De La Salle University

Development of Depth Level Estimation Algorithm For Breast Self-Examination

A Thesis
Presented to the Faculty of
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

By
John Anthony C. Jose

Engr. Melvin K. Cabatuan
Thesis Adviser

April 2015

ABSTRACT

It has been proposed in the previous studies of Mohammadi et al[1] and Masilang et al[2] to develop a computer vision-based Breast Self Examination guidance system. It was able to indicate to the user on which area of the breast should be palpated using a simple webcam and desktop computer. However, a comprehensive Breast Self-Examination guidance system requires not just means palpating all areas of the breasts, but also each palpation should suffice a proper pressure level. The goal of this thesis is to develop a depth level estimation algorithm that can recognize whether the palpation is low, medium or deep palpation using only a simple RGB camera and a desktop computer.

To do this, we gathered a RGB and Depth Images of subjects, with varying cup size, performing BSE using Kinect for Xbox 360. A medical expert supervised the recording of the said dataset. Also, this study introduces an evaluation scheme for quantifying low, medium and deep pressure level coming from the said dataset using a “Fuzzy-like” membership Relation. The said evaluation scheme was also utilized to have a quantitative accuracy for the previous studies of depth estimation for BSE.

For estimating the depth using simple camera, this study proposes the use of Local Binary Pattern Global Histogram Features and 9 Laws’ Textures Histogram as the feature extraction scheme. These features shall be the input to Support Vector Machine. Our depth level estimation algorithm was able to classify depth levels with an overall test accuracy of 77.71%. It provides a 250% higher accuracy than the state-of-the-art.

De La Salle University

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Engr. Melvin Cabatuan, and Dr. Dadios for providing me an opportunity to work on this wonderful research. I am deeply honored for choosing me to be part of the research group.

I wish to thank my friends and volunteers, whom I cannot name for anonymity, for participating in the data gathering. I also wish to express my utmost gratitude for Dr. Reynaldo Joson who agreed to help this research without any compensation. I am also thankful for Florence Culaba and Jennielyn Diagbel who helped me in finding participants, and Nicole Pineda who helped me record the videos using her own DLSR camera during the data gathering phase. Without them, I don't think I would be able to have the RGBD BSE dataset I need for this research.

I would also like to thank Engineering Research and Development for Technology (ERDT) and Philippines Commission on Higher Education Research Network (CHED-PHERNET) Program for funding the pieces of equipment, materials, and data gathering expenses I need for making this study possible.

I would like to thank again my advisor, Engr. Melvin Cabatuan, my thesis panels, Engr. Edwin Sybingco, Dr. Elmer Dadios, and Dr. Laurence Gan Lim for their ideas, comments, and suggestions to improve the substance, grammar, and writing style of this paper.

Lastly, I would like to thank God for providing the strength, will and determination to finish this research. My utmost gratitude also to my family and friends who provided their moral support, encouraged me, and gave me advices to do what I must do especially to finish this thesis.

CONTENTS

ABSTRACT ii

ACKNOWLEDGEMENTS iii

CONTENTS iv

LIST OF FIGURES viii

LIST OF TABLES xi

CHAPTER 1 Introduction 1

1.1 Background of the Study 1

1.2 Statement of the Problem 3

1.3 Significance of the Study 4

1.4 Objective of the Study 4

1.4.1 General Objective 4

1.4.2 Specific Objective 5

1.5 Scope and Limitation 5

CHAPTER 2 Review of Related Literature 6

2.1 Breast Cancer 6

2.1.1 What is Breast Cancer 6

2.1.2 Methods for Prevention 8

2.2 Self-Breast Examination as a method for Prevention 8

2.3 Vision-Based Guidance System as a supplemental tool 10

2.4 Depth Cues 11

2.4.1 Monocular Depth Cues used by the Human Eye 11

2.4.2 Monocular Depth Cues used in Computer Vision 12

2.5 Depth Estimation for monocular image 13

2.6 Depth Estimation in image sequence 14

2.7 Depth Estimation for BSE 16

2.8 Breast Classification 17

2.9 Research Gap **18**

CHAPTER 3 Theoretical Framework **19**

3.1 Image and Video Representation **19**

3.2 Optical Flow **19**

3.2.1 Optical Flow Field 23

3.3 Stereo Depth vs. Monocular Depth **23**

3.4 Depth Map **24**

3.5 RGBD Images **26**

3.6 Linear Programming **28**

3.6.1 L1 Norm Minimization 28

3.6.2 L1 Norm minimization for Learning Parameters 29

3.7 Image Color Space **30**

3.7.1 Grayscale Color Space 30

3.7.2 YCbCr Color Space 31

3.7.3 C1C2C3 Color Space 31

3.8 Textures Features **32**

3.8.1 Laws' Texture Filters 32

3.8.2 Local Binary Pattern (LBP) 35

3.9 Machine Learning Statistical Models **37**

3.9.1 Support Vector Machine (SVM) 37

3.9.2 Gradient Boosted Trees 39

3.10 Breast Cup Size **41**

3.10.1 Breast Modeling 43

CHAPTER 4 Methodology **44**

4.1 Design **44**

4.2 Assumptions of the System **45**

4.3 Hardware Considerations **45**

4.4 Datasets **46**

4.4.1 Gathering of New Datasets 46

4.4.2 Methodology 47

4.4.3 Camera Setup 48

4.4.4 RGB and Depth Map Extraction 48
4.4.5 Box Mask Sequence 50
4.4.6 Creating training and testing images 54
4.4.7 Finger Mask Sequence 55
4.4.8 Creating Ground truth 56
4.5 Quantification of Depth Level 57
4.6 Feature Extractions Schemes 59
4.6.1 Normalized Shadow Area 59
4.6.2 Image Entropy 61
4.6.3 Laws' Textures Histogram 61
4.6.4 Local Binary Pattern Global Histogram 64
4.7 Machine Learning Algorithms 65
4.7.1 Linear Regression 65
4.7.2 Support Vector Machine 65
4.7.3 Gradient Boosted Trees (GBT) 69
4.7.4 Artificial Neural Network 70
4.8 Accuracy Assessment 73
CHAPTER 5 Results 76
5.1 Gathering of Database 76
5.1.1 Kinect Database 76
5.1.2 Segmentation of each cup size to the proper quadrant 77
5.1.3 Box Mask Sequence 78
5.1.4 Finger Mask Sequence 79
5.1.5 Quantification of Low, Medium and Deep 80
5.2 Quantitative accuracy of Chen et al's Pressure Estimation Algorithm 82
5.3 Normalized Shadow Area 84
5.4 Laws' Features Global Histogram Optimization 85
5.5 Local Binary Pattern Global Histogram (LBPGH) Optimization 89
5.6 Model Selection 90
5.7 Selection of optimal Feature Combinations 92
5.7.1 Accuracy Assessment of Each Features 93

5.7.2 Accuracy Assessment for all combinations of Features 97

CHAPTER 6 Discussion And Analysis 100

6.1 RGBD BSE dataset 100

6.2 Quantitative Evaluation 101

6.3 Benchmarking with state-of-the-art 102

6.4 Depth Classifier 103

6.4.1 Prediction model selection and analysis 103

6.4.2 Features selection and analysis 106

6.4.3 Overall analysis 111

CHAPTER 7 Conclusion and Recommendation 113

7.1 Summary 113

7.2 Future Work and Recommendations 114

Appendix 117

References 134

LIST OF FIGURES

FIGURE 2-1. PARTS OF BREAST [3] 7
FIGURE 2-2. SAXENA'S MAKE3D: (A) RAW IMAGE, (B) GROUND TRUTH, (C) PREDICTED DEPTH [20]....14
FIGURE 2-3. GOLF STREAM IN FACTORIZATION METHOD: (A) ORIGINAL IMAGE, (B) RESULT [49] 15
FIGURE 2-4. FLOWER GARDEN SEQUENCE: (A) ORIGINAL IMAGE, (B) ESTIMATED DEPTH MAP [39] 15
FIGURE 2-6. CHEN ET AL'S ALGORITHM: (A) EXPERIMENTAL SETUP, (B) RESULTING GRAPH[19] 16
FIGURE 2-7. ARTIFICIAL NEURAL NETWORK (ANN) PROPOSED METHOD [18] 17
FIGURE 3-1. VECTOR FIELDS IN AN IMAGE [58]....20
FIGURE 3-2. A PIXEL IN THE SAME LOCATION (A) FRAME A, (B) FRAME B [59] 21
FIGURE 3-3. SAMPLES OF OPTICAL FLOW FIELD: (A) ORIGINAL IMAGE, (B) LEFT, (C) RIGHT, (D) UP, (E) DOWN, (F) ZOOM IN, (G) ZOOM OUT 22
FIGURE 3-4. OPENCV EXAMPLE FOR CREATING A DISPARITY MAP [65] 24
FIGURE 3-5. BOWLING BALLS AT VARIOUS DEPTH ORDER [30] 25
FIGURE 3-6:(A) GIRL IN ELEVATOR, (B) ITS DEPTH MAP, (C) MAN ON HALLWAY, (D) ITS DEPTH MAP [46] 25
FIGURE 3-7. CAMERA PROJECTIONS OF REAL WORLD POINTS (SOURCE: [66]) 27
FIGURE 3-8. CAMERA COMPONENTS OF KINECT FOR XBOX 360 (SOURCE: [68]) 28
FIGURE 3-9. LOCAL BINARY PATTERN 3X3 WINDOW (SOURCE: [72]) 36
FIGURE 3-10. A CLOSE-UP IMAGE OF A GIRL: (A) GRAYSCALE IMAGE, (B) LBP TEXTURE MAP 37
FIGURE 3-11. SVM OPTIMAL HYPERPLANE (SOURCE: [74]) 38
FIGURE 3-12. GRADIENT BOOSTED TREES OPTIMIZATION PSEUDO CODE 41
FIGURE 3-13. BREAST CUP SAMPLES: (A) CUP SIZE A [79], (B) CUP SIZE B [80], (C) CUP SIZE C [81], (D) CUP SIZE D [82] 42
FIGURE 4-1. GENERAL EXPERIMENTAL SETUP 44
FIGURE 4-2. SAMPLE IMAGES: (A) DATABASE 1, (B) DATABASE 2, (C) DATABASE 3 46
FIGURE 4-3. CAMERA COMPONENTS OF KINECT FOR XBOX 360 (SOURCE: [68]) 49
FIGURE 4-4. KINECT RGB AND DEPTH EXTRACTION FLOW CHART 51
FIGURE 4-5. SAMPLE BOX MASK SEQUENCE 53
FIGURE 4-6. THE FOUR STEPS IN CREATING TRAINING AND TESTING IMAGES 54

FIGURE 4-7. QUADRANTS OF THE BREAST 55
FIGURE 4-8. CALCULATION OF NONZERO MEDIAN PSEUDO CODE 57
FIGURE 4-9. MODEL USED FOR DEPTH QUANTIZATION 58
FIGURE 4-10. SHADOW SEGMENTATION FLOWCHART 60
FIGURE 4-11. IMAGE ENTROPY FEATURE EXTRACTION FLOW CHART 61
FIGURE 4-12. LAWS' FEATURES EXTRACTION 63
FIGURE 4-13. LBP HISTOGRAM FEATURE EXTRACTION SCHEME.....64
FIGURE 4-14. SVM TRAINING SCHEME 67
FIGURE 4-15. SAMPLE IMAGE AND DEPTH LEVEL EXTRACTION 68
FIGURE 4-16. SVM SAMPLE INPUT 69
FIGURE 4-17. GRADIENT BOOSTED TREES TRAINING SCHEME 71
FIGURE 4-18. ARTIFICIAL NEURAL NETWORK (ANN) TRAINING SCHEME.....72
FIGURE 4-19. PARTITIONS OF THE DATASET 73
FIGURE 4-20. SAMPLE CONFUSION MATRIX 74
FIGURE 5-1. KINECT DATABASE 77
FIGURE 5-2. BOX MASK SEQUENCE FOR QUADRANT 2 CUP B 78
FIGURE 5-3. OCCLUDED FINGERS IN QUADRANT 5 CUP B 79
FIGURE 5-4. FINGER MASK SEQUENCE IN CUP A Q2 80
FIGURE 5-5. FUZZY MODEL 81
FIGURE 5-6. CHEN ET AL 'S QUANTITATIVE ACCURACY.....84
FIGURE 5-7. AVERAGE ACCURACY OF SHADOW AREA, ENTROPY AND BASELINE.....85
FIGURE 5-8. PARAMETER SWEEP GRAPH FOR NUMBER OF BINS OF LAWS' HISTOGRAM 86
FIGURE 5-9. BEST 15 NUMBER OF BINS FOR EACH PRE-PROCESSOR 88
FIGURE 5-10. PARAMETER SWEEP FOR LBPGH 90
FIGURE 5-11. MODEL SELECTION AVERAGE ACCURACY ASSESSMENT 92
FIGURE 5-12. FEATURE SELECTION AVERAGE ACCURACY 95
FIGURE 5-13. FEATURE COMBINATION AVERAGE ACCURACY 98
FIGURE 6-1. MODEL USED FOR DEPTH QUANTIZATION 102
FIGURE 6-2. CHEN ET AL'S ALGORITHM 103
FIGURE 6-3. PREDICTION MODEL SELECTION AVERAGE ACCURACY 105
FIGURE 6-4. COMPARISON OF ACCURACY BETWEEN DIFFERENT REGRESSION SCHEMES 106
FIGURE 6-5. TRAINING AND TESTING ACCURACY OF EACH FEATURES 107
FIGURE 6-6. FEATURE COMBINATION AVERAGE ACCURACY 108
FIGURE 6-7. COMPARISON OF DIFFERENT COMBINATIONS WITH LAWS' FEATURES ...109

FIGURE 6-8. COMPARISON OF DIFFERENT COMBINATIONS WITH LBP FEATURES ONLY 110
FIGURE 6-9. COMPARISON WITH DIFFERENT COMBINATIONS FOR LAWLBP 111

LIST OF TABLES

TABLE 3-1. LAWS' KERNEL PART 1 33
TABLE 3-2. LAWS' KERNEL PART 2 34
TABLE 3-3. STATISTICAL TEXTURE MEASURES 35
TABLE 3-4. SIZE CHART [83] 43
TABLE 4-1. GRADIENT BOOSTED TREES PARAMETERS 70
TABLE 4-2. ANN PARAMETERS 73
TABLE 5-1. NUMBER OF TRAINING AND TEST IMAGES FOR EACH QUADRANT 78
TABLE 5-2. BOX MASK SEQUENCE RESOLUTIONS 79
TABLE 5-3. EXTENDED FUZZY PARAMETERS 81
TABLE 5-4. RANGE OF LOW, MEDIUM AND HIGH PRESSURE LEVEL 82
TABLE 5-5. ENTROPY TRAINING AND TEST ACCURACY 83
TABLE 5-6. TRAINING AND TEST ACCURACY OF SHADOW AREA 84
TABLE 5-7. TOP 15 NUMBERS OF BINS 87
TABLE 5-8. TOP VALUES OF WINDOW SIZE 89
TABLE 5-9. MODEL SELECTION ACCURACY ASSESSMENT 91
TABLE 5-10. FEATURE SELECTION ACCURACY 94
TABLE 5-11. CONFUSION MATRIX (A) LBP, (B) LAW, (C) ENTROPY, (D) SHADOW 96
TABLE 5-12. CONFUSION MATRIX: (A) LAWLBP, (B) ENTLAWLBP, (C) SHALAWLBP 99
TABLE 6-1. BOX MASK RESOLUTION 101
TABLE 6-2. FUZZY PARAMETERS 101

CHAPTER 1
Introduction

1.1 Background of the Study

Breast cancer is one of the malignant tumors that start around the tissues of the breast. It often detected on women, however man can also obtain it [3]. The United States Breast Cancer Statistics says that about 1 in 8 U.S. women (just under 12%) will develop invasive breast cancer over the course of her lifetime. About 39,520 women in the U.S. were expected to die in 2011 from breast cancer, though death rates have been decreasing since 1990 — especially in women under 50 years old. These decreases are thought to be the result of treatment advances, earlier detection through screening, and increased awareness [4]. The rates of breast cancer patients have continuously increased for more than 2 decades. 69% of all breast cancer deaths occur in developing countries according to WHO Global Burden of Disease 2004 [5].

Survival rates vary worldwide. Sometimes it also depends on the type of country. Developed countries reach a survival rate of around 80% in North America and 60% in Japan while the less developed countries only reach below 40%. This low survival rate is mainly caused of lack of early detection and lack of facilities [5]. Anderson et al points out that in a limited-resource country, one of the main obstacles is lack of breast cancer awareness, creating health cares and insufficient funds [6].

There are three popular early detection methods for Breast Cancer: Mammogram, Clinical Breast Exam (CBE) and Breast Self-Examination (BSE). Among the three methods, mammogram is the most recommended early detection [3]. It is simply performing an X-ray scan to your breast. It can detect cancer lumps 2 years before the signs and symptoms shows up [7]. In clinical

breast exam, a physician will inspect the breast and its surrounding area to check for any signs of breast cancer by finger palpation with different pressure levels to check for lumps. BSE is similar to CBE except that it is the woman who will inspect her breast. Mammogram and CBE have been the popular methods for early detection practices. However, Mammogram and CBE will be very costly for developing countries. In the Philippines, digital mammogram costs around 2000-2800 pesos in St. Luke Hospital [8]. This is costly for a country only having a GDP per capita of 66000 pesos [9]. Another limitation of the sole use of Mammogram is that it cannot detect all lumps. It has an accuracy of around 88.5%. Some lumps are detected with the use of CBE but not detected by the mammogram. When both methods are used on a patient, the efficacy of detecting lumps is around 94.5% [10]. That's why, CBE is considered as the supplement of mammogram. There was a study conducted whether CBE can be used as a sole prevention method as it was found that there were similar mortality rate between the CBE-only group and the CBE-and-mammography group [11]. That's why, CBE has been argued as the cost-effective early detection approach especially for limited-resource country [12]. However, in the Philippines, cultural and logistic barriers have impeded this approach. It has been found that the test sensitivity with annual repetition is 53.2% [12]. Meaning, about 53.2% of the women would have second thoughts for having annual CBE. Filipino women are not culturally prepared for doctors invading their breast.

In the Breast Health Global Initiative 2005 Guidelines, for a limited resource country, one of their recommendations for early detection for the basic level is to promote breast awareness and breast self-examination [6]. Parvani claims that BSE seems to be the realistic approach for developing countries [11]. But, some medical researchers have disputes regarding its effectiveness [13]. A Shanghai-based research team created a large-scale randomized trial

for 266,064 Chinese women. After five years of monitoring, there was about 135 (0.1%) breast cancer deaths in the BSE-instructed group while 131 (0.1%) breast cancer deaths in the control group which suggests that BSE does not reduce mortality rate of breast cancer [14]. However, in a recent study under a Singapore-based research group, the breast self-examination has been founded effective in detecting breast lesions in Chinese female patients [15]. A breast lesion is an abnormal change in breast tissue. It can be either cancerous or benign [16]. The study of [15] showed that BSE users which have regular follow-up with physicians has an accuracy of 78% for detecting breast lesions. Hence, the probable cause of low efficacy might be that many women are not well-trained in doing the proper method [17]. A possible solution is to use computer vision-based Breast self-examination guidance system. It can help dictate whether the hand is on its proper placement, the pressure is sufficient enough to determine if the breast have lumps at that level [18].

1.2 Statement of the Problem

One of the key features in creating a Breast Self-Examination Guidance System is determining the pressure level in each palpation. An actual BSE requires low, medium and high palpation level in each stroke as some lesion only exist at a certain level [3]. In other words, a lesion at a medium level can be felt only at a medium pressure and not at light and heavy palpation.

Chen et al studied the pressure estimation in BSE [19]. However, their algorithm is not realistic enough for a true BSE performance since their depth estimation only assumed that the finger movement is either palpating or not. It has not included the natural lateral and rotational movement in a palpation process.

In order to make the BSE guidance system more realistic, a single regular camera similar to the one seen in an android tablet which are called

monocular camera. But one of its main difficulties is the 2D to 3D conversion. The system should be able to acquire depth in 2D image sequence. Moreover, the present literature only have algorithm for depth estimation from monocular image for outdoor and indoor images. The literature had not tackled on close-up images and small variations in depth like in BSE. The present state-of-the-art algorithm for estimating depth of outdoor images is able to correctly classify 64.9% of the images only [20]. Since the structure of the algorithm depends on the context of outdoor images. For instance, whenever there is a blue color in the upper region of image, it has more probability that the blue is a sea, hence farther depth. If there is green in the lower region, it is more likely grass, hence nearer depth. These assumptions are not applicable to close-up indoor images like in BSE. Therefore, there should be a new algorithm that can address the issue of these types of images. However, as stated earlier, no literature is presently available.

1.3 Significance of the Study

First, this study will address the depth estimation for monocular close-up image that is not available in the literature. This estimation can be useful not just in estimating depth for BSE but also in other application such as dress fitting using computer vision, hand gestures, sign language recognition, etc.

This study is also significant as it addresses a solution for estimating slightly varying depth in an image sequence that changes only as small as 5-10mm. This study requires more dense depth map estimation compared to what is seen in the literature.

1.4 Objective of the Study

1.4.1 General Objective

- To develop a computer vision-based algorithm that can recognize high, medium and low pressure level palpation in a Breast Self-Examination

1.4.2 Specific Objective

- To design a fuzzy-based depth model for cup sizes A, B, C that will be used for depth estimation
- To implement a computationally intelligent algorithm that utilizes depth cues to aid in increasing the accuracy of depth estimation
- To perform a comparison with a state-of-the-art techniques benchmarked with depth camera
- To create a new RGB-D database showing the different palpation level

1.5 Scope and Limitation

This study will focus on estimating depth pressure for low, medium and high-pressure level only. In other words, the pressure level output will either be low, medium high or none. It cannot estimate the absolute depth of palpation. The study will assume that it can obtain a small region of interest (ROI) segmented by other research. This region contains the fingers palpating a small breast area. The fingernail should be visible enough in the image sequence.

As this research is still new to its field, its focus will be creating a working algorithm for depth estimation and interpretation. The study will not address its real-time implementation. Optimization of the codes will be done in the succeeding research.

CHAPTER 2
Review of Related Literature

2.1 Breast Cancer

2.1.1 What is Breast Cancer

Breast cancer disease occurs when malignant cancer cells started to grow tumors in the breast area. After starting at the breast, cancer cells will typically scatter around the body. Most of the breast cancer patients are women. However, it can also occur at men [3].

To understand the cause of breast cancer, it is advisable to understand first the anatomy of breast as seen in Figure 2-1. The main components of breast are lobules, ducts and fatty connective tissue. Lobules are the glands responsible for producing milk. Ducts are the passageways of the milk from the lobule to the nipple. Fatty connective tissue surrounds the lobules and ducts. It is often called stroma.

Most breast cancer starts often ducts or lobules. A small percentage starts at other tissues. Oftentimes, many women may find lumps in their breast. However, it turns out that most of the lumps are not malignant [21]. These are called benign lumps. Lumps are often formed due fibrosis. It is a process where there is a change in breast tissue. All women will encounter it sometime in their life. However many physician still recommends women to consult a specialist if they find lumps in their breast.

