*, vol. 146, no. 7, pp. 516-526, April 2007.
- [20] C.M. Sehgal, S.P. Weinstein, P.H. Arger, and E.F. Conant, "A review of breast ultrasound," *Journal of mammary gland biology and neoplasia*, vol. 11, no. 2,



pp. 113-123, April 2006.

- [21] S. Huber, M. Wagner, M. Medl, and H. Czembirek, "Real-time spatial compound imaging in breast ultrasound," *Ultrasound in medicine & biology*, vol. 28, no. 2, pp. 155-163, February 2002.
- [22] S.G. Orel and M.D. Schnall, "MR imaging of the breast for detection, diagnosis, and staging of breast cancer," *Radiology*, vol. 220, no. 1, pp. 13-30, 2001.
- [23] D. Saslow et al., "Clinical breast examination: practical recommendations for optimizing performance and reporting," *CA: a cancer journal for clinicians*, vol. 54, no. 6, pp. 327-344, November 2004.
- [24] Canadian Breast Cancer Foundation, "Earlier detection and diagnosis of breast cancer: A report from It's About Time! A Consensus Conference," Ontario Region, 2010.
- [25] N. Oestreicher, C.D. Lehman, D.J. Seger, D.S. Buist, and E. White, "The incremental contribution of clinical breast examination to invasive cancer detection in a mammography screening program," *American journal of roentgenology*, vol. 184, no. 2, pp. 428-432, February 2005.
- [26] J.G. Elmore et al., "Ten-year risk of false positive screening mammograms and clinical breast examinations," *The New England journal of medicine*, vol. 338, no. 16, pp. 1089-1096, April 1998.
- [27] N. Baxter, "Preventive health care, 2001 update: Should women be routinely taught breast self-examination to screen for breast cancer?," *Canadian medical association journal*, vol. 164, no. 13, pp. 1837-1846, June 2001.
- [28] Johns Hopkins Medicine. Johns Hopkins Medicine. [Online].  
<http://www.hopkinsmedicine.org/healthlibrary/conditions/>
- [29] Breastcancer.org. The Five Steps of a Breast Self-Exam. [Online].  
[http://www.breastcancer.org/symptoms/testing/types/self\\_exam/bse\\_steps](http://www.breastcancer.org/symptoms/testing/types/self_exam/bse_steps)



- [30] Stony Brook Medicine University Physicians. Breast Care Center - Breast Self Exam. [Online]. <http://www.stonybrookphysicians.com/center/Breast-Care-Center-PI-Self-Exam.asp>
- [31] J. Zeng, Y. Wang, R. Turner, M. Freedman, and S.K. Mun, "Vision-based finger tracking of breast palpation for improving breast-self examination," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Amsterdam, 1996, pp. 148-149.
- [32] J. Zeng, Y. Wang, W. Freedman, and S.K. Mun, "Color-feature-based finger tracking for breast palpation quantification," in *IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, 1997, pp. 2565-2570.
- [33] BIOCORE. BIOCORE. [Online].  
<http://nestor.coventry.ac.uk/biocore/html/projects.htm>
- [34] Y. Hu et al., "A real-time hand tracking and multimedia-based system for breast self-examination: the simplified 3D hand model," in *Proc. IEEE Int. Conf. Info. Tech. App. in Biomed*, Birmingham, 2003, pp. 146-148.
- [35] Y. Hu, R.N.G. Naguib, A.G. Todman, S.A. Amin, and H. Al-Omishy, "HCRA: a hand configuration recognition algorithm for use in breast self-examination," in *Proc. IEEE Int. Conf. Info. Tech. App. in Biomed*, Birmingham, 2003, pp. 334-337.
- [36] Y. Hu, R.N.G. Naguib, A.G. Todman, S.A. Amin, and H. Al-Omishy, "Transitional appearance-based motion tracking for real-time breast self-examination supervision," in *Proc. Int. Conf. IEEE Eng. in Med. and Biol. Soc*, San Francisco, 2004, pp. 3182-3185.
- [37] Y. Hu, R.N.G. Naguib, A.G. Todman, S.A. Amin, and A. Oikonomou, "Hand motion segmentation against skin colour background in breast awareness applications," in *Proc. Int. Conf. IEEE Eng. in Med. and Biol. Soc.*, San Francisco, USA, 2004, pp. 3221-3224.
- [38] Y. Hu, R.N.G. Naguib, A.G. Todman, S.A. Amin, and H. Al-Omishy, "Search



strategies for the automatic delineation of the breast area in a multimedia breast self-examination system," in *Proc. Int. Conf. IEEE Eng. in Med. and Biol. Soc.*, Cancún, Mexico, 2003, pp. 1315-1318.

- [39] H. Mirzaalian, M.R. Ahmadzadeh, and F. Kolahdoozan, "Breast contour detection on digital mammogram," in *Information and Communication Technologies*, vol. 1, Damascus, 2006, pp. 1804-1808.
- [40] M. Masek, Y. Attikiouzel, and C.J.S. de Silva, "Skin-air interface extraction from mammograms using an automatic local thresholding algorithm," in *15th Biennial International Conference BIOSIGNAL*, Brno, Czech Republic, 2000, pp. 204-206.
- [41] S.L. Lou, H.D. Lin, K.P. Lin, and D. Hoogstrate, "Automatic breast region extraction from digital mammograms for PACS and telemammography applications," *Computerized Medical Imaging and Graphics*, vol. 24, pp. 205-220, 2000.
- [42] M.A. Wirth and A. Stapiński, "Segmentatiion of the breast region in mammograms using active contours," *Visual communications and image processing*, vol. 5150, pp. 1995-2006, 2003.
- [43] M. Wirth, D. Nikitenko, and J. Lyon, "Segmentation of the breast region in mammograms using a rule-based fuzzy reasoning algorithm," *ICGST-GVIP*, vol. 5, no. 2, pp. 45-54, January 2005.
- [44] I.K. Maitra, S.K. Bandyopadhyay, S. Nag, and T. Kim, "A novel approach to detect accurate breast boundary in digital mammogram using binary homogeneity enhancement algorithm," in *Int Conf on Ubiquitous Computing and Multimedia Applications*, Daejeon, 2011, pp. 71-75.
- [45] D. Mazumdar, A.K. Talukdar, and K.K. Sarma, "Gloved and free hand tracking based hand gesture recognition," in *1st Int. Conf. Emerging Trends and Applications in Computer Science*, Shillong, 2013, pp. 197-202.
- [46] F.S. Chen, C.M. Fu, and C.L. Huang, "Hand gesture recognition using a real-