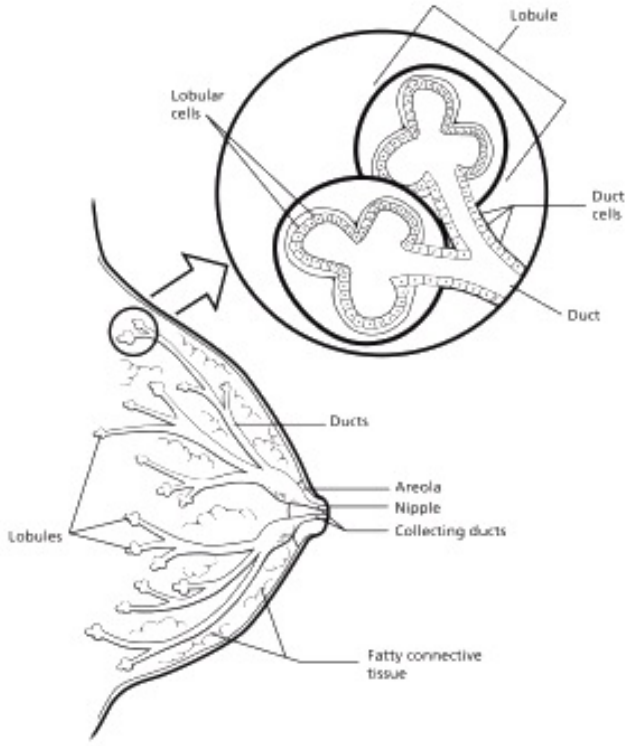


Figure 2-1. Parts of Breast [3]

The primary risk factor for having breast cancer is gender. Females have 100 folds larger risk than men [4]. Another Risk factor is age. About 77% of the women breast cancer patients are 50 years old and above [7]. Moreover, women who started menstruating before 12 years old have greater risk [3]. If a woman had a history of breast cancer, she also has a larger risk of developing breast cancer again. Recent studies have shown that diets and lifestyle attribute also to the risk of developing breast cancer. High fat diet, obesity, tobacco use causes a larger risk. Women who drink one alcoholic beverage a day only have a slight increase in risk. However, a woman who drinks three bottles a day has twice the risk of developing cancer. Studies have shown that there is a positive correlation between alcoholic beverage and estrogen secretion [21]. Women who had no born any children or gave birth after 30 years old have larger risk factor

2.1.2 Methods for Prevention

Physicians would wish to have early detection so that it can save many lives. Having early detection tests can be used when they grow older so that they can use this test for higher rate of breast cancer detection. In [3], it proposed a guideline for early detection scheme. They had categorized it depending on the women's age:

- Women 40 years old and above – they are advised to take an annual mammogram. Presently, mammogram is the most effective method for detecting breast cancer cells.
- Women in 30's – they are advised to take clinical breast exam (CBE) every 3 years. CBE gives an opportunity for women and doctor to take about the status of their breast and discuss whether it had changes or not. The doctors will also discuss the lifestyle of the woman, which may increase her risk factor. It provides a venue for women to get to know their breast better. They will be more sensitive in matters of their breast. So that when they are at 40's they can tell their doctors immediately if they had feel any new symptoms are changes in breast, swelling, retraction, swelling, etc.
- Women in 20's –They are recommended to have Clinical breast exam or Breast Self-examination. Breast self-examination is not as effective as mammogram. However, doing breast self-exam provides more awareness for their breast. It had been shown that women who chose to do regular breast self-examination could readily more detect any signs and symptoms if any changes in breast occur.

2.2 Self-Breast Examination as a method for Prevention

Overall, physicians recommend the use of mammogram for breast cancer early detection. Mammogram can detect 80-90% of the lumps not felts by the woman [21]. It is most trusted early detection tool for breast cancer.

However, one main problem of mammogram applied in the context of developing countries is that they are expensive. Digital mammogram costs around 2000-2800 pesos in St. Luke [8]. This will cost a lot for a country having a GDP per capita of only 1501.83 dollar or equivalent to 66000 pesos [9]. Mammograms are not affordable enough for developing countries like Philippines.

Clinical Breast Exam is a method wherein the breast examination will be conducted by an attending physician, nurse or any medical practitioner [7]. CBE are a supplemental tool for mammography. It can detect lumps that are not detected by the machine. Studies have even shown that doing the lump detection rate is similar for woman who had combined mammography and CBE and CBE alone [11].

The woman will first be advised to undress their top. The first step is to do inspection. The medical practitioner will check the symmetry of the breast, abnormalities in size and shape, nipple retraction, swelling, etc. Then, they will now palpate the prone area around the breast. They will check if there are swelling lymph nodes, inspect if there are lumps detected. There are three methods: vertical slice, radial, concentric. The vertical slicing means that finger shall run a path from top to bottom starting from the leftmost part of the breast until it covers the whole breast. The radial method is when the palpation starts from nipple and gently goes down to the chest area. The finger shall be rotating clockwise until it covers the whole breast. The concentric method is when the finger palpation starts from the nipple and it will circularly go spirally until it covers the whole breast [22]

Breast Self-Examination method is similar to CBE method. The difference is that the woman herself will perform BSE. It has been studied that this method has been more appealing to woman. It boosts their confidence and

awareness in their breast and their fight against breast cancer. As CBE is argued to be a cost-effective choice for developing countries, the culture of the Philippines prevents them to actually implement and promote CBE [12].

2.3 Vision-Based Guidance System as a supplemental tool

BIOCORE presented novel researches about Breast self-examination training through the use of multimedia system. The research team is focused on improving the BSE training relative to the leaflets and videos currently use as channels training. Multimedia incorporates the elements of audio, image, animation, and texts. It has been that the use of multimedia will create better venue for training as it has multiple elements. They proposed a system wherein the user can determine their BSE performance by recording themselves. Afterwards, the system will present a video playback wherein it will highlight the mistakes they can improve on [23] . They improved on this system by incorporating the use of Augmented Reality called interactive reality system (IRiS). They wish to create the multimedia system similar QuickTime VR player wherein the user can view their performance available in all angles. It would use 6 video cameras recording at the same time to create the virtual reality effect [24]. However, it has not presented any real-time multimedia system feedback for the user while doing the breast self-examination in a ubiquitous platform.

The research team had progresses in relation to the hand tracking and breast delineation. Hu et al [25] proposed implementing a delineation of the breast area that should be palpated by the user for the vision-based breast self-examination guidance system. This research can be helpful in enhancing the image more by removing more noise from feature extraction. There are two proposed algorithm namely: BADA1 and BADA2. The performance of BADA1 is considerably fair in terms of speed. It produces the output image within 5 seconds. However, the algorithm is only good for typical breast. If the color of the nipple is similar to the color of the breast like in an aged woman, the

algorithm might have some difficulty. On the other hand, BADA2 is more robust in detecting the proper area. It employed Gaussian Pyramiding and Hough Transform. It overcomes the difficulty shown by BADA1. However, it can only produce an output every 30 seconds.

Hu et al [26] created a hand motion segmentation model for breast self-examination multimedia system. The purpose of this model was for studying hand recognition and motion. Since it is a model, it cannot be used for real use yet as it has not incorporated the varying light intensity. However, the purpose is clear; it is for studying the hand motion while performing breast self-examination training. The hand was clearly segmented from the breast area. It used Y-Cb-Cr video format. It initially segments the breast area as baseline first using Gaussian mixture model. Afterwards, it was just segmented using simple refined color difference method.

2.4 Depth Cues

Human eyes use multiple cues for identifying depth. Because of the rise of 3D binoculars usage in movie industry, many thinks that we need the two eyes to perceive depth [27]. But in [27], it states we use two types of cues for perceiving depth: stereo and monocular cues. It has been perceived that stereo cues is important however, monocular cues are also equally important [28]. Psychologists have shown without stereo cues, depth perception can still be known qualitatively. In other words, a stand-alone depth estimation using only monocular cues is possible. A single eye can qualitatively perceive depth even with the use of one eye [28]. Stereo cues only helps in estimating depth more accurately within 4m distances.

2.4.1 Monocular Depth Cues used by the Human Eye

In the field of vision research, several monocular cues have been determined and each of which have been exploited in different applications:

1. Occlusion – this cues indicates that whenever an object is in front of another objects such the object from behind is partially seen and partially unseen (i.e. partially occluded), the object in front is perceived as nearer to the observer. These cue has been exploited in computer vision by [29]–[32]
2. Texture – this cue tells that by looking at the texture of an image, it can help identify the perceived depth of a monocular image. It is considered farther when the features are smaller and features become denser whenever it is farther. In the field of computer vision, [20], [33]–[35] exploited the use of texture features to add more robust algorithm for constructing depth map of outdoor images.
3. Linear perspective – this cues tells that a farther objects tends to be smaller compared if it is nearer to the observer. Moreover, as scene goes farther, objects tends to converge to a point. In computer vision, [36], [37] used linear perspective by identifying the vanishing point to estimate depth in real-time. In another research, the depth was estimated by acquiring size of familiar objects in an images and comparing it to its actual size.[38]
4. Structure from Motion – This cue tells that a depth can be estimated by analyzing the motion of the scene. Whenever an object moves and the observer is stationary, the perceived velocity of the observer will depend upon the depth of each features of the moving object. This depth cue have been integrated with an optimization process to compute depth [39], [40]. This was also exploited together with other cues to make depth estimation more robust [41] .

2.4.2 Monocular Depth Cues used in Computer Vision

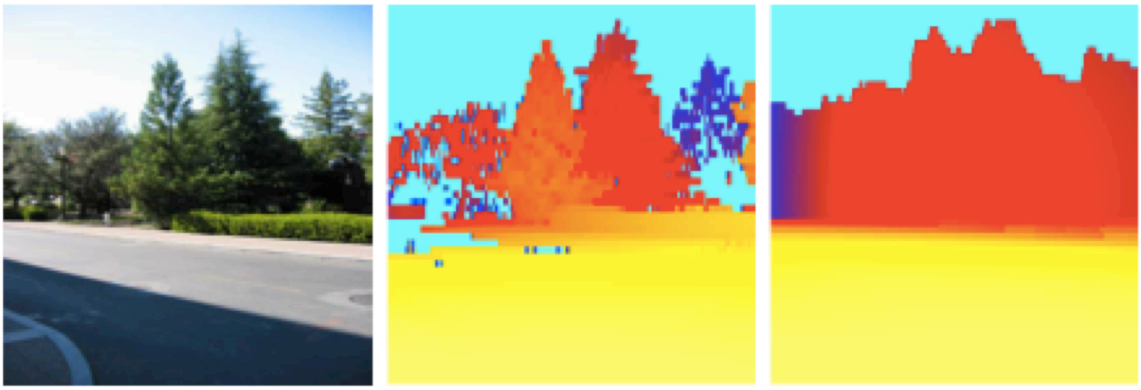
In the current research of depth from monocular images, depth cues used by the human eye have been extensively utilized in order to create a

realistic machine vision applications for robotics [42], 3D reconstruction [34], obstacle avoidance [37],etc.

Other researchers created their own method without relying on what features/methods used by the human eye. One proposed method is to use an optimization problem having Minimum Description Length (MDL) as the fitness function [39], [40]. A research work in MIT have proposed the obtaining the depth of ratio and proportion of a familiar standard structure [38], [43] using feature description of GIST [44]. For example, when CRT computer monitor is seen on the image. It will find the area covered by CRT monitor in the image. Since the monitor has a standard size, it can calculate the ratio of the actual area to the standard area. From, it can infer its absolute depth from the camera.

2.5 Depth Estimation for monocular image

Although monocular depth cues used by human eye seems to be the most logical method to infer depth, many researchers have utilized other means discovered in the computer vision field. One of them is to use Markov Random Field as a model for estimating depth. It is a supervised network wherein training image are 3D images of various images. Afterwards, each region of the input test image will estimate depth based on the. Training bias[20]. These had been improved by adding semantic labeling in the network [45]. Another research group used non-parametric sampling to infer depth. Essentially, each features of the test image will be compared to a database of the features of RGBD images. After getting the nearest match, it will use the nearest match's depth to obtain the desired pixel's depth. [46]. A sample results for Saxena's algorithm is shown in Figure 2-2.



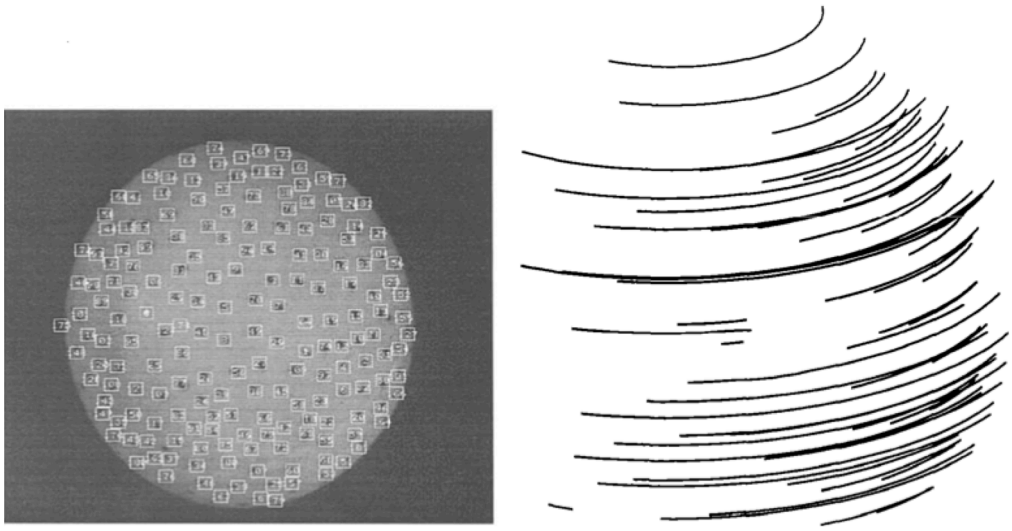
(a) (b) (c)

Figure 2-2. Saxena's Make3D: (a) raw image, (b) Ground truth, (c) Predicted Depth [20]

Other notable algorithms are the work of [47] It used Bayesian Network model to estimate depth. In another research, conditional random field is used as the probabilistic model for depth estimation [48].

2.6 Depth Estimation in image sequence

One of the advances done in the field of computer vision is to reconstruct depth map from an image sequence. As long as the image sequence consists of non-planar point, one can estimate its depth map. Tomasi and Kanade introduced factorization method [49]. It was able to reconstruct a 3D image of a video consisting of spinning golf ball as shown in Figure 2-3. This method was extended to incorporate flexible models [50], But the disadvantage of this method is that it requires full set of features from the previous image to the succeeding image. Meaning, It cannot lose the features tracked from the previous image.



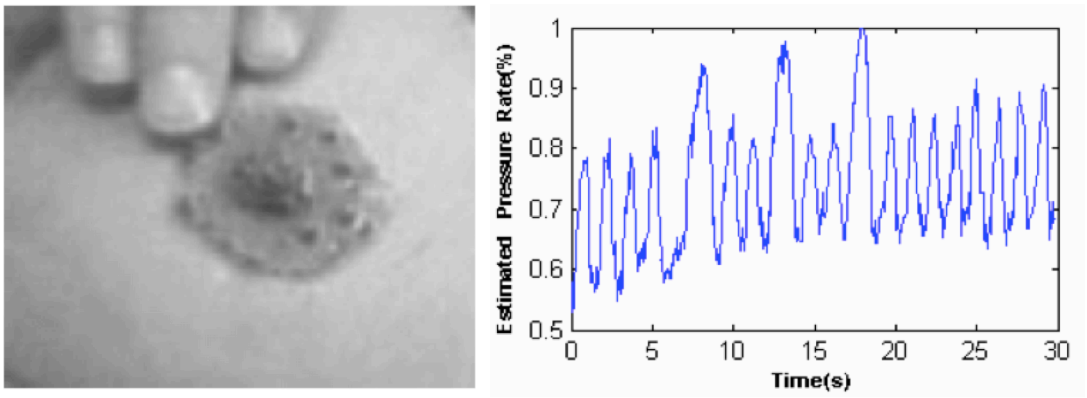
(a) (b)

Figure 2-3. Golf Stream in Factorization Method: (a) original image, (b) result [49]

The work of Mitiche and Hadjres formulated depth estimation as an optimization problem [39]. It used minimum description length as its objective function. They were successful to create a considerable depth map in butterfly sequence, soft drinks sequence, and flower garden sequence. However its main disadvantage is its complexity of the algorithm where it is estimated to be in cubic. A sample result is shown in Figure 2-4.

(a) (b)

Figure 2-4. Flower Garden Sequence: (a) Original Image, (b) Estimated Depth Map [39]



(a) (b)

Figure 2-6. Chen et al's Algorithm: (a) Experimental Setup, (b) Resulting Graph[19]

Recently, [51] created dense 3D reconstruction from image sequence. It used the Light fields as its feature, which they claim as good features for high-resolution images.

2.7 Depth Estimation for BSE

In the present literature, only two researches are available for estimating depth for breast self-examination. The Research Group in Coventry was able to develop a hand pressure estimation using entropy. Its input image are hard-coded video of hand palpation in one fixed location as shown in Figure 2-5(a). When the user palpates the breast, it will track the difference of the initial image and final image using entropy. The entropy measure how disperse the information is. This paper proposes a linear model between pressure and entropy [19]. The obtained graph for a 30 second frame is shown in Figure 2-5(b). Their results have not indicated a numerical accuracy. The disadvantage of using entropy method is that it had assumed that palpation of the user will only consists of up and downward movement. It had not factored in the natural translational and rotation movement of finger during palpation. It can be shown that this algorithm can produce high value (i.e. representing deep palpation) even when the finger consists only of lateral movement. It is still not applicable for practical use.

The second research used the neural network to estimate depth in BSE. It was trained using a monocular image sequence from a BSE video downloaded from the Internet. The setup of the woman is standing up at about 370-410mm distance from the camera. A sample output is shown in Figure 2-7. The final output classifies only of low, medium and high class, which is among the proposed output to this study. It was reported to achieve an accuracy of 87.5% [18]. However, the main disadvantage of their algorithm is in relation with the training images: they extracted training, validation and testing images from a single video. Hence, their scope is not generalized for all women.

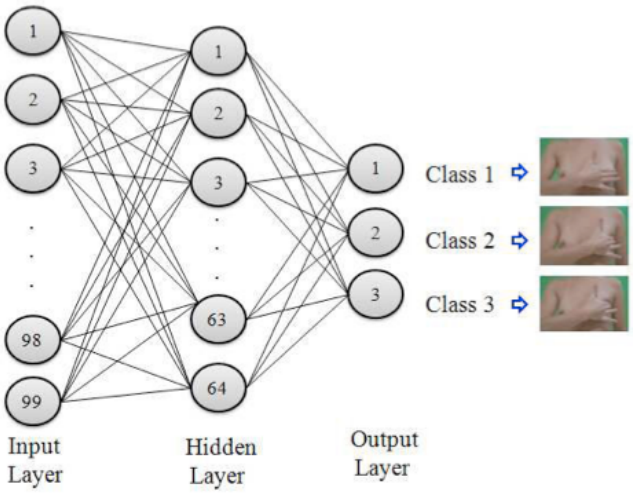


Figure 2-7. Artificial Neural Network (ANN) proposed method [18]

2.8 Breast Classification

The idea of breast classification can be attributed to the work of Nedovic et al [52]. This study stated that due to complexity of various indoor scenarios, he proposed to classify first the context of the scene so that it can easily infer some clues on the meaning of colors in the image. His works used a tree of possible scene context so that multiple pieces of information can be obtained.

Other works who used classification for depth are the work of Battiato et al [53] where they used unsupervised learning and classify input images first to estimate depth. Jung & Kim [36] classified first the geometry of the scene in

order to get the context and create a range possible depth. The research group in Spain [30], [54]–[56] used binary partition tree to classify the images to estimate the relative depth.

2.9 Research Gap

In summary, the literature has a research gap on estimating depth for close-up image sequence like in BSE. Currently, there were two research works that tackled on this topic. The first research by Chen et al is related to the BIOCORE's BSE multimedia system [19]. They were able to fit a linear model to predict a BSE pressure level estimation. However, this work is not yet applicable for realistic performance, as it has not able to tackle the natural lateral and rotational hand movement during the palpation process. The second research work by Cabatuan et al tackled the same problem with the proposed thesis. Their experimental setup is similar to our case wherein the woman is standing up with a distance of 370-410mm from the camera as shown in Figure 2-7. They have reported to obtain an 87.5% overall accuracy for estimating low, medium and high class [18]. However, their research tackled estimating the depth from a single video only. This method is not generalized for every woman possible. Hence, there is no available research work that is applicable for a realistic BSE performance.

CHAPTER 3
Theoretical Framework

3.1 Image and Video Representation

An image in machine vision is represented as a two-dimensional function, f (x, y) where x and y are its spatial coordinates. The value of each element represents its gray intensity value. For a normalized image, it can vary from 0 to 1. This representation is just for a grayscale image. Color images are represented by three grayscale images. For RGB system, the first image represents the red value of the image, the second would be the green, and the third would be blue [57]. Other color system can be devised depending on the application. Other color system can be HSV, YCbCr, CMYK, etc. In image processing, the element in an image is called pixel.

Making pictures in motion creates videos. Playing a series of images where each image is similar to the previous image creates a video. That is why, in image processing field, video is understood as a four-dimensional image, f (x, y, z, fs) where f_s is called sampling rate. Sampling rate measures the number of frames being sequentially shown per second. For a real-time application, video should be played around 30 frames per second.

3.2 Optical Flow

Optical Flow is an image processing technique wherein the motion of an object is estimated using the vector fields of the successive frames. Vector fields of an image can be visualized similar to what is seen in Figure 3-1. It can be observed that the image has motion when the vectors and its neighbor are going uniformly towards a certain direction. The magnitude of the velocity can also be observed by looking how vectors are close with each other.

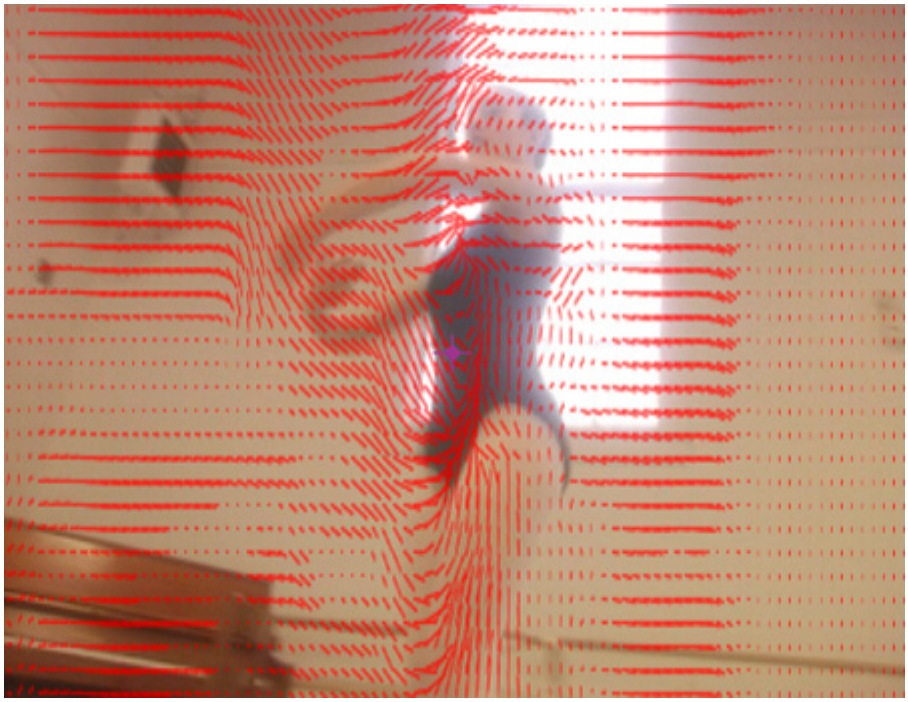


Figure 3-1. Vector Fields in an Image [58]

However, computers do not understand such qualitative approach. In order to realize the estimation of motion using vector fields, one of the famous methods is the Lucas-Kanade algorithm. Lucas-Kanade has two assumptions:

- The two successive frames are only separated with small increments of Δt.
- The object of interest has different level of gray intensity with respect to the background. [59]

In a nutshell, given a pixel with a specific gray intensity level, the algorithm will simply search on what direction the pixel went. Shown in Figure 3-2(a) is the pixel located at (x, y) with gray intensity level a in frame A. In the successive frame, the pixel in the same location has now a gray intensity level of b as shown in Figure 3-2b. Of course, it seems plausible to believe that the pixel with intensity *a* just moved nearby. It might just have displaced. If the rate

of increase of brightness in the x-direction, $I_x(x, y)$, and the rate of increase in the y-direction, $I_y(x, y)$, then with the equation:

Equation 3-1

$$I_x(x,y) \cdot u + I_y(x,y) \cdot v = b - a$$

Where:
u = Number of pixels moved in the x-direction
v = Number of pixels moved in the y-direction

Hence, by moving with *u* pixels in the x-direction and *v* pixels in the y-direction, the pixel with intensity *a* can now be located [60].

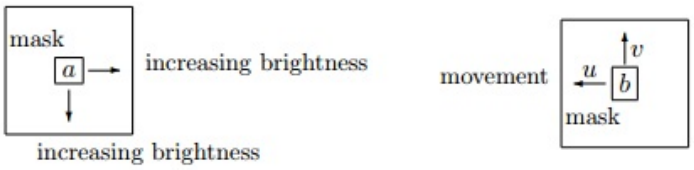


Figure 3-2. A pixel in the same location (a) frame A, (b) frame B [59]

However, it can be seen that the equation is underdetermined. It is a one-equation-two-unknown problem. To add more equations, the neighbors of the pixels can be added to more equation. For example, a 3x3 window, where there are 9 pixels, will be used to gather the system of equation. The assumption is that the 9 pixels have a uniform movement and refers to the same object. Hence, it will have 9 equations two unknown. This creates the system as overdetermined. The Lucas-Kanade algorithm tells that the user should use least square method to obtain the best fit curve [61]. Overall, the result of the algorithm is to create an optical flow vectors similar to what is seen in Figure 3-1. Moreover, the magnitude and direction of velocity fields can be extracted [62].

(a) (b)

(c) (d)

(e) (f)

(g)

Figure 3-3. Samples of Optical Flow Field: (a) original image, (b) left, (c) right, (d) up, (e) down, (f) zoom in, (g) zoom out

3.2.1 Optical Flow Field

Figure 3-3 shows samples of optical flow fields. In the (b), it was shown that when the object moves to the left (in the reader's perspective), the flow field shows multiple lines of force having a direction towards left. Figure 3-3 (c), (d) and (e) shows the flow field when the object moves to the right, up and down. In this study, it is also notable that whenever the object goes in as shown in Figure 3-3 (f), it creates a flow field showing lines of forces going out. Figure 3-3 (g) shows the object going out, it creates a flow field that goes out. Mathematically, this phenomenon is called divergence. Divergence measures the net flow of a vector field. When it is positive, then there is a net outflow. If it is negative, then it has a net inflow. [63]. Equation 3-2 shows the formula used to calculate the divergence in rectangular coordinates.

$$div(D) = \frac{\partial D}{\partial x} + \frac{\partial D}{\partial y} + \frac{\partial D}{\partial z}$$ **Equation 3-2**

Where:
D – Vector Field

3.3 Stereo Depth vs. Monocular Depth

In a conventional situation, depth maps are calculated by measuring the disparity between the left image and right image as shown in Figure 3-4. Conventionally, it requires a pair of stereo images in order to obtain the depth map needed [64]. This is similar to what the human eyes use in order to estimate depth called stereopsis.

Recent researches right now focuses on how to obtain depth using only a single camera. This scenario has more practical situation. It will be very helpful in the field localization and mapping, 3D reconstruction, 3D movie production, etc. Hence, in order to separate from the conventional scenario, this research is called monocular depth. It is termed monocular, as it is not stereo; it only uses one camera.

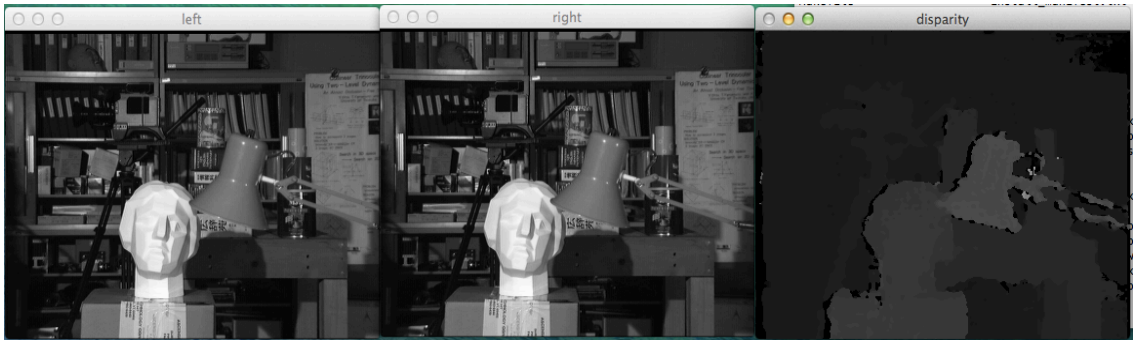


Figure 3-4. OpenCV example for creating a Disparity Map [65]

The goal of monocular depth is to generate a depth map using only monocular image or image sequence. It is still an on-going research worldwide. Several methods have been exploited in order to generate depth map for 3D reconstruction. The most mature method is to use structure from motion. However, it requires non-planar images. Meaning, the camera should be rotating.

3.4 Depth Map

There are two types of Depth Map: relative and absolute. Relative depth map are easier to construct. It is the depth map used for 2.5D. Relative depth map represents only the ordering of the objects. A simple example would be an image of bowling bowl at different order of depth as in Figure 3-5. The whiter the color, the nearer it is in the viewer. Hence, the whitest object is the considered the front object and the black object is the background of the image. In Figure 3-5, the yellow object is the topmost object hence its corresponding pixels in depth map have white color. The red, blue, green and black ball follows it. That's why the shade have become darker compared to its object ahead. Relative Depth map provides the depth ordering of the object in the image. However, it does not provide how far it is from the viewer.

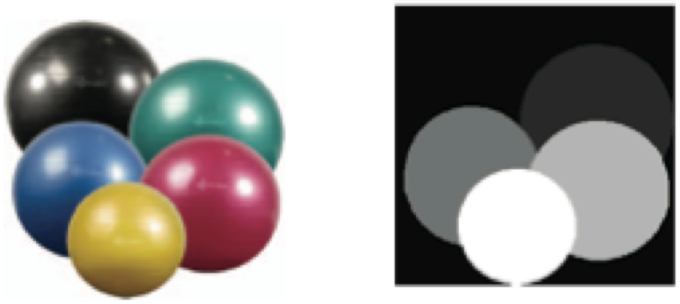


Figure 3-5. Bowling Balls at various depth order [30]

Absolute depth map provides the estimation of the focal distance from the camera. Hence, topmost object does not necessarily have corresponding white color. The corresponding color of pixel is now dependent on how far it is from the camera. To make it clearer, two samples of absolute depth map is shown in Figure 3-6. This sets of images are obtained from the Microsoft Research's database [46]. They used Microsoft Kinect Xbox 360 to gather RGB and depth data. Its depth map is interpreted by observing the intensity of color. The white is considered the farthest while the black is considered nearest. By comparing the two sets of images, the topmost object in Figure 3-6(a) (i.e. the girl) is farther from the camera compared to the topmost object in Figure 3-6(c) (i.e. the man). But, depth map of the girl in (b) is more greyish than the depth map of the man in (d).

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

Figure 3-6:(a) girl in elevator, (b) its depth map, (c) man on hallway, (d) its depth map [46]

3.5 RGBD Images

With the rise of 3D cameras, these sets of cameras do not just capture color images (i.e. RGB images) but also depth image. It has been known to all that given an RGB image, it provides the spatial information. However, it has removed the depth information of the real-world objects. In Figure 3-6(a), the image shows a woman standing in front of an elevator. However, just by observing the image alone, no one can directly know how far is the woman from the elevator. However, the person who took the image can actually estimate the distance of the woman from the elevator.

This problem is due to that fact whenever real-world points are projected to the camera image, the camera shall only capture the projection of that point to the camera as in Figure 3-7. In the figure, the points *Pi* and C*i* seems to be close to each other in reference to the camera image. However, in the real world coordinates, it is actually farther than what can be seen from the camera image. That's why engineers developed a camera that can capture all spatial dimensions called, in layman's term, 3D cameras. But for researchers, this is called RGBD cameras.

RGBD cameras stands for camera that can capture both RGB image plus Depth image. If the RGB image provides the spatial color information of each pixel, the depth image provides the absolute distance of that pixel in reference to the camera. Hence, ideally, RGBD camera provides a holistic view of real world coordinates.

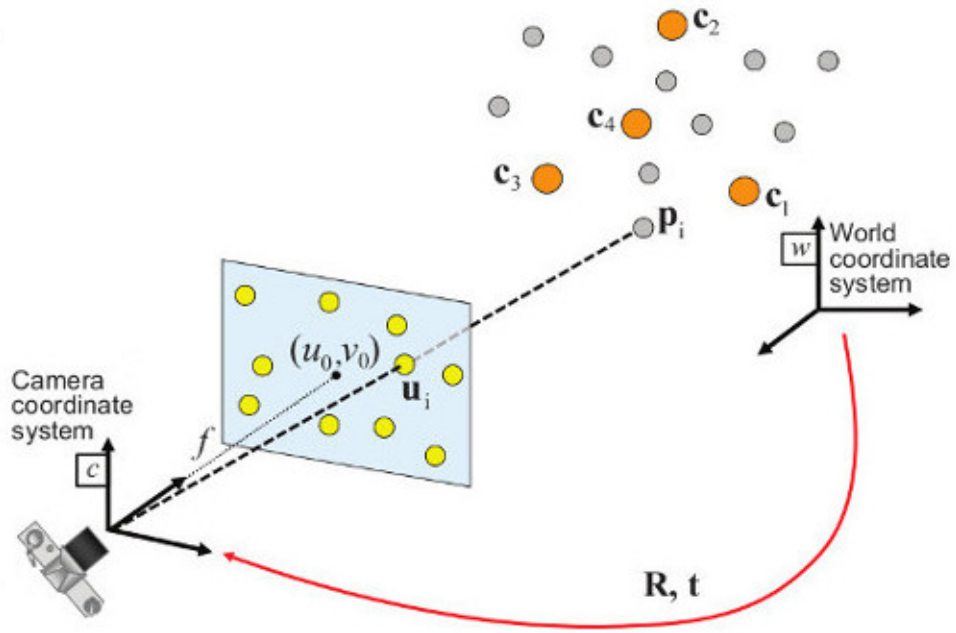


Figure 3-7. Camera projections of real world points (source: [66])

The trick in capturing RGB and Depth images is to utilize one camera for RGB image and another camera for Depth Image. An example of which is the Kinect for Xbox 360 as shown in Figure 3-8. The camera labeled as 2 is the RGB camera. While, the cameras labeled as 1 is the depth camera. One person might notice that there are two cameras labeled as 1. However, when using the Kinect, only one depth image is produced. Actually, there are two depth cameras because of how Kinect's capturing scheme is done. They utilized a method called triangulation[67]. The idea is left depth camera will capture his own depth Image while right depth camera does the same. Now, these two frames shall pass an internal post-processing that constructs the new depth based on these two images and some specifications of Kinect.

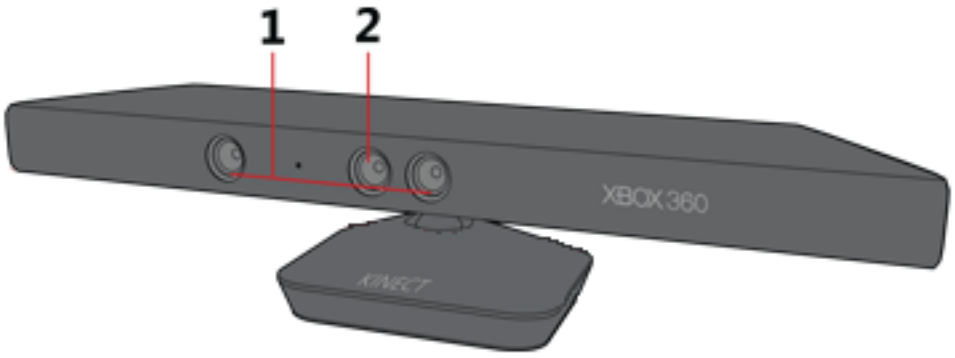


Figure 3-8. Camera components of Kinect for Xbox 360 (Source: [68])

3.6 Linear Programming

Linear programming is a branch of mathematics wherein it attempts to find the vector that provides the global minimum of a linear expression given a system of linear inequalities as constraint. Concretely,

$$\min_{\theta} \boldsymbol{\theta}^{\boldsymbol{T}}\boldsymbol{X} \qquad \textbf{Equation 3-3}$$

Subject to:

$$\boldsymbol{A} * \boldsymbol{X} < \boldsymbol{b}$$

Where:
$\boldsymbol{A} \in \mathbb{R}^2$
$\boldsymbol{\theta}, \boldsymbol{b} \in \mathbb{R}$

3.6.1 L1 Norm Minimization

Lp norm minimization is defined as

$$minimize \ \|\mathbf{Ax} - \mathbf{b}\|_p \qquad \textbf{Equation 3-4}$$

$where$:
$A \in \mathbb{R}^{mxn}$
$x \in \mathbb{R}^n$
$b \in \mathbb{R}^m$
$\|\cdot\|_p \ is \ the \ p-norm \ on \ \mathbb{R}^m$

If the residual, $\mathbf{r}$, is defined as $\mathbf{r} = \mathbf{Ax} - \mathbf{b}$, then L1 norm can seen as sum of absolute residuals,

$$\|\mathbf{Ax} - \mathbf{b}\|_1 = |r_1| + |r_2| + \cdots + |r_m| \quad \textbf{Equation 3-5}$$

In the context of Linear Programming, the equation above is not an LP problem, as it requires minimizing sum of absolute values while LP considers only the linear combination of X. But, it can be converted in to an LP problem using:

$$minimize\ \mathbf{1}^T\mathbf{t} \quad \textbf{Equation 3-6}$$

$$subject\ to:$$

$$\mathbf{Ax} - \mathbf{b} \leq \mathbf{t}$$
$$-\mathbf{Ax} + \mathbf{b} \leq \mathbf{t}$$

Where:
$\mathbf{A} \in \mathbb{R}^{mxn}$
$\mathbf{x} \in \mathbb{R}^{n}$
$\mathbf{t} \in \mathbb{R}^{m}$

3.6.2 L1 Norm minimization for Learning Parameters

The succeeding concept are followed from the success of Saxena's work [20], [33], [34]. This same concept can be used as a method for implementing Multiple Linear Regression scheme. The general idea is to find the hyperplane that minimizes error between the actual value and the predicted value. Now, quantifying error can either be L2 norm or L1 norm (Section 3.6). In this study, L1 norm minimization is the algorithm of choice as it is robust to outliers compared to L2 norm.

$$minimize \sum_{K=1}^{N}\sum_{i=1}^{M}\left|d_{i,K} - X_{i,K}\theta\right| \quad \textbf{Equation 3-7}$$

Where:
$M = number\ of\ patches$
$N = number\ of\ training\ images$

To convert it into a linear programming problem

$$minimize \sum_{i=1}^{M} \sum_{K=1}^{N} \boldsymbol{\varepsilon}_{i,K} \quad \textbf{Equation 3-8}$$

$subject\ to$:

$$-\boldsymbol{\varepsilon_1} \leq X_{i,K}\boldsymbol{\theta} - d_{i,K} \leq \boldsymbol{\varepsilon_1}$$
$$\vdots$$
$$-\boldsymbol{\varepsilon_{M,N}} \leq X_{M,N}\boldsymbol{\theta} - d_{M,N} \leq \boldsymbol{\varepsilon_{M,N}}$$

3.7 Image Color Space

Perhaps, the most famous color Space even to non-researchers is RGB. In computer vision, an RGB image is implemented by creating an *m x n x 3* matrix where m is the pixel height and n is the pixel width. The first *m x n* matrix contains the red pixels, the second contains the green pixels and the last contains the blue pixels [57]. Together, they form images commonly seen in phones, computers and tablets. However, there are other color spaces that can be very useful in other applications. Some examples are YCbCr, HSV, Lab, YUV, etc. In this section, only three-color spaces shall be discussed.

3.7.1 Grayscale Color Space

Grayscale image is a data matrix that contains the information of the said image in the shades of gray. Grayscale Images are also known as Intensity Image. There are two commonly used data type for its pixel: (1) *unsigned* char, (2) *unsigned short*. If each pixel has a data type of *unsigned char,* then the range of values is [0,255]. If the pixel has a data type of *unsigned short, then* the range of values is [0,65535].

Depth Images from RGBD cameras are also considered grayscale image. However, each pixel contains the absolute depth in reference from the camera rather than the actual shades of grays information. In many cases though, the depth Images are visualized as a grayscale Image.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$ **Equation 3-10**

If given RGB Image and the user wants to utilize the intensity image map, then they can use the formula as in Equation 3-14[57].

$$I = 0.299R + 0.587G + 0.114B$$ **Equation 3-9**

3.7.2 YCbCr Color Space

The YCbCr color space has been extensively used for digital video. In this color space, the intensity is contained in the luminance (Y). The color information is stored in the blue and red chroma (Cb and Cr). The chrominance Cb is defined as difference of blue value and a reference value. While, the chrominance Cr is defined as difference of red value and a reference value. The conversion of RGB color space to YCbCr is shown in Equation 3-10[57].

3.7.3 C1C2C3 Color Space

C1C2C3 color space is an uncommon color space. In many computer vision packages, there is no direct conversion from RGB to C1C2C3. But its advantage is that this color space has photometric invariance [69]. Meaning, even if the object undergo some lightness variation, c1 c2 c3 values should contain the same values of pixel as it is invariant to light intensity (i.e. photometric) variation. The RGB pixel to C1 C2 C3 conversion is shown in Equation 3-11, Equation 3-12 and Equation 3-13.

$$c1 = \tan^{-1}\frac{R}{\max(G, B)}$$ **Equation 3-11**

$$c2 = \tan^{-1}\frac{G}{\max(R,B)}$$ **Equation 3-12**

$$c3 = \tan^{-1}\frac{B}{\max(R,G)}$$ **Equation 3-13**

3.8 Textures Features

3.8.1 Laws' Texture Filters

In search for rapid texture identification for image segmentation, Kenneth Laws developed 9 5 x 5 kernel mask that will characterize the texture of an image[70]. It was created with 4 vectors, namely: local average vector (L5), edge vector (E5), ripple vector (R5) and Spot vector (S5).

$$\boldsymbol{L5} = [\ \mathbf{1}\quad \mathbf{4}\quad \mathbf{6}\quad \mathbf{4}\quad \mathbf{1}]^{\boldsymbol{T}}$$ **Equation 3-14**

$$\boldsymbol{E5} = [\mathbf{-1}\quad \mathbf{-2}\quad \mathbf{0}\quad \mathbf{4}\quad \mathbf{1}]^{\boldsymbol{T}}$$

$$\boldsymbol{S5} = [\mathbf{-1}\quad \mathbf{0}\quad \mathbf{2}\quad \mathbf{0}\quad \mathbf{-1}]^{\boldsymbol{T}}$$

$$\boldsymbol{R5} = [\ \mathbf{1}\quad \mathbf{-4}\quad \mathbf{6}\quad \mathbf{-4}\quad \mathbf{1}]^{\boldsymbol{T}}$$

The uses of each vector can be derived from their names alone: L5 produces local average, E5 detects edges, S5 detects spots and R5 detects ripples. Now, in his next step, what he did is to produce the 9 5x5 kernels. To do this, he simply multiplies the transpose of one vector with another vector and do this with all combinations. All 16 combinations is shown in Table 3-1 and Table 3-2.

Table 3-1. Laws' Kernel part 1

Name of kernel	Equation	Kernel matrix
L5L5	$L5^T * L5$	$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$
E5E5	$E5^T * E5$	$\begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 2 & 4 & 0 & -4 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -4 & 0 & 4 & 2 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}$
S5S5	$S5^T * S5$	$\begin{bmatrix} 1 & 0 & -2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 1 \end{bmatrix}$
R5R5	$R5^T * R5$	$\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$
L5E5	$L5^T * E5$	$\begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}$
E5L5	$E5^T * L5$	$L5E5^T$
L5S5	$L5^T * S5$	$\begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$
S5L5	$S5^T * L5$	$L5S5^T$
L5R5	$L5^T * R5$	$\begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ 4 & -16 & 24 & -16 & 4 \\ 6 & -24 & 36 & -24 & 6 \\ 4 & -16 & 24 & -16 & 4 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$
R5L5	$R5^T * L5$	$L5R5^T$

Table 3-2. Laws' Kernel part 2

Name of kernel	Equation	Kernel matrix
E5S5	$E5^T * S5$	$\begin{bmatrix} 1 & 0 & -2 & 0 & 1 \\ 2 & 0 & -4 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 4 & 0 & -2 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$
S5E5	$S5^T * E5$	$S5E5^T$
E5R5	$E5^T * R5$	$\begin{bmatrix} -1 & 4 & -6 & 4 & -1 \\ -2 & 8 & -12 & 8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & -8 & 12 & -8 & 2 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$
R5E5	$R5^T * E5$	$E5R5^T$
S5R5	$S5^T * R5$	$\begin{bmatrix} -1 & 4 & -6 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & -8 & 12 & -8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -6 & 4 & -1 \end{bmatrix}$
R5S5	$R5^T * S5$	$S5R5^T$

To create the texture map, one Laws' kernel shall be convolved with intensity Image. Now, from 16 combinations of kernels, only 9 texture maps are created. This is seeing the list below:

E5E5	L5E5/E5L5	E5R5/R5E5
S5S5	E5S5/S5E5	L5R5/R5L5
R5R5	L5S5/S5L5	R5S5/S5R5

One may wonder why there are two kernels side-by-side with each other. It means that the first step is to convolve first kernel with intensity Image. The second step is to do the same thing with the second kernel. Now, there will be two textures map. The last step is to do pixel-wise averaging between the two texture maps [71]. After processing all kernels as listed above. There should be nine (9) texture maps with dimensions equal to the input image.