time tracking method and hidden Markov models," *Image Vision Computer*, vol. 21, no. 8, pp. 745-758, March 2003.

- [47] Q.Y. Zhang, M.Y. Zhang, and J.Q. Hu, "Hand gesture contour tracking based on skin color probability and state estimation model," *JMM*, vol. 4, no. 6, pp. 349-355, December 2009.
- [48] S.K. Hee, K. Gregorij, and B. Ruzena, "Hand tracking and motion detection from the sequence of stereo color image frames," in *Proc. ICIT 2008*, Chengdu, 2008.
- [49] Z. Pan et al., "A real-time multi-cue hand tracking algorithm based on computer vision," in *IEEE Virtual Reality*, Massachusetts, 2010, pp. 219-222.
- [50] A.K.C. Leung et al., "Internipple distance and internipple index," *Journal of the National Medical Association*, vol. 96, no. 8, pp. 1092-1096, August 2004.
- [51] L.G. Shapiro and G.C. Stockman, *Computer Vision*.: Prentice Hall, 2001.
- [52] D.A. Forsyth and J. Ponce, *Computer Vision, A Modern Approach*.: Prentice Hall, 2003.
- [53] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. (1996, November) A standard default color space for the internet - RGB.
- [54] M. Fairchild, "Color appearance models: CIECAM02 and beyond," in *IS&T/SID 12th Color Imaging Conference*, Scottsdale, AZ, 2004.
- [55] Merriam-Webster. (n.d.) Merriam-Webster Dictionary. [Online].  
<http://www.merriam-webster.com/dictionary/lightness>
- [56] OpenCV Dev Team. (2014, May) Shi-Tomasi Corner Detector & Good Features to Track - OpenCV 3.0.0-dev documentation. [Online].  
[http://docs.opencv.org/ trunk/ doc/ py\\_tutorials/ py\\_feature2d/ py\\_shi\\_tomasi/ py\\_shi\\_tomasi.html](http://docs.opencv.org/ trunk/ doc/ py_tutorials/ py_feature2d/ py_shi_tomasi/ py_shi_tomasi.html)



- [57] H. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Carnegie-Mellon University, Tech Report CMU-RI-TR-3, 1980.
- [58] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147-151.
- [59] OpenCV Dev Team. (2014, April) Canny Edge Detector — OpenCV 2.4.9.0 documentation. [Online]. [http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny\\_detector/canny\\_detector.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html)
- [60] H.G. Barrow and J.M. Tenenbaum, "Interpreting line drawings as three-dimensional surfaces," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 75-116, 1981.
- [61] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.
- [62] P. Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition*, 2001, pp. 511-518.
- [63] Intel IPP. Object Detection Using Haar-like Features. [Online]. [https://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi\\_ch14/ch14\\_object\\_detection\\_using\\_haar\\_like\\_features.html](https://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi_ch14/ch14_object_detection_using_haar_like_features.html)
- [64] OpenCV Dev Team. (2014, May) Optical Flow — OpenCV 3.0.0-dev documentation. [Online]. [http://docs.opencv.org/trunk/doc/py\\_tutorials/py\\_video/py\\_lucas\\_kanade/py\\_lucas\\_kanade.html](http://docs.opencv.org/trunk/doc/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html)
- [65] D.H. Warren and E.R. Strelow, *Electronic spatial sensing for the blind: Contributions from perception.*: Springer, 1985.
- [66] IMAIOS. IMAIOS - Breast cancer. [Online]. <http://www.imaios.com/e-Cases/Channels/Radiology/Radiological-classifications-commonly-used-on-medical-imaging/Breast-cancer>



- [67] World Health Organization. (2010) International Classification of Diseases (ICD). Document.
- [68] A.W. Fitzgibbon and Fisher R.B., "A buyer's guide to conic fitting," in *Proc. 5th British Machine Vision Conf.*, Birmingham, 1995, pp. 513-522.
- [69] A.W.M. Smeulders et al., "Visual tracking: an experimental survey," *IEEE Trans Pattern Analysis and Machine Intelligence*, November 2013.
- [70] B. Babenko, M.H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," *CVPR*, 2009.
- [71] B. Karasulu and S. Korukoglu, "A software for performance evaluation and comparison of people detection and tracking methods in video processing," *MTA*, vol. 55, pp. 677-723, 2011.
- [72] J. Kwon and K.M. Lee, "Tracking of abrupt motion using Wang-Landau Monte Carlo estimation," *ECCV*, 2008.
- [73] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters: Special issue: ROC analysis in pattern recognition*, vol. 27, no. 8, pp. 861-874, June 2006.
- [74] The Linux Information Project. (2005, June) Definition of robust. [Online]. <http://www.linfo.org/robust.html>
- [75] Itseez. OpenCV. [Online]. <http://itseez.com/OpenCV/>
- [76] Qt Project Hosting. Qt Creator 2.8. [Online]. <http://qt-project.org/doc/qtcreator-2.8/>