Now, the texture Maps are full images. After obtaining all nine texture maps, many variations are seen depending on its applications. The idea is to use any statistical texture energy descriptors. A partial list is seen in the table below. Some of which are reference from the book of Gonzalez et al[57].

Table 3-3. Statistical Texture measures

Statistical Descriptors	Expression	Behavior
Mean	$\frac{1}{N}\sum_{i=0}^{N-1} I(i)$	Measures the average texture energy measure
Standard Deviation	$\sqrt{\sum (I(i)-\mu)^2/N}$	Measures the dispersion of texture energy
Smoothness	$1-\frac{1}{1+\sigma^2}$	Measures the smoothness of the intensity in a local region
Third Moment	$\sum_{i=0}^{N-1}(I(i)-\mu)^3\left(\frac{1}{I(i)}\right)$	Measures the skew of the texture histogram.
Entropy	$-\sum \mathrm{I(i)}\log I(i)$	Measures randomness
Histogram		Summarizes the energy map into n bins

3.8.2 Local Binary Pattern (LBP)

Local Binary Patterns utilizes simple relational and bitwise operations but has shown to be very useful descriptor for facial recognition. It starts by extracting a 3x3 window from an intensity image. Then, the center pixel shall be compared to its neighboring pixels. If the neighboring pixel is greater than or equal to the center pixel, then it will have a value of 1 to the new window. For instance, in Figure 3-9, the center left pixel has a value of 9 while the center pixel is 5. Then, the center left pixel in the new window will have a value of 1. Now, when the center pixel compared to the upper left pixel (i.e. 1), since its

value less than the center pixel, then it contains a value of 0. After iterating to the whole window, the next step is to consider it as one byte starting from the upper-left pixel as MSB rotate until the center left pixel. Then, convert the binary byte into its decimal equivalent. If the center pixel is at location (i, j), this decimal value shall be the value of texture Map at location (i, j). In the Figure 3-9, the binary value is 00010011. Hence, it has an equivalent decimal value of 19. The texture Map at (i, j) shall have a value of 19.

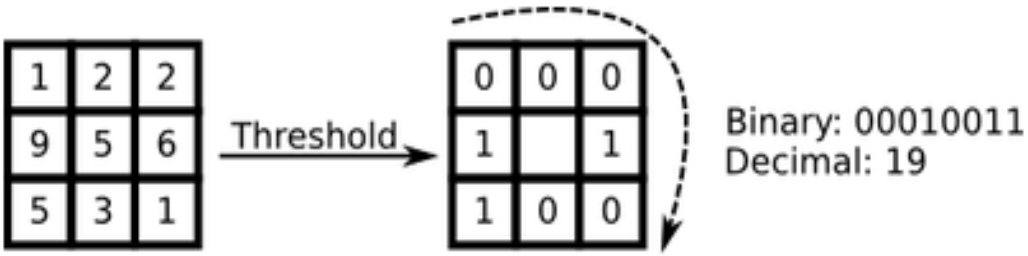


Figure 3-9. Local Binary Pattern 3x3 Window (Source: [72])

Now, this 3x3 window shall be done in the whole Image. It will be moving window one pixel to the right and/or one pixel downwards. The final texture map shall have dimensions equal to the input image. If the window is only 3x3, there can be 2^8 combinations of values. Hence, the range of values for the texture Map shall only be from 0 to 255.

Lastly, when the center pixel is a border pixel, there are two solutions. The first and simplest solution is to consider all border pixels of the texture Map as zero. The second solution is to pad the input image with zero borders.

A sample texture Map of LBP is shown in Figure 3-10. This image intuitively shows why LBP proves to be very distinctive as texture Descriptor. LBP captures even the fine details of the face shown in this image, the light inside on the eyes, the small separation among the hairs, and even the line around the eyes. Moreover, the background light in the input image is

attenuated in the texture Map. In essence, the LBP texture Map enhances the edges while attenuates the uniform background noise.

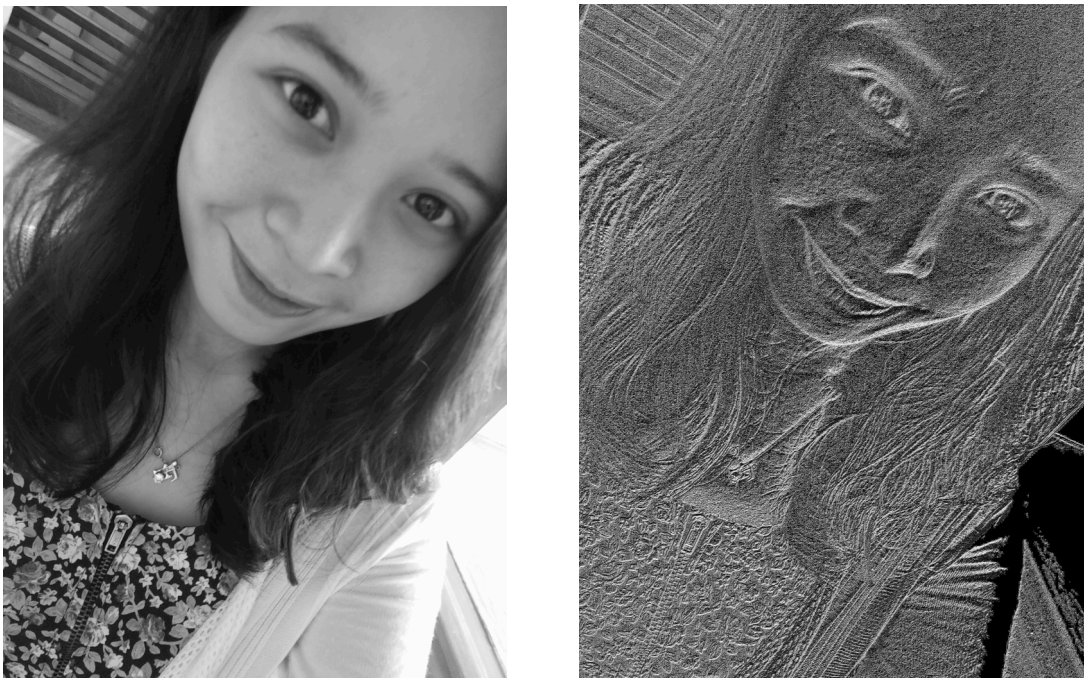


Figure 3-10. A close-up Image of a girl: (a) Grayscale Image, (b) LBP Texture Map

3.9 Machine Learning Statistical Models

3.9.1 Support Vector Machine (SVM)

Support Vector Machine can be simplified as a classifier. For a binary classification, its goal is to find the optimal hyperplane that can separable the two labels as shown in Figure 3-11. The data points closest to the optimal hyperplane are called support vectors[73]. The distance between the support vectors measures the margin between labels. Note that it is considered the optimal hyperplane whenever it maximizes the margin between labels.

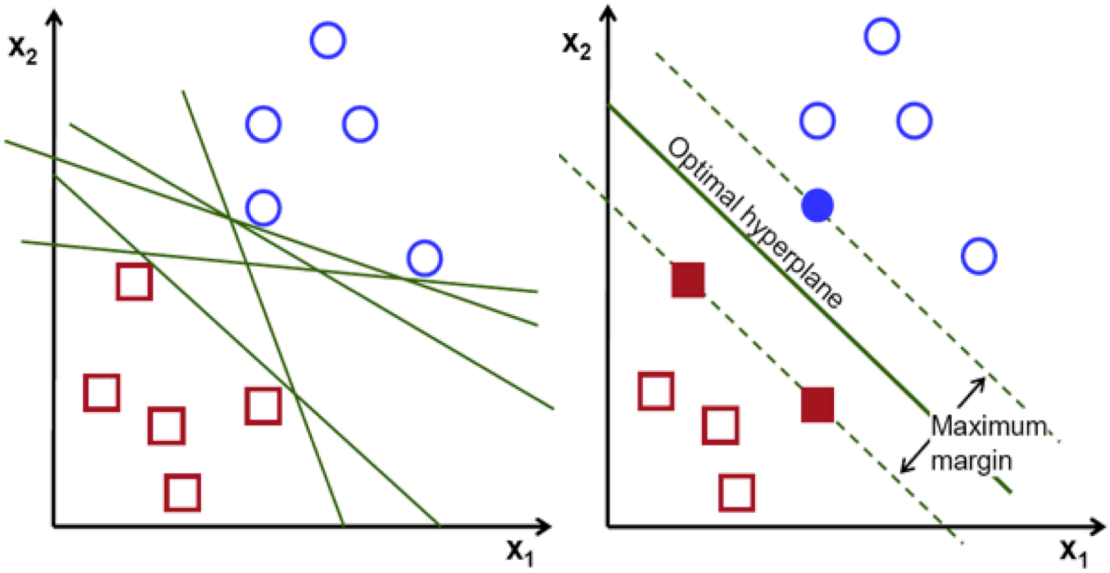


Figure 3-11. SVM Optimal Hyperplane (Source: [74])

Concretely, Given X is the feature vector the objective function of SVM is shown in Equation 3-15 [75].

$$\min_{\theta} C \sum_{i=1}^{m} [y_i Cost_1(\theta^T X_i) + (1 - y_i) Cost_0(\theta^T X)] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 \qquad \textbf{Equation 3-15}$$

The *C* parameter provides the trade-off between minimizing training error and complexity of decision boundary[76]. $Cost_1(X)$ and $Cost_0(X)$ are cost functions for label 1 and label 0 respectively. These costs functions can be linear and nonlinear in nature. OpenCV provides 4 types of kernel. Note that all point l_j are points of support vectors.

1. Linear Kernel - the fastest and simplest kernel function. Its idea is similar to max-margin linear regression.

$$Cost(X_i, l_j) = X_i^T l_j \qquad \textbf{Equation 3-16}$$

2. Polynomial – provides higher degree of decision boundary in comparison to linear kernel.

$$Cost(X_i, l_j) = (\gamma X_i^T l_j + coef0)^{deg}, \gamma > 0. \quad \textbf{Equation 3-17}$$

3. Radial Basis Function – the Gaussian kernel. Often, it provides the best prediction for nonlinear feature space[73].

$$Cost(X_i, l_j) = e^{-\gamma \|X_i - l_j\|^2} \quad \textbf{Equation 3-18}$$

4. Sigmoid – the kernel that provides the sigmoid function in SVM. When using this kernel, it is notable the SVM's structure will be similar to Multilayer Perceptron Artificial Neural Network [73].

$$Cost(X_i, l_j) = \tanh(\gamma X_i^T l_j + coef0) \quad \textbf{Equation 3-19}$$

3.9.2 Gradient Boosted Trees

The utilization of boosting algorithm is motivated by the works of Viola-Jones[77] and Mohammadi et al [1]. For Viola-Jones, they were successfully in detecting faces even with the use of haar-like wavelets whose accuracy is just above random. For Mohammadi et al, they were successful to find the position of palpation in Breast Self Examination.

Now before fully understanding Gradient Boosted Trees, there is a need first to discuss loss functions. Loss functions provide functions to quantify the error between the actual data compared to the predicted data. OpenCV supports four loss functions wherein the first three are used for regression problems while the last one is used for classification problems.

1. Squared Loss - squared loss is similar to the loss function utilized in linear regression. It estimates error by providing the squared difference between the predicted value and the actual value. Its equation is shown in Equation 3-24. Its main advantage is faster optimization since it provides higher error estimates and higher gradient value towards the

global minimum compared to absolute loss. However, its main disadvantage is its final parameter estimates are affected by outliers as it also tries to fit the estimates with the outliers.

$$L\big(y, f(x)\big) = \frac{1}{2}(y - f(x))^2 \qquad \textbf{Equation 3-20}$$

2. Absolute Loss - it is similar to L1 norm approximation. It returns the absolute difference between the predicted value and the actual value. It is shown in Equation 3-21. Its advantage is it less prone to outliers. However, the sum of values of errors is generally smaller compared to the squared loss.

$$L\big(y, f(x)\big) = |y - f(x)| \qquad \textbf{Equation 3-21}$$

3. Huber Loss - This function attempts to combines the advantage of absolute loss and squared loss. When the absolute loss is smaller than δ (i.e. 0.2 in OpenCV implementation), then it will rather use a squared loss. Otherwise, use a modified absolute loss that provides lower error estimate and lower gradient value compared to the normal absolute loss. Its equation is shown in Equation 3-22. It is notable that the Huber loss function is continuous at all points due to the modified absolute loss. Hence, it is differentiable at all points.

$$L\big(y, f(x)\big) = \begin{cases} \delta\,(|y - f(x)| - \delta/2), & when\ |y - f(x)| > \delta \\ \frac{1}{2}(y - f(x))^2, & when\ |y - f(x)| \le \delta \end{cases} \qquad \textbf{Equation 3-22}$$

4. Deviance loss - This is the function used for classification problems. Given that there are *K* output classes. Then the loss function is the entropy value of the error as in Equation 3-23. Note that this loss functions accounts the errors in all classes. It is not iteratively on each class.

$$L\left(y, f_1(x), f_2(x), \dots, f_K(x)\right) = -\sum_{i=0}^{K} 1(y = i) \ln \frac{\exp\left(f_i(x)\right)}{\sum_{j=0}^{K} \exp\left(f_j(x)\right)}$$ **Equation 3-23**

In comparison to Adaboost, Gradient Boosted Trees supports multi-class classifications. Its optimization scheme is to find the difference of the gradient of a summary loss function of the current iteration and the gradient of a summary loss function of the previous iteration that satisfies the stopping criterion as in Equation 3-24.

$$\nabla \mathcal{L}(F_{curr}) - \nabla \mathcal{L}\left(F_{prev}\right) < \varepsilon_s$$ **Equation 3-24**

Given that M is the number of features in the model, the general procedure for training the gradient boosted trees is as follows [78]:

```
For each i in M
    Compute the antigradient
    Grow the regression tree
    Change values in the tree leaves
    Add the tree in the model
    end
```

Figure 3-12. Gradient Boosted Trees Optimization Pseudo code

The objective function used is now as in Equation 3-25. Note that for classification problem, this is done for each class.

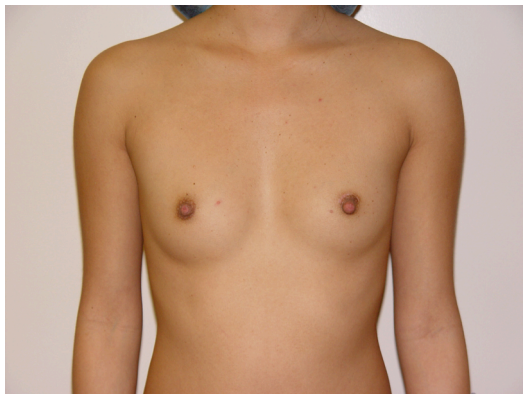
$$f_k(x) = f_{0_k} + \nu \cdot \sum_{i=0}^{M} T_i(x)$$ **Equation 3-25**

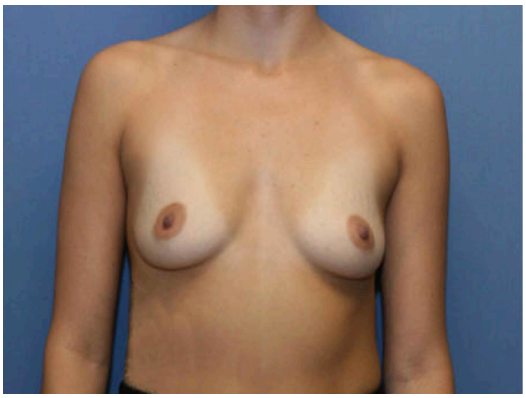
3.10 Breast Cup Size

If pressure level will be categorized to a qualitative metric: low, medium and deep. Its equivalent absolute depth will differ depending on the maximum depth it can go. To make the point clearer, compare a 10ft swimming pool to a 20ft swimming pool. If we are restricted to a 10ft swimming pool, a low, medium and deep level will have the range of 0-3.33ft, 3.34-6.66ft and 6.67-10ft

respectively. But in a 20ft, the depth level range for low, medium and high shall be 0-6.66 ft., 6.67-13.32 ft., and 13.33-20 ft. This is also similar to breast cup sizes. Sample image of cup size from A to D are shown in Figure 3-13. By observing the absolute depth level of a cup size A to cup size D, the absolute value of a “deep” in a cup size A might just be “low” in cup size D.

(a) (b)





(c) (d)

Figure 3-13. Breast cup samples: (a) cup size A [79]**, (b) cup size B** [80]**, (c) cup size C** [81]**, (d) cup size D** [82]

Moreover, since it is expected that the end user shall be standing when using the algorithm, it can be observed that there is a larger amount of breast fats in the lower region than in the upper region. Hence, depth level range shall also be different for the upper region compared to lower region.

3.10.1 Breast Modeling

In order to create a more accurate and generalized depth estimation, there is a need to classify the breast cup size of the user. “Low”, “medium”, and “deep” palpation will depend on how deep it can go on the user’s breast. Depth level threshold for each classification can be different depending whether the palpation process happens on the upper or the lower region. Although there are many variations on what is the exact depth for each cup size, the proposed model will follow the U.S. cup size chart hello World according to Wizard of Bras. The maximum depth will be based on Table 3-4 [83].

Table 3-4. Size Chart [83]

Depth	Cup Size (U.S.)
1 inch	A
2 inch	B
3 inch	C
4 inch	D

CHAPTER 4
Methodology

4.1 Design

The setup of the research is shown in Figure 4-1. At start, the user will perform breast self-examination. It shall be captured by the input peripheral: a web camera. The processing unit will be the one that will estimate the depth done by the user. The speaker will produce a feedback to the user. Note that the feedback audio might not exist as it might be done in the other research work. The focus of this research is to estimate depth. If there will be audio feedback, it will be limited to produce an audio that recognizes that the user palpated low, medium, or high-pressure level.

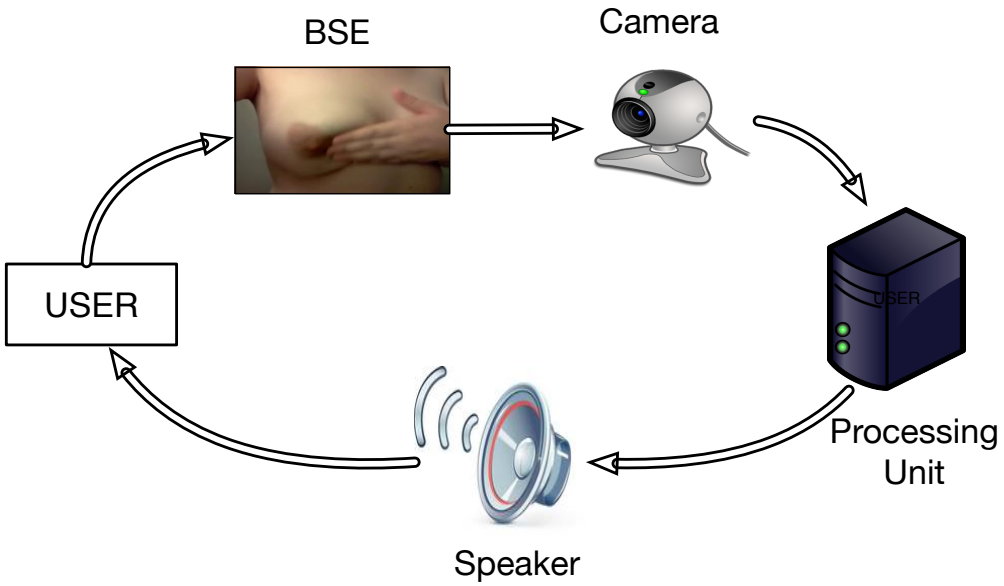


Figure 4-1. General Experimental Setup

4.2 Assumptions of the System

There are assumptions used by the system so that it can conform to the proper depth estimate such as:

- It can extract the Region-of-interest (ROI) that only contains the finger palpating a breast area which is done by the previous thesis of Mohammadi [84] .
- The user have encoded her breast cup size prior to the start of the BSE performance
- The user will fully extend their arms and rotate with respect to the vertical axis before she starts the BSE palpation process.

The first assumption is realistic as it has been created in the previous research [84]. The third assumption is part of full BSE performance wherein it has ocular inspection prior to the palpation process.

4.3 Hardware Considerations

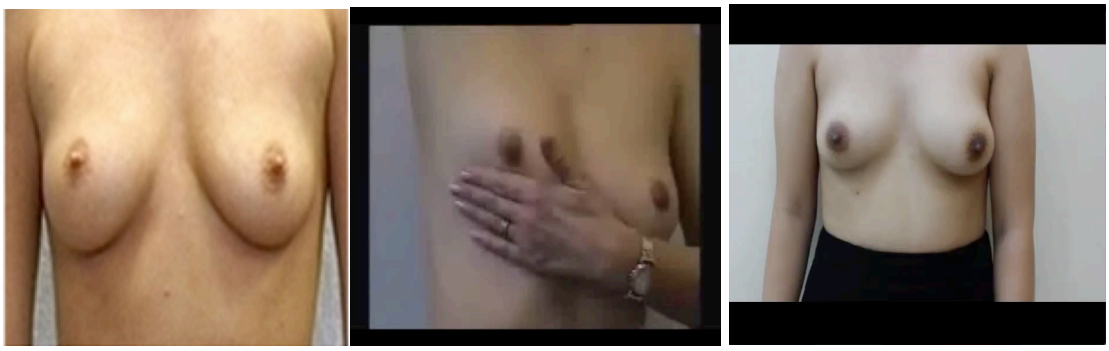
The main equipment used for processing will be a MacBook Air with a processor of 1.7GHz dual-core Intel Core i7, a GPU of Intel HD Graphics 5000 and a 8GB DDR3 1600Hz Memory. The camera to be used is the built-in camera, 720p Face Time HD camera. However, in case there is a need for an external view, a simple external VGA webcam will be used with a minimum resolution of 640 x 480 pixels. The software will mainly be based on MATLAB and/or C++ with OpenCV only.

A Kinect for Xbox 360 was used to record the RGBD database. Its actual technical specification as studied by Khoshelham[67]. Kinect's accuracy is comparable to a high-end 3D laser scanner with less than 3 meters. Moreover, Kinect for Xbox 360 has already been used for many standard RGBD dataset like NYU Dataset [85], [86], Sun3D dataset[87], B3DO[88], Cornell-RGBD[89], and Stanford[90].

The maximum resolution of RGB and Depth for a 30 frame / second recording is 640 x 480. And the minimum distance from the camera for normal and near mode is 0.8 and 0.5 m, respectively.

4.4 Datasets

There are three existing database available in the research group. The first database is the collection of breast images downloaded from the internet by the previous researcher [84]. It can be used for testing the cup model algorithm. The second database is a set of videos downloaded from the Internet. These are BSE videos done by anonymous users. The third database is the set of videos recorded and sponsored by the CHED-PHERNET. These two databases are useful for estimating depth. However, it may still require annotation for the proper low, medium and high-pressure level.



(a) (b) (c)

Figure 4-2. Sample Images: (a) Database 1, (b) Database 2, (c) Database 3

4.4.1 Gathering of New Datasets

There is a need to acquire new database, as the first objective requires depth map estimation. There is no existing BSE and depth map database available. Hence, it will require recording new BSE videos using RGB-D camera. It will be needed to quantify the amount of error in the depth estimation.

4.4.2 Methodology

Before the recording, the subjects are chosen such that they would have different cup sizes, and their age shall be less than 35 years old. The age shall be limited up to 35 years old because older women tend to have more tender breast than than younger women. The study focused first on developing an algorithm for the younger women. Subjects were also properly informed beforehand on all things during the recording proper. The surrounding place of the women volunteers was cleared so that the recorded video shall not have objects having any shades of color similar to the skin of the subject.

The research also requested the supervision from Dr. Reynaldo Joson, the Medical Expert. Although all medical practitioners knows the concept of Breast Self examination, the physicians who specializes in either oncology, gynecology, and surgery are preferable since their fields really apply Breast Self Examination, Clinical Breast Examination and/or Mammogram in their practice. Dr. Reynaldo Joson specializes in surgical oncology and breast surgery [91] that's why he is good reference for Breast Self Examination RGBD video recording. Prior to the recording, he taught them the proper breast self-examination. He taught them his method of Clinical Breast Examination he used in practice, which we translated to Breast Self Examination. The whole process of BSE starting from visual inspection up until nipple discharge is comprehensively taught to them. It should be noted that Dr. Joson and the researcher agreed to create a standard wherein the first circular motion shall be a low palpation, second shall be medium palpation, and third shall be deep palpation.

The clinician supervised breast palpation proper. Ideally, with their clinician's instruction and intensive practice before the recording proper, it shall be the basis for pressure estimation. The RGB-D camera's output shall be used as the quantitative reference. Some famous RGB-D cameras are Microsoft

Kinect for Xbox and Asus Xtion Pro Live. In this study, Kinect for Xbox is chosen.

4.4.3 Camera Setup

There are three considerations in setting up the Kinect:

(1) The camera position is setup in such a way that only the subject's torso will be seen in the camera. The RGB image should at least see the shoulder line in the upper part of the image and the belly button in the lower part of the image.
(2) The area of the torso should be maximized.
(3) The camera's distance should meet the minimum distance, i.e. 0.5 m.

The first constraint is due to the non-disclosure agreement between the researcher and the subjects. The second constraint is due to the limited RGB resolution of Kinect. And the third constraint is due to the depth processing limitation of Kinect. The specifications and accuracy of Kinect is discussed in section 4.3. Now, this setup of Kinect shall be experimentally found during the actual RGBD BSE recording. It should be noted that the actual distance from the Kinect camera was experimented during the actual recording. By trial and error, the best distance between the Kinect and the subject is around 0.6 to 0.8m.

4.4.4 RGB and Depth Map Extraction

During the actual recording, in order to preserve all details from the Kinect, the native software called Kinect studio is utilized. This is the Microsoft developed software for Kinect for Windows developers. This type of connection was comprehensively studied by Khoshelham and Elberink [67]. It can be shown that the software interface used for Kinect for Windows is still compatible with Kinect for Xbox 360.

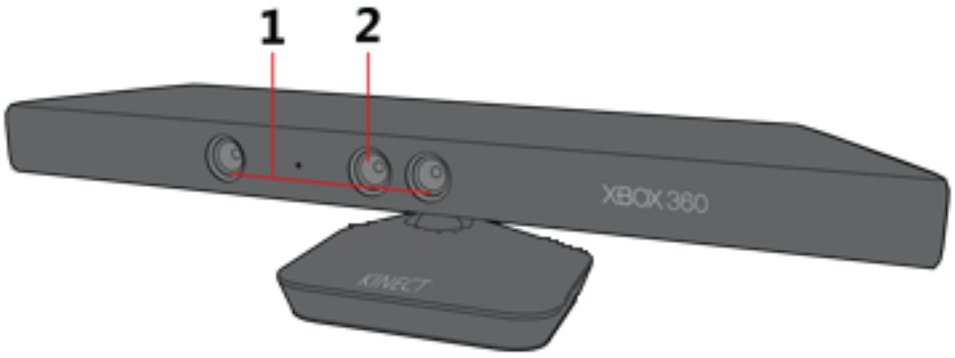


Figure 4-3. Camera Components of Kinect for Xbox 360 (Source: [68])

Unfortunately, to preserve all details, Microsoft created their own format for Kinect, which is XED format. There was no knowledgeable XED converter to common image or video format available in the Internet that is applicable to this study. The commonly available converter truncates the 14-bit depth value of Kinect to 8 bits. It will effectively change the depth pixel format from 1mm / unit to 1.6cm / unit. This study requires a change in depth with at least a format of 1mm / unit as the Breast palpation only produces low change in absolute depth value.

The second reason is that common converter does not include alignment of RGB and depth image. As shown in Figure 4-3, there are three cameras on an Xbox 360 Kinect. The camera labeled as one (1) is the RGB camera. While the cameras labeled as two (2) is the depth camera. As seen from the figure, the RGB and depth camera have gaps between them. They would then capture images with different perspective. This idea is similar to the different perspective seen by our left eye compared to the right eye. To create the perfect 3D point cloud, the RGB and depth images should be aligned with each other. This problem is called image registration.

Hence, the researchers created their own wrapper that can write depth with 16 bits and with image registration. The flowchart for extracting RGB and

Depth from Kinect is seen in Figure 4-4. It should be noted that this style of coding was designed for multi-threading process. Initialize first the Kinect then check whether it is available for grabbing the frames. If it does, from the right side, extract first the depth frame from the image stream of Kinect. Lock the frame to avoid the Kinect in grabbing new frames. Use the built-in depth map coordinates to color map coordinates for RGB alignment later on. Extract the 16-bit depth from the locked frame. Then unlock the image stream. Afterwards, from the left after status checking, check if the Depth frame grabbing is done. Extract the RGB frame from Kinect stream. Lock the RGB frame. Grab the proper RGB pixels using the Color coordinates map from the Depth Frame grabbing. Then unlock stream.

The actual code snippet that uses Kinect SDK and OpenCV is seen in Appendix. This code is compiled with Intel Thread Building Block and Intel Multi-thread Compiler. It was able to capture up to 20 frames/ second. To minimize error, it should be noted that the RGB and depth image sequence were written in PNG format that uses lossless compression.

4.4.5 Box Mask Sequence

RGB and Depth Map were extracted from Kinect as discussed in the previous section. However, Kinect captures the whole torso of the subject. In this study, what is needed is the specific region where the subject palpates her breast. This is due to the assumption that researches about extracting this specific region was already done by [2], [84]. So, to extract the region of interest in both RGB and Depth Image, an interactive Image Processing tool called Adobe Photoshop was utilized. The idea is to create a mask frame sequence that contains a rectangular box that indicates the region of interest. For each frame, we manually annotate the ROI based on the following criteria:

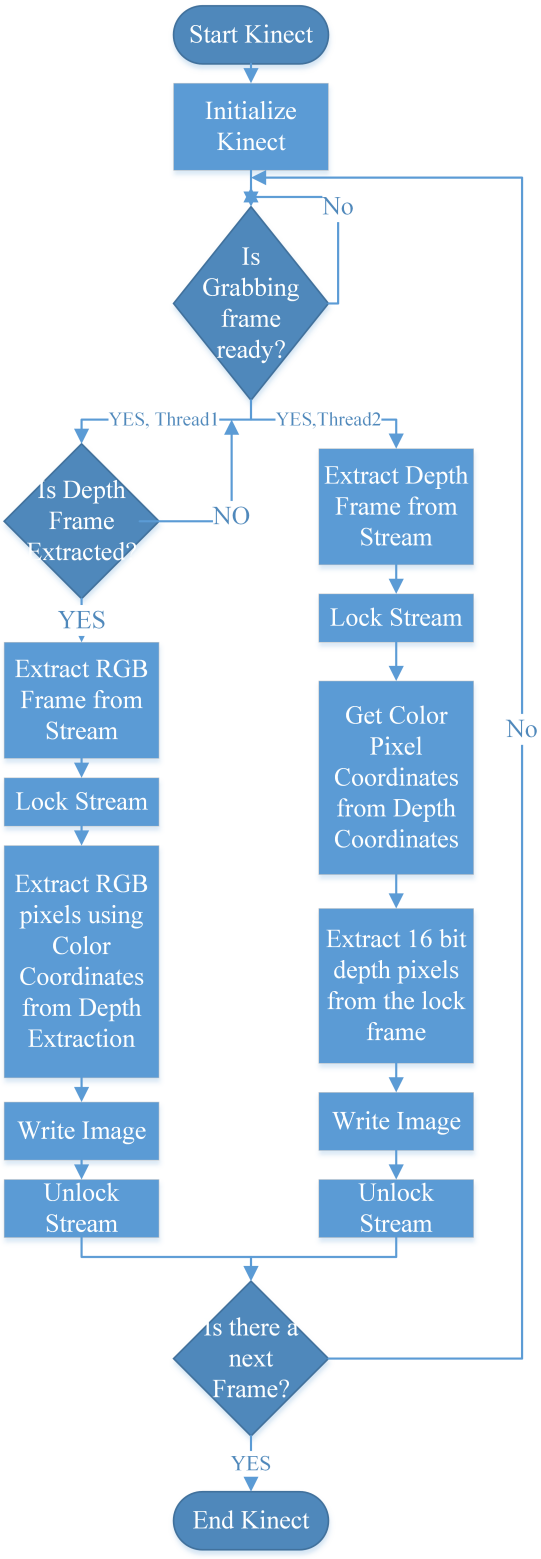


Figure 4-4. Kinect RGB and Depth Extraction Flow Chart

(1) An ROI is a box containing the three main fingers for palpation, specifically, index, middle and ring finger.
(2) To capture the texture of both finger and the pressed area, the ROI will only contain the finger up until the second joint of the middle finger. The ROI should contain the all textures of the pressed area.
(3) The ROI should also contain some unpressed area just enough to make the pressed area differentiable to the unpressed area.
(4) Each quadrant shall have uniform dimensions of ROI rectangular box.

By using criteria (1) to (3), it will focus only the features of the depth and the current depth palpation and disregards other dominant features irrelevant to depth. Some notable irrelevant dominant features are the bending of third joint of the three main fingers, the movement of breasts due to palpation, etc.

In criteria (4), it is specified that each quadrant shall uniform dimensions of rectangular box. However, the uniformity of dimensions for all quadrants and cup size cannot be done as it will remove some details in some quadrants or it will provide too many details in other quadrants. Hence, the uniformity has been restricted for each quadrant.

Figure 4-5 shows a sample sequence of the said box Mask. The images are sequentially arranged from left to right, top to bottom. The rectangular box with the true color is the image and depth ROI. The other pixels with false color are considered the background that will not be considered for further processing. . It is notable that the box Mask sequence is simply a mask. In the program, it is only a black and white Image. It cannot directly extract the image and depth ROI. The extraction of the ROI is discussed in Section 4.4.6.

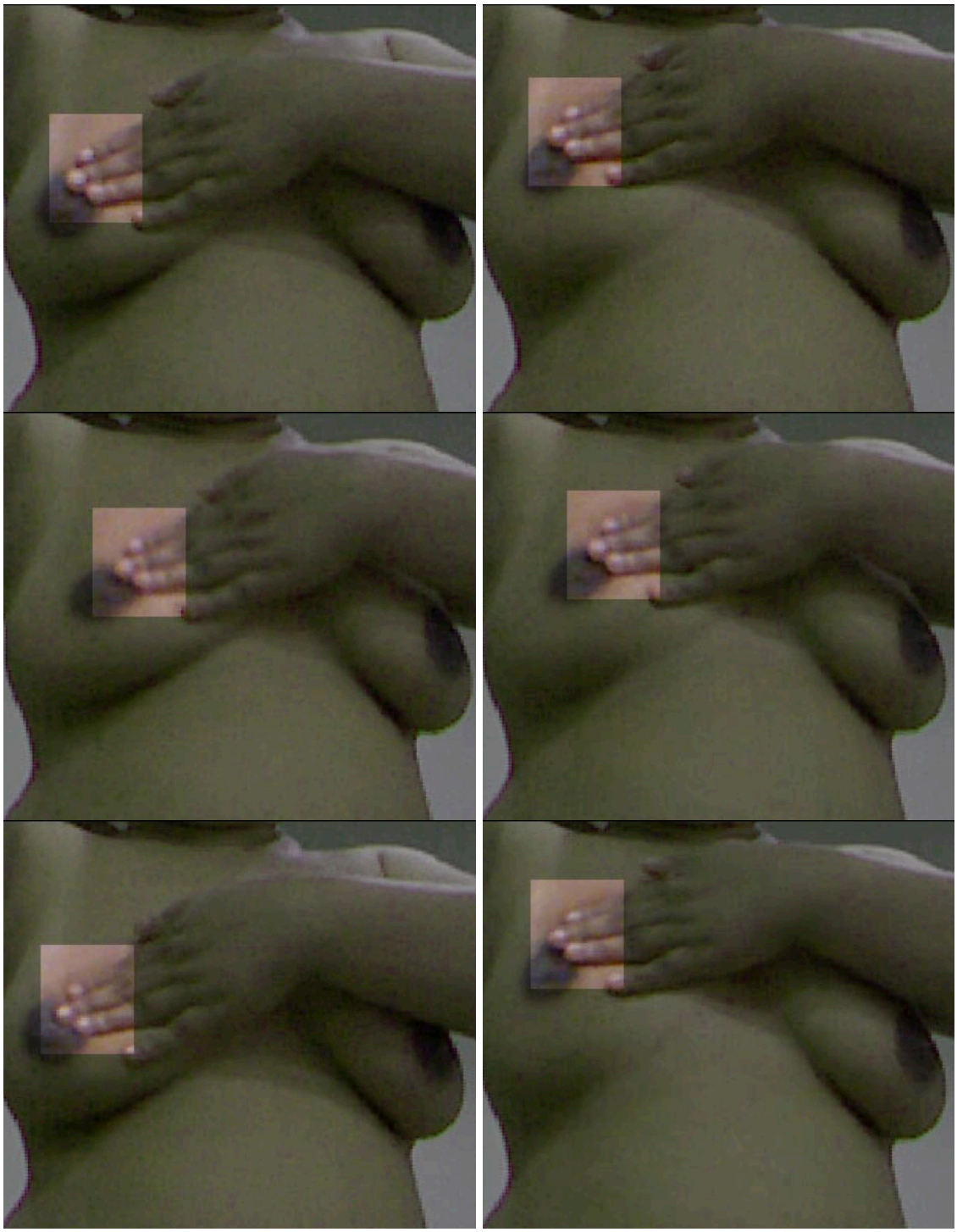


Figure 4-5. Sample Box Mask Sequence

4.4.6 Creating training and testing images

This section discusses the general step for creating the actual training and testing images utilized for the depth level estimation algorithm. Meaning, from the converted Kinect RGBD video as discussed in section 4.4.4, this section provides post-processes for the RGB images of the RGBD BSE dataset before making it as input to the depth level estimation algorithm. The post-processes of Depth image of the RGBD BSE dataset shall be discussed in section 4.4.8. There are four general steps and it is shown in Figure 4-6.

Figure 4-6. The four steps in creating training and testing images

In a comprehensive Breast Self Examination, the subject palpates the breast by not just in one area, but the whole breast area. In this study, the method devised in this study to palpate the whole breast is to segment it into quadrants for the left and right breast. The labeled used for this study is shown in Figure 4-7. The standard used for quadrant labeling is taken from U.S. National Cancer Institute [92]. But for convenience, the left and right breast is the left and right relative to the viewer rather than relative to the subject.

Step 1 and 2 of Figure 4-6 means that the after converting the dataset, partition the dataset first if it is left or right breast (Step 1). Then, for the left and right breast images, partition it into quadrants based on Figure 4-7 (Step 2). For instance, one whole RGBD BSE video contains 4000 pairs of RGB and depth frames. After step 1, the video will be subdivided into 2 (i.e. for left and right breast). Hence, there will be 2000 pairs for the left breast and 2000 for the right breast. In doing step 2, the 2000 pairs of the left breast shall be divided again

into 5 quadrants. Hence, it shall be subdivided to 400 pairs of image for each quadrant. The same subdivision shall also done on the right breast.

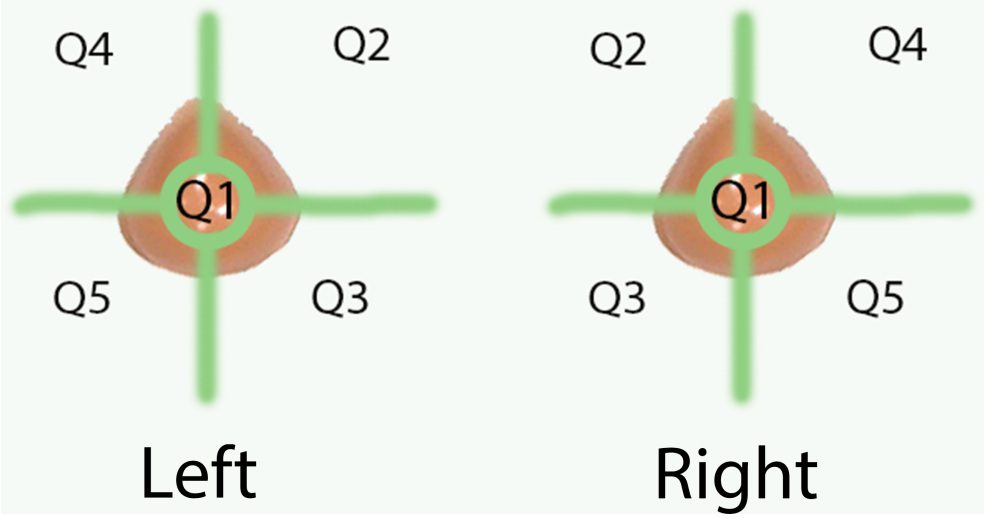


Figure 4-7. Quadrants of the Breast

After doing step 1 and 2, the images still contains the torso of the women. It has only partition it into quadrants. But it has not yet extracted ROI of each images. Step 3 and 4 shall do the proces of ROI extraction. Step 3 means create a box Mask sequence for each quadrant. The creation of box Mask sequence is discussed and visualized in Section 4.4.5. Using the box Mask sequence, extract ROI image of each Frame (Step 4). It should be noted that the raw image is 640 x 480, but the ROI image shall be, for example, 80 x 80 pixels. In other words, the output of this subsection is an 80 x 80 image from 640 x 480 image.

4.4.7 Finger Mask Sequence

The Finger Mask Sequence was done by manually annotating the fingers in each frame. Unlike in section 4.4.5 wherein the annotation is a rectangular box, the annotation here is extracting only the three main fingers (i.e. index, middle and ring) up until the second joint of each finger. Hence, the annotation is an irregular shape.

In this subsection, we used an interactive Video Editing tool called Adobe After Effects CS6, specifically, the Roto brush tool [93]. The tool has an ability to track an irregular shape set by the user. If the predicted closed shape is not very accurate for a given frame, the mask can interactively be edited so that the predicted closed shape is what the user desires for that frame.

Similar to Box Mask Sequence, the output in this subsection is a black and white image sequence. The white pixels indicate that it is a pixel of the fingers while black pixels indicate it is otherwise. This sequence will become useful in the next section where it discusses how to calculate a singular depth value from a given depth Image.

4.4.8 Creating Ground truth

For clarity, the author defines ground truth as the singular depth value of the palpation at the current frame. It should be noted that extracting ground truth straightforward from Kinect is impossible as Kinect only provides the Depth Map of the torso. Averaging the Depth Map from the ROI discussed in the previous subsection is prone to error as the ROI contains unpalpated area of the breast. Calculating the ground truth should only be done at the area of the finger so that it only contains pixels that change value when palpating. That is, it does not contain the unpalpated area where in the depth change is not proportional to depth change by the finger area. Hence, to properly extract the singular depth value (ground truth), this subsection utilizes Finger Mask Sequence (Section 4.4.7) rather than Box Mask Sequence (Section 4.4.5).

As discussed in Section 3.5, every pixel of the Depth Map contains the distance between the RGBD camera and the real world point for that coordinate. In other words, the depth Map contains absolute distances in mm (i.e. Kinect for Xbox 360). So, to extract the depth values of the finger, mask the

Depth Map using Finger Mask Sequence. The masking operation is discussed in more details in the next paragraph.

Afterwards, ground truth is the median value among the depth pixel from the masked Depth Image. The pseudocode used for "scalarization' is shown in the figure below. In the first two lines, it requires the input image matrix Im and FingerMask. Im is the depth Image of the current frame. The FingerMask matrix is the corresponding FingerMask frame from the Finger Mask Sequence. The third line is the extraction of specific depth values of the depth Image using the FingerMask. Recall that the pixel values of FingerMask is either 1 or 0. It contains 1 if it is a pixel of the finger area while it contains 0 if it is otherwise. So, a simple element-wise multiplication will extract the depth values from the depth Image Im. The resulting image Matrix shall be assigned to ImNew. Line 4 removes the zeros results of ImNew by assigning all nonzeros values of ImNew to SparseIm. Lastly, Line 5 is the calculation of the median value.

```
[1]Input Image Matrix Im
[2]Input Image Matrix FingerMask
[3]Set ImNew as the element-wise multiplication of Im and FingerMask
[4]find non-zero elements of ImNew and assign to SparseIm
[5]return the median of SparseIm
```

Figure 4-8. Calculation of Nonzero Median Pseudo Code

4.5 Quantification of Depth Level

In section 4.4.6, it has shown manual segmentation of the RGBD Dataset, which is needed to partition different situations. In section 4.4.7, a manually annotated finger Mask Sequences for all available quadrants in all cup sizes was created for extracting the finger location for each frame. The finger

Mask sequence is not enough to know the current palpation because this research wants to determine whether its low, medium and deep depth level not its scalar depth. This section provides the last post-processing needed in the RGBD dataset.

To quantify the breast palpation, the depth is quantized using a uniform "fuzzy-like" relationship. Low and deep is modeled as semi-trapezoidal relationship while medium has a triangular membership function as shown in Figure 4-9. After calculating the membership of the ground truth, the one with the highest membership value will be its depth level.

A breast can be visualized as a paraboloid subject to gravity. Hence, there will be different minimum and maximum values for each quadrant. Furthermore, MIN and MAX values will also vary depending on the cup size, which was defined in Table 3-4.

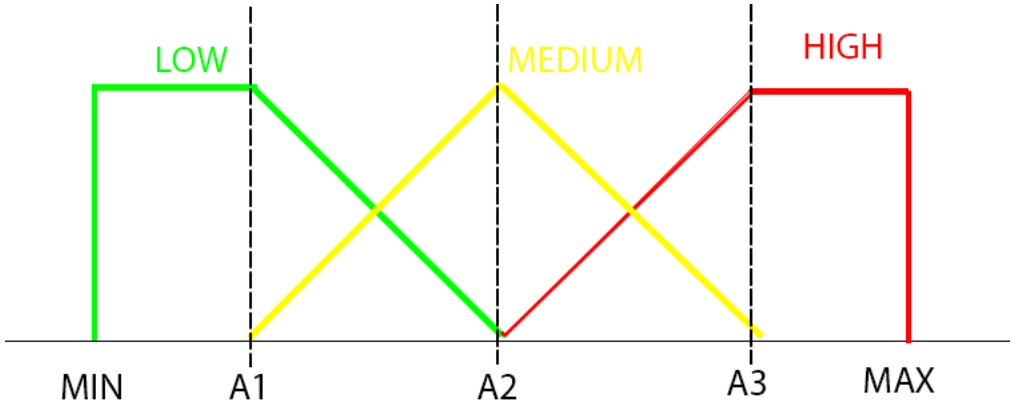


Figure 4-9. Model used for depth quantization

where:

$$A1 = 0.25 * (MAX - MIN) + MIN \quad \textbf{Equation 4-1}$$

$$A2 = 0.50 * (MAX - MIN) + MIN \quad \textbf{Equation 4-2}$$

$$A3 = 0.75 * (MAX - MIN) + MIN \quad \textbf{Equation 4-3}$$

So, the membership model is only dependent on the variables MIN and MAX. They are obtained by first extracting the ground truth (i.e. as discussed in the previous subsection) of all training for one quadrant. Then find the minimum and maximum values.

4.6 Feature Extractions Schemes

4.6.1 Normalized Shadow Area

In order to extract the normalized shadow area, this study would like to utilize a shadow segmentation algorithm. Afterwards, normalized shadow area is calculated by dividing total pixel count of the shadow output and total pixels of the ROI frame.

The process of shadow segmentation is similar to the works of Salvador et al[69] which is summarized via flowchart as seen in Figure 4-10. It starts by grabbing the current frame and the reference frame. The reference frame is the frame wherein the object is not yet seen in the image. For this study, that means it is the frame wherein the subject have not started palpating the breast.

After grabbing the current and reference frame, convert these two images into C1C2C3 color space using the set of equations from section 3.7.3. Using the RGB image of the current and reference frame, calculate their absolute image difference. Afterwards, do some post-processing. In this study, it is averaging filter and image thresholding. Using the C1C2C3 images of the current and reference frame, Calculate the edges using Canny edge detector in the C1C2C3 image of the current frame. Then, use the morphological open operation for post-processing.

The last step is using the results of both RGB frames processing and C1C2C3 frames processing, classify the shadows whether it is a self-shadow or

a cast shadow. However, for this study, this last step is bypassed so that it will use all shadow area.

This algorithm works only on frame sequence. At start, there should be a frame wherein the desired object (i.e. in this study, the fingers) is not seen. In Breast Self-Examination, reference frame can easily be grabbed during the start of palpation proper as there is visual inspection step where they will simply raise their arms upwards and check for discoloration and irregularities.

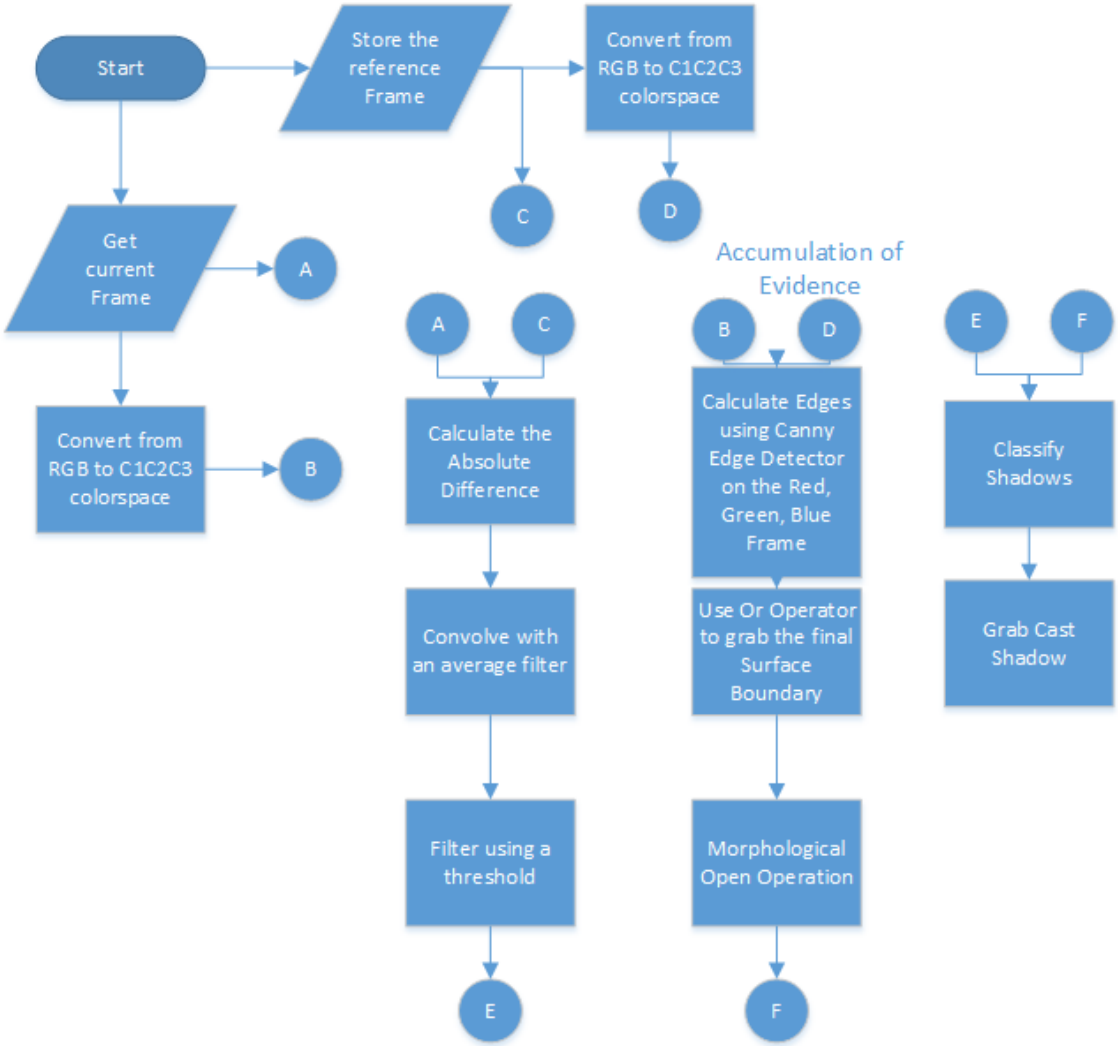


Figure 4-10. Shadow Segmentation Flowchart

4.6.2 Image Entropy

Image Entropy as commonly used in image processing as a measurement of randomness. It is commonly used in texture analysis to identify how random is the data presented [57]. In the works of Chen et al[19], it was used to measure the RGB image difference from a reference frame. The feature extraction flow is shown in Figure 4-11.

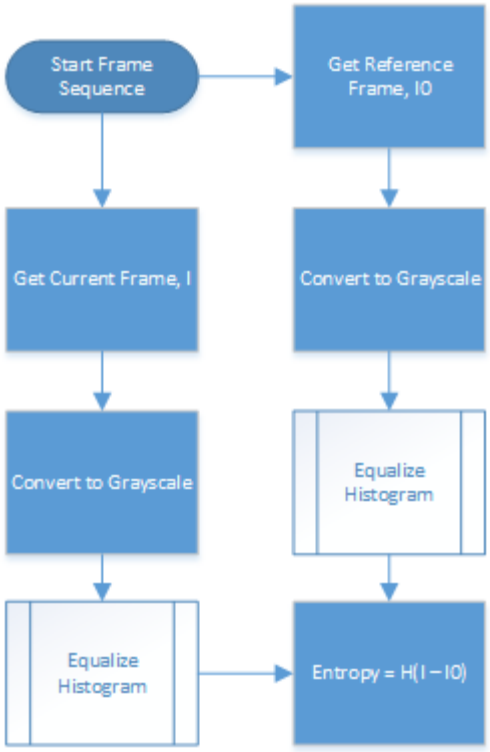


Figure 4-11. Image Entropy Feature Extraction Flow chart

4.6.3 Laws' Textures Histogram

The implementation of the Laws' Histograms Feature Extraction is shown in Figure 4-12. As seen from the figure, the implementation can be broken down into three main stages. In stage 1, the image is first extracted from the PNG sequence of a BSE video. It is assumed that the initial color space is RGB. Afterwards, following the works of Saxena[33]–[35], the RGB image is

converted to YCbCr. In this study, only the intensity channel will be used. That's why the last step of this stage is to extract the luma Image.

Stage 2 and 3 are about extracting the Texture Map. Together, they both handle the kernels with symmetric pairs or not. Stage 2 processes the texture Map of the first symmetric kernel. It first creates an iterator for the Laws' kernel. Then convolves it with Luma intensity Image. Afterwards, it will check if it contains a symmetric pair or not. If it does, it will go to stage 3. If not, it will skip stage 3 and immediately do the stage 4 processes. It should be noted that the sequence of kernel in the flowchart follows the sequence from Table 3-1 and Table 3-2. As an example, the Luma Intensity Image is 150 x 100 and the Laws' kernel at iterator i is L5E5. Now, it will be convolve to the intensity image, which provides a texture Map that has 100 x 100 dimension. Now, since it has a symmetric pair (i.e. E5L5), then it shall go to the on-page connector B that is connected to stage 3. However, if laws' kernel at iterator I is E5E5, it doesn't contain any symmetric pair. Hence the processed texture Map due to the convolution of intensity image and E5E5 shall be the input for stage 4.

Stage 3 processes the texture Map of the second symmetric kernel and combines it with the prior texture Map. Again, the flowchart shall only go to stage 3 if it contains a symmetric pair like L5E5, R5E5 and S5E5. If observed properly, stage 3 is similar stage 2 but only with some additional process. After producing the texture Map from Stage 2 and texture Map from stage 3, the texture maps shall be combined using pixel-wise averaging. The output of this averaging shall be the input for stage 4.

Lastly, stage 4 processes final histogram vector for each Law's kernel. Note that the input texture Map for stage 4 is a full image. Meaning, it contains the dimensions the input image. Now, the number of bins is hardcoded by the user. Varying the number of bins may affect test accuracy of the algorithm.

Stage 4 also handles some post-processing scheme and concatenates it to the 9 Laws' final feature vector. So, If the number of bins for each histogram is called *nbins*, then, at the end of the feature extraction, the final feature vector should have 9**nbins* x 1 dimensions.

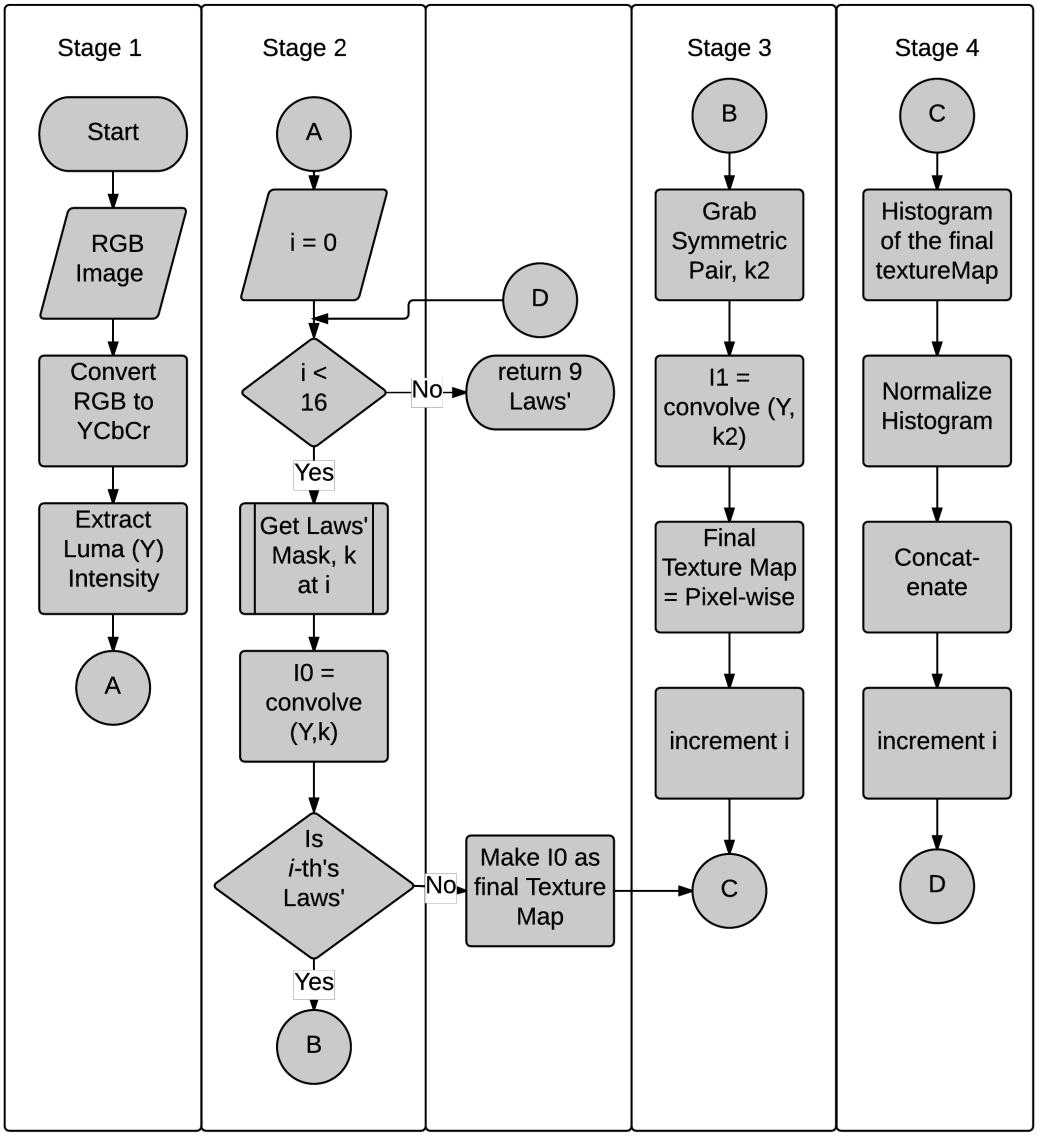


Figure 4-12. Laws' Features Extraction

4.6.4 Local Binary Pattern Global Histogram

The process of extracting Local Binary Pattern (LBP) Global Histogram is very simple, which is shown in Figure 4-13. The first step is to convert the input image in grayscale color space using the equations from Section 3.7. Afterwards calculate the LBP texture Map using the discussion from Section 3.8.2. The size of the texture Map should be a full Image. Afterwards, compute the image histogram of nbins with normalization.

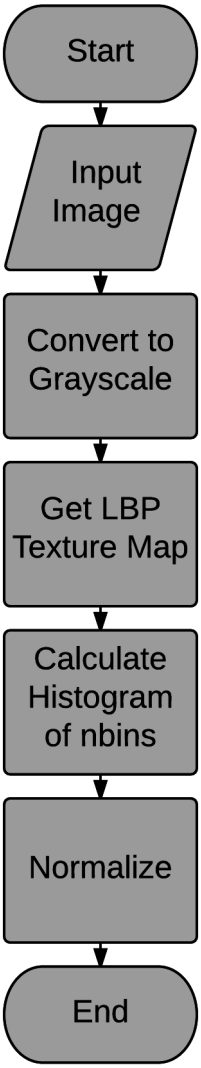


Figure 4-13. LBP Histogram Feature Extraction Scheme

It is notable that the moving window for LBP can be larger than 3x3 window. Moreover, there has even been an extension to the classic LBP as done by Ojala[94]. However, in this study, only the classic 3x3 LBP is utilized. Similar to the previous section, varying the nbins may vary the accuracy for LBP histogram.

4.7 Machine Learning Algorithms

In this study, multiple Machine learning is experimented to obtain the model that provides the best behavior.

4.7.1 Linear Regression

As instructed by the study of [19], a linear regression was used in estimating depth level. Concretely, in this study, we implemented it by:

1. Find the reference frame
2. Find the entropy value of each frame of the training set as discussed in section 4.6.2.
3. Use the output of section 4.4.7 to use as the ground truth depth.
4. Evaluate the model using the test set.

This model is only implemented because this thesis wishes to find the quantitative accuracy of Chen et al[19] especially to the Kinect RGBD dataset.

4.7.2 Support Vector Machine

The training scheme for SVM statistical Model is shown in Figure 4-14. It is notable that since there are different training sets for each quadrant of each cup size, there will be different SVM model for each quadrant of each cup size. The process starts by extracting the RGB and Depth frames, and Box and Finger Mask Frames from the RGBD dataset. The process of extracting these four frames is discussed in Section 4.4.4, 4.4.5 and 4.4.7. Using RGB and Box Mask frame, and Depth and Box Mask frame, extract the RGB region of interest

(ROI) image and Depth ROI image as discussed in section 4.4.6, respectively. Afterwards, process the RGB ROI by doing some preprocessing and extracting the features. These features can be any feature extractions schemes discussed in Section 4.6: Normalized Shadow Area, Image Entropy, Laws' Textures Histogram, and Local Binary Pattern Global Histogram. After the feature extraction, normalize the feature vector. In this study, only L1 normalization is utilized.

In processing the depth ROI, the goal here is to convert the absolute depth Map into the class label, which can be read by the Machine Learning Model. This process is done by first following the steps from Section 4.4.8 and 4.5. In summary, when the Depth ROI is processed as discussed in Section 4.4.8, it will produce a singular absolute depth value of the current palpation. When this absolute depth value is processed (or fuzzified) as in section 4.5, it will produce low, medium and deep depth level. Now, after grabbing the depth level, the next step is to simply format it so that the SVM model can read it.

The SVM implementation utilized is the implementation of OpenCV. OpenCV also automatically finds the parameter values and kernel methods. This is done by using 10% of the training set as the validation set. And the parameter value is considered optimal when validation error reaches the global minimum.

A sample input data is shown in Figure 4-15. This is an image from the cup B, left breast, quadrant 2 dataset. The top left is the RGB image and the mask (i.e. black & white image) beside it is its corresponding box Mask Sequence. The colored image on the top right is the corresponding Depth Image shown in HSV colormap. The mask image beside the latter is the corresponding finger Mask Sequence.

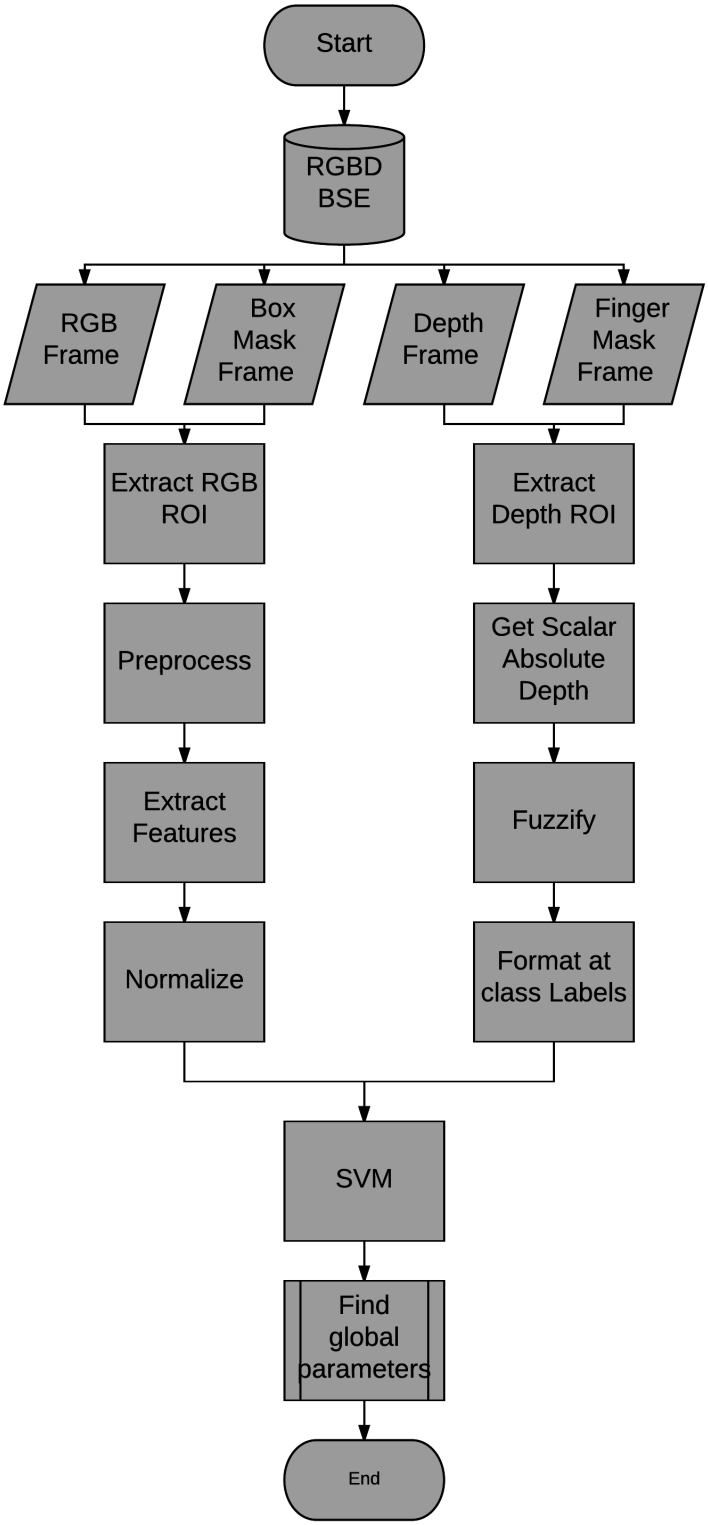


Figure 4-14. SVM training scheme

As stated in Figure 4-14, the next step for the RGB image and the box Mask image is extract RGB ROI. Afterwards, it will be preprocessed. For the Depth Image and finger Mask Image is to extract Depth ROI. In depth ROI, only pixels with colors orange are the one to be further processed. The pixels in blue are left out and shown only for visual appearance. Now, the depth ROI will be processed to extract the scalar Absolute Depth value and its value is 766 in this example. Then, it will be fuzzified. In this quadrant, this value shall have a depth level of Low.

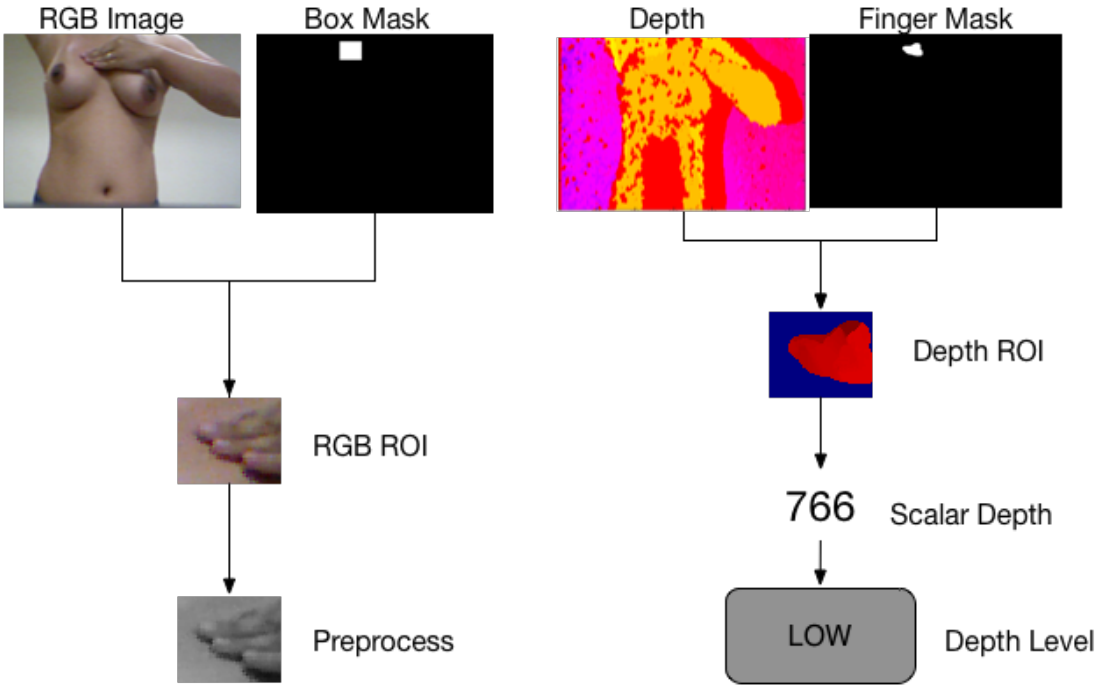


Figure 4-15. Sample Image and Depth Level Extraction

Figure 4-16 shows the features to be inserted to the SVM. The target depth level is LOW. The Feature Extraction schemes utilized here is the combination of Laws' Textures Histogram and Local Binary Pattern Global Histogram. Now, as stated in section 4.6.3, when using Laws' Histogram, it shall provide a total of nine (9) Histogram. While as discussed in section 4.6.4, the LBP histogram shall provide only one (1) histogram. Therefore, a total of 10

histograms is feeded to the SVM. These histograms are shown in the leftmost part of Figure 4-16. Each histogram of the Laws' Histogram is assumed to have 39 bins while the histogram of the LBP has 41 bins. Moreover, every bin of each histograms shall be the input of the SVM. Hence, each Laws' Histogram provides 39 input features and the LBP histogram provides 41 features. Now, in the perspective of Figure 4-14, the extraction of Laws' Histogram and LBP histogram is considered only one process called Extract Features. Since, in this example, two feature extraction schemes are utilized, these two features shall be concatenated to become one feature vector to the SVM. The total number of features is now 39*9+41 = 392 features.

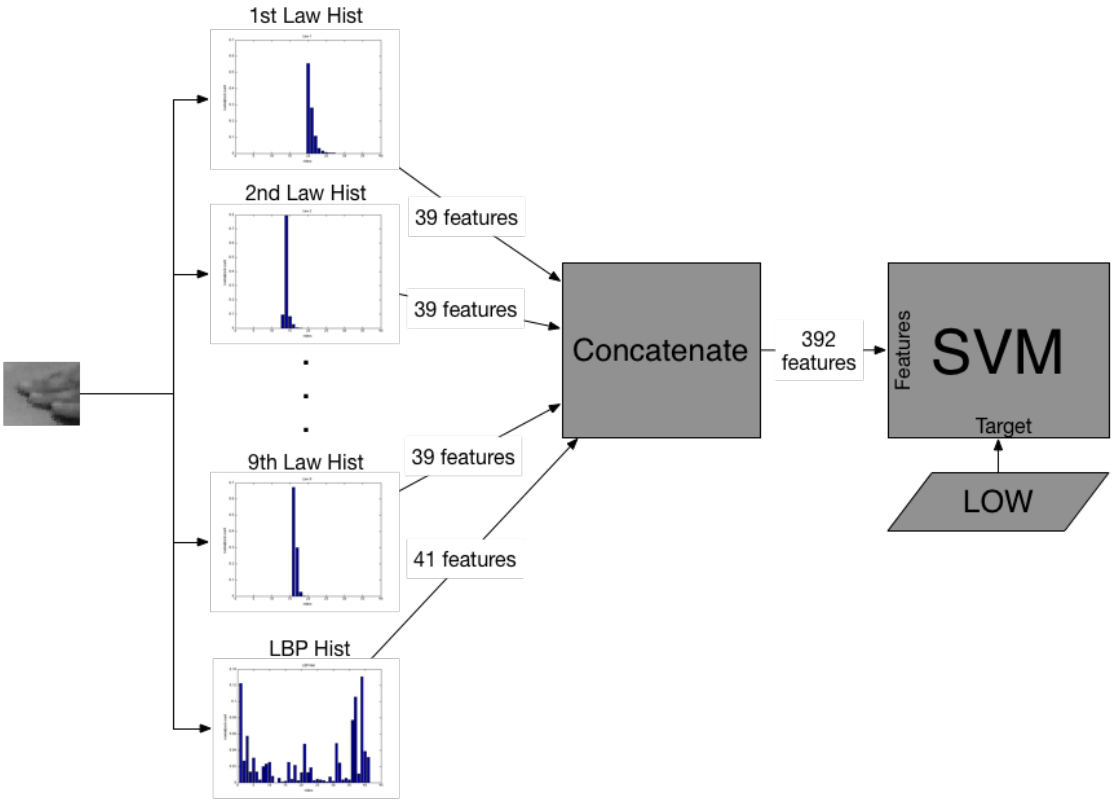


Figure 4-16. SVM sample Input

4.7.3 Gradient Boosted Trees (GBT)

The implementation is shown in Figure 4-17. The process for utilizing Gradient Boosted Trees is similar to Support Vector Machine (4.7.2). However,

its main difference is in the configuration of the statistical model. In the SVM process, the optimal parameters like C and gamma parameters (see section 3.9.1 for discussion) are automatically found by OpenCV. Hence in relation to its implementation, it is unnecessary to manually find the optimal parameters for the given dataset. However, for GBT, there is no automated finding of optimal parameter. Rather than finding the optimal parameters for each training set, the parameters are set uniformly for all datasets. It is shown in the Table 4-1. The rationale here is that it is more important to determine whether this prediction model is better than any other models than making this model optimal.

Table 4-1. Gradient Boosted Trees Parameters

Parameter Name	Value
Weak_count	100
Shrinkage	0.1
Subsample Ratio	1.0
Max_depth	2.0
Use_surrogates	False

4.7.4 Artificial Neural Network

The training scheme is almost similar to section of Gradient Boosted Trees (GBT). From the RGBD dataset, grab the image and depth. Grab also the corresponding box mask frame and finger mask frame. Using the image frame and box Mask frame, extract image ROI. Preprocess this image ROI. Extract any of the features listed in Section 4.6. Then, the set of normalized features shall be the input feature vector for ANN.

Using depth frame and the corresponding finger Mask frame, extract depth ROI. Then create the ground truth as instructed in section 4.4.8. Then quantize the absolute depth value as instructed in section 4.5. The formatted ground truth labels shall be the target class of the ANN training.

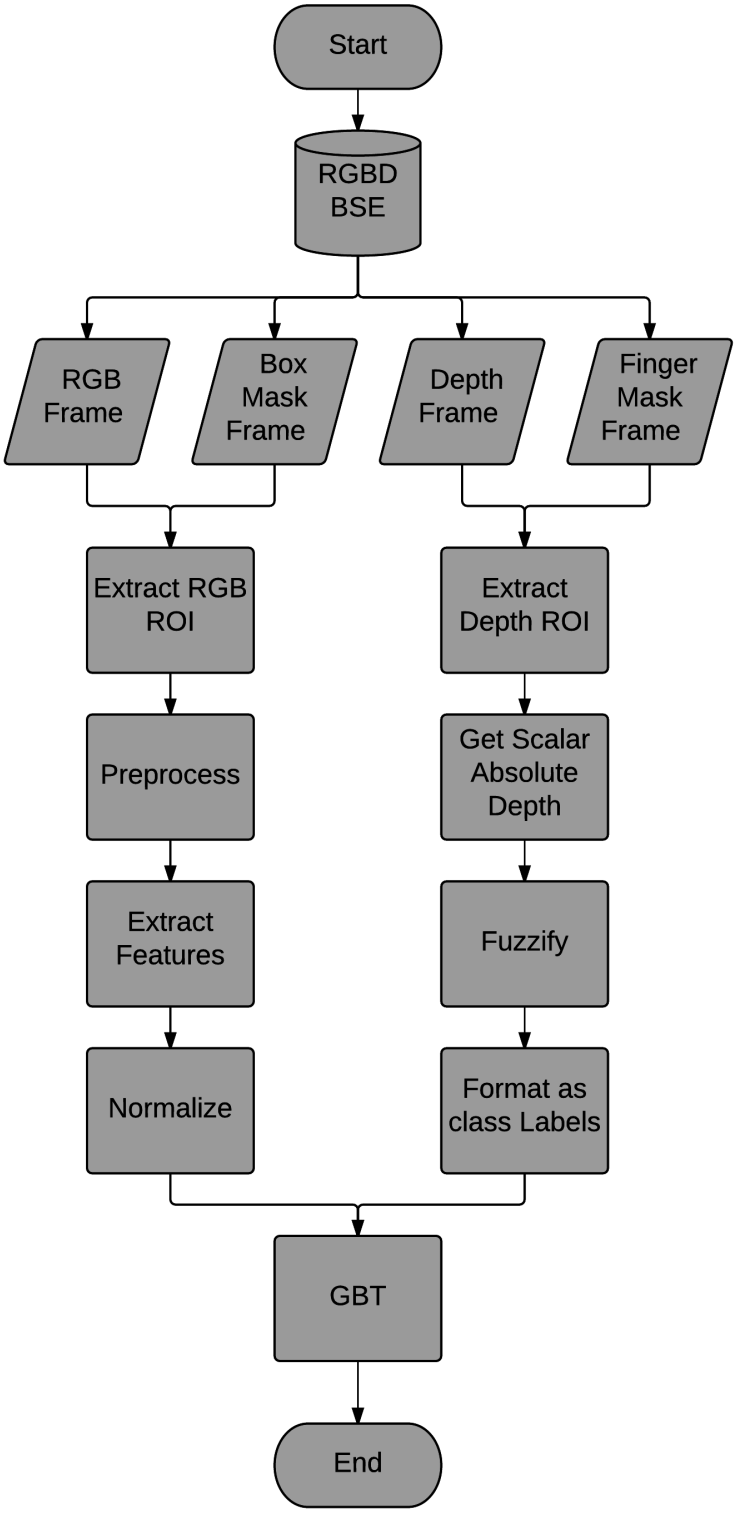


Figure 4-17. Gradient Boosted Trees Training Scheme

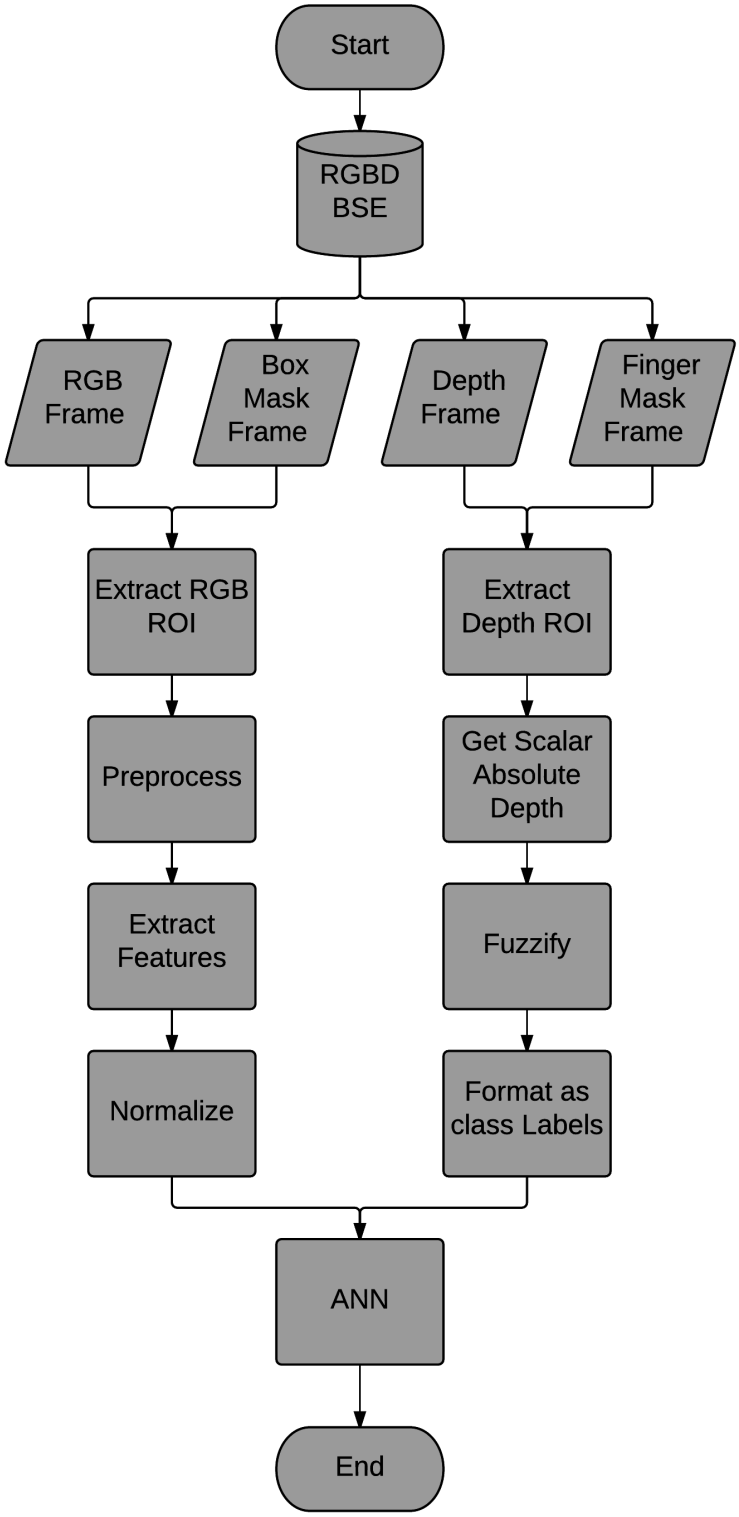


Figure 4-18. Artificial Neural Network (ANN) training scheme

Now, type of ANN utilized is the multi-layer perceptron with back-propagation scheme and one hidden layer. The parameters used for the training scheme is shown in Table 4-2. These are the parameters used in all ANN training.

Table 4-2. ANN parameters

Parameter Name	Value
Hidden layer size	40
Activation function	Sigmoid function
Weight of gradient term	0.1
Weight of moment term	0.1

4.8 Accuracy Assessment

In order to assess the accuracy of the algorithm, it is vital that the RGB and depth images used for training the prediction model as seen in section 4.7 should not be used again for assessing the algorithm's accuracy. The whole dataset shall be partition into two as in Figure 4-19. The first partition is called the training set and the second partition is the test set. The training set is the RGB and Depth Image Frames used for training any of the Machine learning Algorithms discussed in section 4.7. The test set shall be used for assessing the overall accuracy of an algorithm. 85% of the whole dataset shall be alllocated to the training set and 15% shall be for the test set.

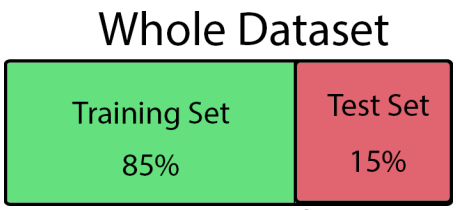


Figure 4-19. Partitions of the Dataset

Ensuring the reliability of the algorithm, it is imperative to quantify the error of the algorithm. The main evaluation scheme used is the Confusion Matrix. This metric is normally used in classification problems. It is used to measure whether the classifications are true positive, false positive, true negative, or false negative in a compact table form. A sample confusion matrix

is shown in Figure 4-20. The boxes in green are the accurate classifications. So in the first green box, it states that the desired (i.e. target) class is low, and the algorithm used classified that item into low (i.e. output). The red boxes are the misclassified data. If seen along the row, the red boxes are the true negatives. In row 1, the algorithm classified it as class *low* however the desired class should be either medium or high class. If seen along the column, these errors are the false positive. In column 1, the desired class should be *low*, however the algorithm classified it as either medium or high. It is also notable that the percentages contained in either red or green box means the percentage of that data to the whole sample set.

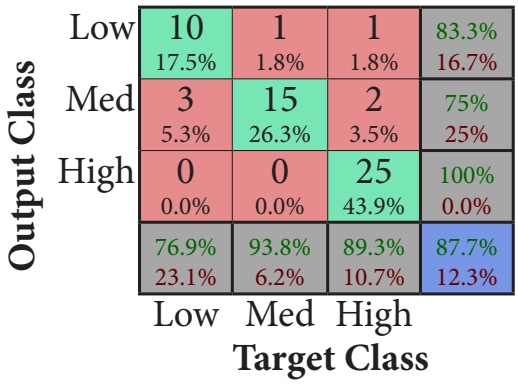


Figure 4-20. Sample Confusion Matrix

In the same Figure 4-20, there are still gray boxes and a blue box that has not been discussed. The upper text in the blue box is the overall accuracy of the whole three-by-three table. It is the major evaluation assessment used in this study. The lower text in the blue box is the misclassifcation rate. It is the complement of the overall accuracy. To compute the overall accuracy, using confusion matrix,

$$Accuracy = \frac{Sum\ of\ diagonals}{sum\ of\ all\ elements}$$ **Equation 4-4**

Other supplementary performance assessment like true negative rate of each class and false positive positive of each class is shown below.

$$True\ Negative\ rate\ of\ class\ k = \frac{diagonal\ element\ of\ row\ k}{sum\ of\ row\ k}$$ **Equation 4-5**

$$False\ Positive\ rate\ of\ class\ k = \frac{diagonal\ element\ of\ col\ k}{sum\ of\ col\ k}$$ **Equation 4-6**

Using Figure 4-20, the gray boxes in the fourth column is the true negative rate. For instance, row 1 column 4 is the true negative rate of class LOW. Its value is obtained by taking the diagonal element (10) and dividing it with the sum of the whole row (12), which is 0.833 or 83.3%. Now, the percentages in red in each gray boxes is simply the complement of its respective rate. For instance in row 1 column 4, the complement of the true negtive rate of class 1 is simply $1 - 0.833 = 0.167$ or 16.7%.

CHAPTER 5
Results

5.1 Gathering of Database

5.1.1 Kinect Database

The environment used is a conference room of De La Salle University. Figure 5-1 shows sample sequence recorded using Kinect for xbox 360. In this figure, the first pair of sequence is the cup A, the second pair of sequence is the cup B, and the third is the cup C. Now, The first of each pair is the RGB image sequence. As discussed in RGBD Images (Section 3.5), this is the usual videos captured by normal cameras. The second sequence of each pair is the Depth Images displayed in HSV colormap. Originally, this is a grayscale image. But it is converted and scaled to HSV in this document for more visual acuity. It should be emphasized that the Depth Image seen in HSV colormap is not the used for the program. Rather, this HSV depth image is utilized only for better display of depth Map. In the actual code, the 16-bit grayscale depth images are used for further processing like in Quantification of Depth Level (4.5), Feature Extractions Schemes (4.6), and Machine Learning Algorithms (4.7).

The camera setup of the Kinect is discussed in Section 4.4.3. As stated in the said section, the distance of the Kinect from the subject is between 0.6 and 0.8.

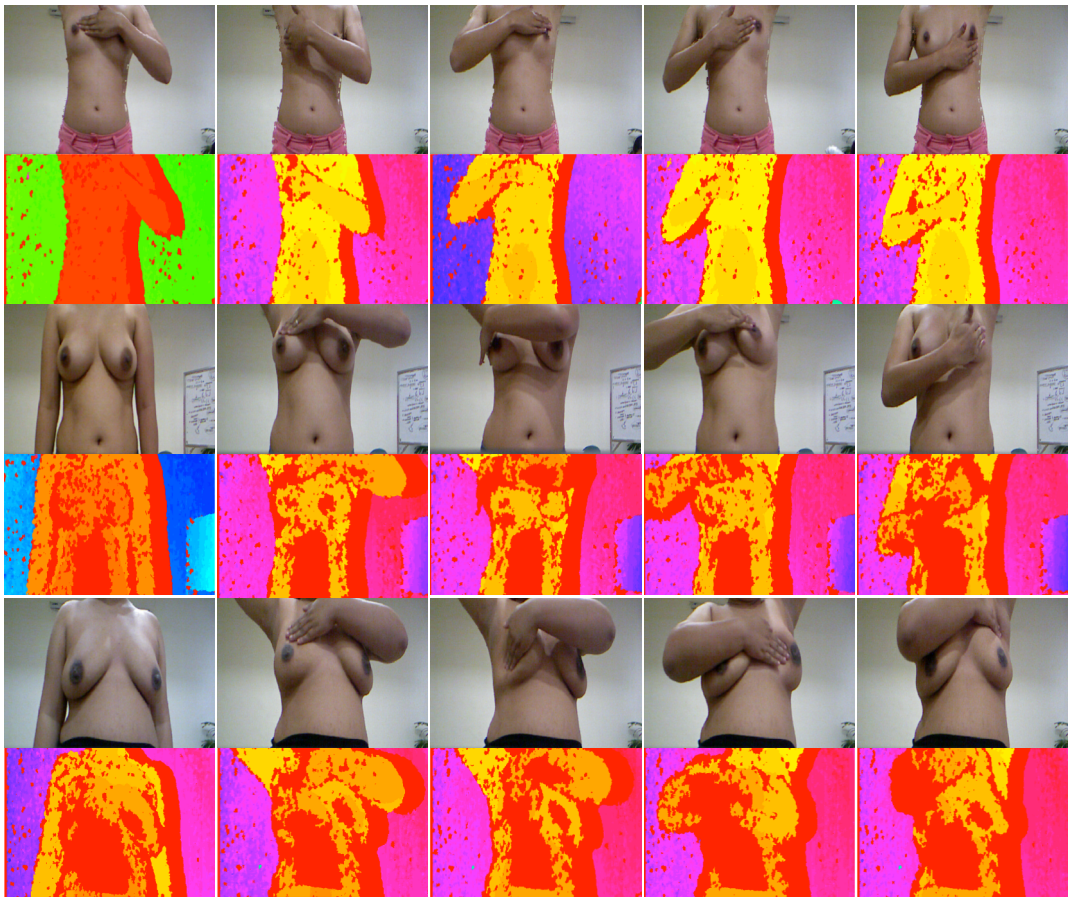


Figure 5-1. Kinect Database

5.1.2 Segmentation of each cup size to the proper quadrant

Given the dataset above, it is manually segment based on Section (1). However, as it will be shown in the next section, for each breast, only its quadrant 2 and quadrant 3 is obtained. Afterwards, for each quadrant, 15% will go to test set and 85% to the training set. The number of frames for the training and test set is shown in Table 5-1.

Table 5-1. Number of Training and Test Images for each Quadrant

Number of Frames	Cup A		Cup B		Cup C	
	Training Set	Test Set	Training Set	Test Set	Training Set	Test Set
Left_Q2	101	18	142	25	61	11
Left_Q3	120	21	113	19	85	15
Right_Q2	105	18	120	21	75	13
Right_Q3	99	18	108	19	81	14

5.1.3 Box Mask Sequence

A sample sequence of the box Mask is shown in Figure 5-2. Note we wish to achieve the criteria specified in section 4.4.5. This box Mask is created using the Rectangular Tool and smart object. However, Photoshop cannot make the original resolution better. The image resolution of the Kinect is 640 x 480. And the breast palpation is just small fraction with the image. That is the reason why the following image sequence seems blurry in nature.

Figure 5-2. Box Mask Sequence for Quadrant 2 cup B

As stated in the said criteria as in Section 4.4.5, it has been stated that the resolution of box Mask should be uniform for each quadrant. Table 5-2 provides the final resolution of the RGB image ROI on each quadrant for each cup size.

Table 5-2. Box Mask Sequence Resolutions

Box Resolution	Cup A	Cup B	Cup C
Left_Q2	38 x 33	49 x 59	68 x 58
Left_Q3	46 x 30	77 x 55	74 x 43
Right_Q2	44 x 47	67 x 61	56 x 37
Right_Q3	46 x 38	84 x 55	60 x 39

Figure 5-3 provides an example of occluded fingers that cannot meet the specified criteria. The frame sequences cannot provide the finger texture and pressed area. It frequently happens at Q5 and Q4 on all cup sizes. Since there are no current literature that tackles depth / pressure estimation in an actual recording yet, this study focused on obvious palpation which is, specifically, palpation in Quadrant two and three of left and right breast.

Figure 5-3. Occluded Fingers in Quadrant 5 cup B

5.1.4 Finger Mask Sequence

A sample Finger Mask Sequence is shown in Figure 5-4. The enclosed area by the pink line is the region of interest. As discussed in 4.4.7, it is desired to enclose the three main fingers up until the second joint since these are main sources of depth level changes. As it can be seen, the three fingers are not perfectly captured in every frame. In the actual annotation, rather than perfectly grabbing the entire desired region, the main priority is to minimize the true negatives. In other words, as long as the enclosed desired area is sufficiently large and minimizes the breast pixels, the mask is good enough.

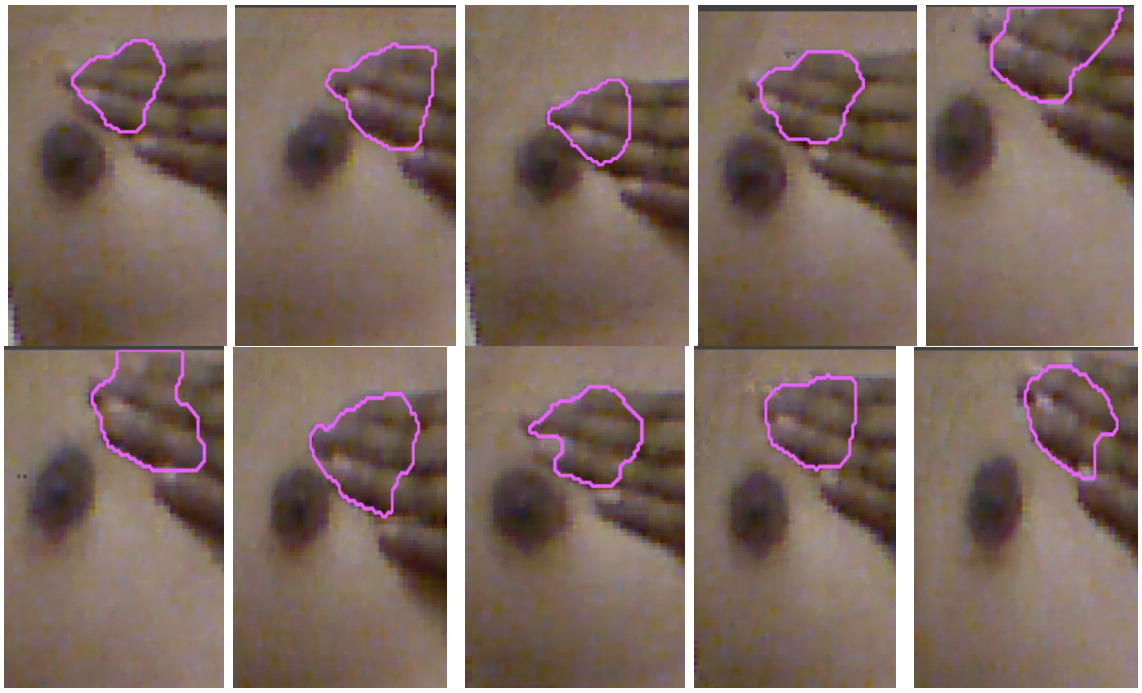


Figure 5-4. Finger Mask Sequence in cup A Q2

5.1.5 Quantification of Low, Medium and Deep

This section provides the results for quantification of low, medium and deep palpation for the ground truth. As stated in Section 4.5, the procedure done is iterating over the training set and finds the minimum scalar depth value and maximum scalar depth value. The resulting values are shown in Table 5-3. Again, the variable MIN are MAX are utilized to characterize the fuzzy model parameters for each quadrant. These variables enable the fuzzification of the ground truth into low, medium or high state. The values of A1, A2 and A3 can be calculated using Equation 4-1, Equation 4-2, and Equation 4-3 and is also shown in Table 5-3. Using two-point form equation of a line, one can easily find the characteristic equation of the slanted lines of the fuzzy model parameters.

Now, having calculated variables A1, A2 and A3, calculating the range of each depth level is straightforward. The final ranges is shown in Table 5-4. It should be noted that these ranges are inclusive to the left, and exclusive to right, that is [L,R). These ranges are found by first observing the membership

function of the fuzzy model as seen in Figure 5-5. Whenever there are overlaps between two membership, its fuzzified model shall be the one that provides the highest membership value. Consequently, it means that the range of Low shall be from the value of MIN to the middle of A1 and A2; the range of Medium shall be from the middle of A1 and A2 until the middle of A2 and A3; the range of High shall be from the middle of A2 and A3 until the value of MAX.

Table 5-3. Extended Fuzzy Parameters

Quadrant	MIN (mm)	MAX (mm)	A1 (mm)	A2 (mm)	A3 (mm)
Cup A					
Left_Q2	762	794	770	778	786
Left_Q3	744	771	750.75	757.5	764.25
Right_Q2	772	790	776.5	781	785.5
Right_Q3	771	801	778.5	786	793.5
Cup B					
Left_Q2	607	678	624.75	642.5	660.25
Left_Q3	603	617	606.5	610	613.5
Right_Q2	619	684	635.25	651.5	667.75
Right_Q3	614	658	625	636	647
Cup C					
Left_Q2	568	609	578.25	588.5	598.75
Left_Q3	563	607	574	585	596
Right_Q2	591	668	610.25	629.5	648.75
Right_Q3	597	673	616	635	654

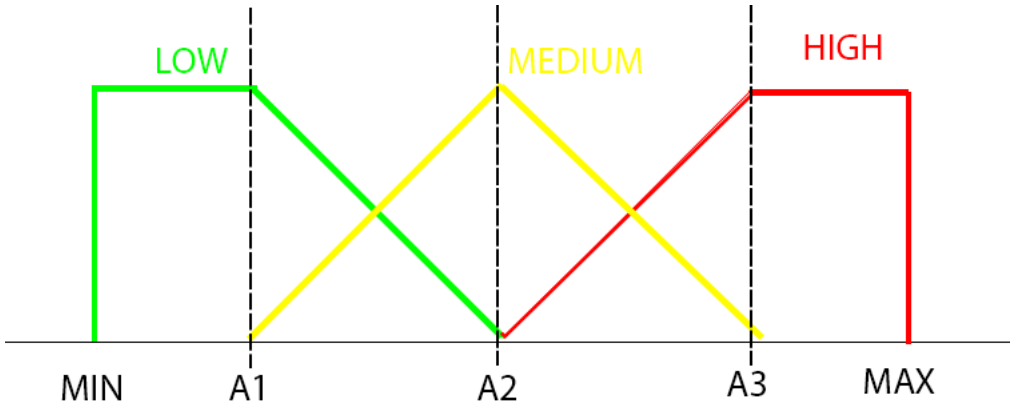


Figure 5-5. Fuzzy Model

Table 5-4. Range of Low, Medium and High Pressure Level

Quadrant	LOW (mm)	MED (mm)	HIGH (mm)
Cup A			
Left_Q2	762 - 774	774 - 782	782 - 794
Left_Q3	744 - 754.125	754.125 - 760.875	760.875 - 771
Right_Q2	772 - 778.75	778.75 - 783.25	783.25 - 790
Right_Q3	771 - 782.25	782.25 - 789.75	789.75 - 801
Cup B			
Left_Q2	607 - 633.625	633.625 - 651.375	651.375 - 678
Left_Q3	603 - 608.25	608.25 - 611.75	611.75 - 617
Right_Q2	619 - 643.375	643.375 - 659.625	659.625 - 684
Right_Q3	614 - 630.5	630.5 - 641.5	641.5 - 658
Cup C			
Left_Q2	568 - 583.375	583.375 - 593.625	593.625 - 609
Left_Q3	563 - 579.5	579.5 - 590.5	590.5 - 607
Right_Q2	591 - 619.875	619.875 - 639.125	639.125 - 668
Right_Q3	597 - 625.5	625.5 - 644.5	644.5 - 673

5.2 Quantitative accuracy of Chen et al's Pressure Estimation Algorithm

The current state-of-the-art depth level estimation algorithm for BSE is from the works of Chen et al [19]. However, as previously stated, although the study provided the results for its effectiveness, they did not attempt to quantize it into low, medium and deep.

Hence, there was a need to re-implement the code and so that it can be properly benchmarked. The model to be used is the Image Entropy with linear regression as outlined in Section 4.7.1. It is re-implemented by first extracting

the entropy features as discussed in Section 4.6.2. Afterwards, train the linear regression model integrated with the Fuzzy Breast Model as discussed in section 4.7.1. Then, predict the test results using the test set. The resulting training and test accuracy for each of the available quadrants and cup size is shown in Table 5-5.

Table 5-5. Entropy Training and Test Accuracy

Entropy	Training Accuracy			Testing Accuracy		
	Cup A	Cup B	Cup C	Cup A	Cup B	Cup C
Left_Q2	44.60%	55.60%	24.60%	16.00%	39.30%	36.40%
Left_Q3	46.70%	23.80%	24.80%	21.10%	23.50%	20.00%
Right_Q2	44.60%	55.60%	40.00%	38.10%	30.70%	7.70%
Right_Q3	46.70%	23.80%	50.00%	42.10%	29.60%	21.40%

In all classifiers, it is intuitive that a trained classifier should at least be better than random selection. If not, then it is better not to use any model at all. The classifier for depth classification is a 3-class classification. Whenever there are three choices with only one correct answer and the selection process is at random, the probability to choose the correct answer is 0.3333. Hence, for a 3-level classifier, the minimum (baseline) accuracy should be 33.33%.

The average training and testing accuracy of the said algorithm in the whole dataset is plotted in Figure 5-6. Moreover, the baseline accuracy is also plotted for comparison. The blue columns are entropy accuracy while the red columns are the baseline accuracy.

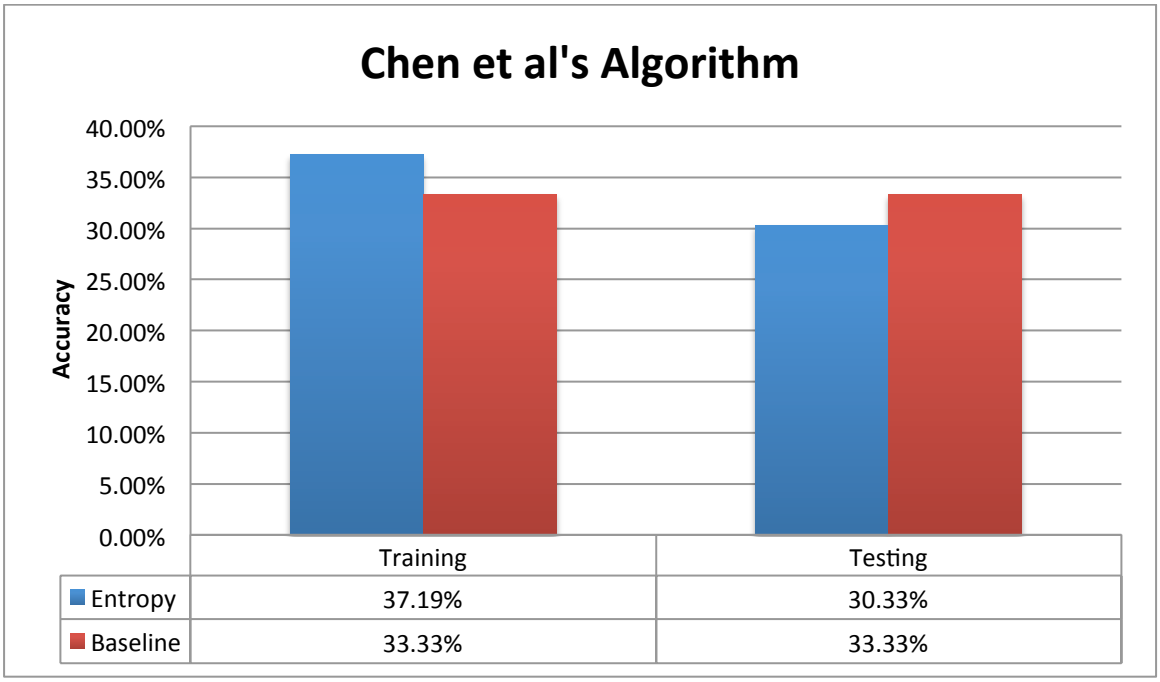


Figure 5-6. Chen et al 's quantitative accuracy

5.3 Normalized Shadow Area

The goal of this section is to provide results similar to section 5.2. However, rather than using the image entropy as features, the area of shadow segmentation as discussed in section 4.6.1 will be extracted. The prediction model is still the linear regression integrated with Fuzzy Breast Model as discussed in 4.7.1 so that this feature extraction scheme can be fairly compared to the results of the previous section. The resulting training and test accuracy is shown in Table 5-6.

Table 5-6. Training and test accuracy of Shadow Area

Shadow	Training Accuracy			Testing Accuracy		
	Cup A	Cup B	Cup C	Cup A	Cup B	Cup C
Left_Q2	37.60%	39.40%	36.10%	55.60%	16.00%	18.20%
Left_Q3	26.70%	38.90%	29.40%	23.80%	26.30%	46.70%
Right_Q2	37.60%	38.30%	32.00%	55.60%	23.80%	69.20%
Right_Q3	26.70%	71.30%	28.40%	23.80%	84.20%	35.70%

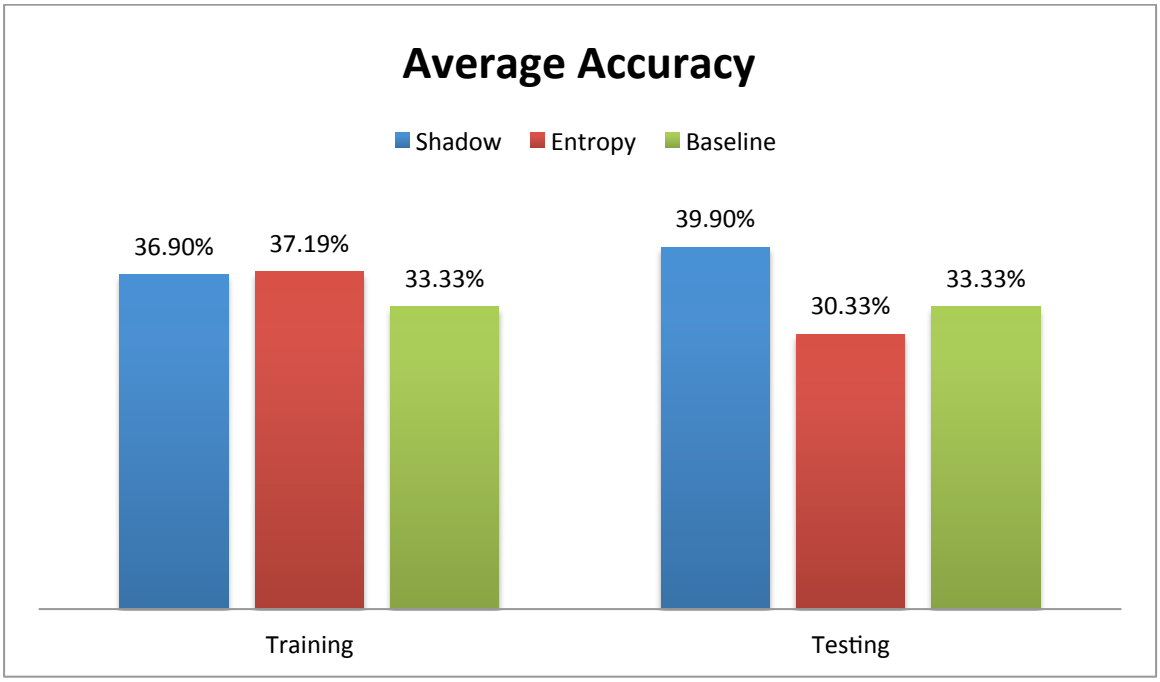


Figure 5-7. Average accuracy of shadow area, entropy and baseline

Now, this feature extraction is now benchmarked with the image entropy and baseline accuracy for comparison as seen in Figure 5-7. By observing their test accuracy, it can initially be concluded that the shadow area provides a better among the three test results. However, by observing the training accuracy, it can be seen that the shadow area has similar training accuracy and entropy. Given the observation from training accuracy, the higher test accuracy of the shadow area compared to image entropy might not be very significant. The only conclusion that can be drawn from Figure 5-7 is that the shadow area provides an accuracy that is greater than or equal to image entropy features when using linear regression as the prediction model.

5.4 Laws' Features Global Histogram Optimization

The results of finding the optimal parameters for Laws' Features are discussed in this section. The first step is to find the optimal number of bins for the 9 Laws' Histogram. It can be found by sweeping the number of bins from 1 to 100 and obtain the number of bins that provides the best results. Due to

limited memory of the utilized laptop, only two models are used for sweeping, (1) Multiple Linear Regression and (2) Support Vector Machine. The resulting graph is shown in Figure 5-8. The Series name RegTr and Reg represents Training and Testing Accuracy of Regression Model, respectively. While the series name SVMTr and SVM represents the training and testing accuracy of SVM model.

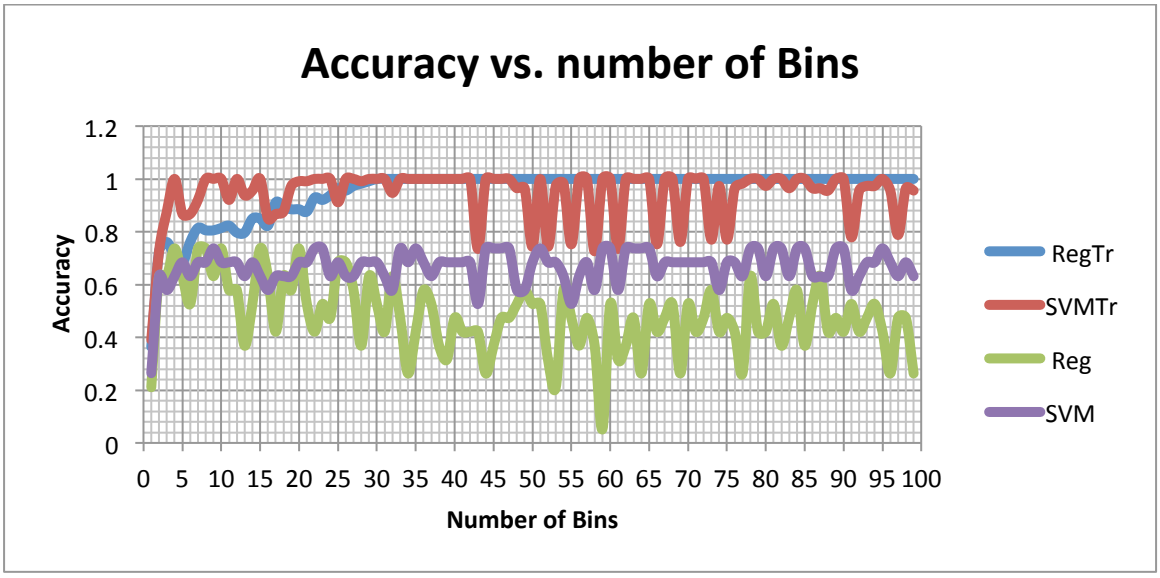


Figure 5-8. Parameter sweep Graph for Number of Bins of Laws' Histogram

It can be seen from the figure that the testing accuracy of SVM (violet) is generally higher than the regression model (green). The first conclusion drawn from the graph is SVM model is generally more accurate than the Regression model. Hence, in succeeding parameter learning, SVM model will be utilized as statistical model.

According to Ng[75], whenever the training error is higher than the test error, then the model is considered underfit. But if the training error is below the test error, then it is considered overfit. Hence, the best way to find the optimized model is to find the minimum difference of training and test error.

Since the model is considered nonlinear and contains many local minima, the technique utilized to find the top 15 numbers of bins is to find 15 results provide the minimum difference of training and test error then find the number of bins that provides the highest accuracy. The resulting list is shown in Table 5-7. There are six numbers of bins that provide the same test accuracy. But considering the relative value of their train and test accuracy difference, the best number of bins is 50.

The Laws' Features Histogram seen above does not contain any preprocessing scheme. This type of laws' histogram will be prone to varying illumination of the object. There are two pre-processing scheme to be studied. The first is the original preprocessing scheme by implemented by Laws himself [70]. The second one is the preprocessing done by Tan & Triggs[95]. This preprocessing is said to make images more robust under varying lighting conditions. It has been proven to be good preprocessor for face recognition.

Table 5-7. Top 15 numbers of Bins

Number of Bins	Train and Test Accuracy Difference	Testing Accuracy
1	12.62%	26.32%
2	9.41%	63.16%
5	18.30%	68.42%
43	20.82%	52.63%
50	5.92%	68.42%
52	5.92%	68.42%
55	22.59%	52.63%
58	14.67%	57.89%
61	15.56%	57.89%
66	12.06%	63.16%
69	7.69%	68.42%
73	8.57%	68.42%
75	8.57%	68.42%
91	19.98%	57.89%
97	15.60%	63.16%

Figure 5-9 provides the summary of parameter sweep for each pre-processor. Note that the original (i.e. no processors) is included for comparison. It can be seen that in general the Tan-Triggs preprocessor does not help much in improving the accuracy of estimation. Using the original Laws' pre-processor style, it has improved the accuracy to 73.68% when the number of bins has values of 39,43,46 and 53. It is notable that this pre-processor does not just improve its accuracy but also make the laws' texture feature more robust to illumination variation. Therefore, when using laws' feature, the best way is to use laws' preprocessor scheme and a number of bins of 39.

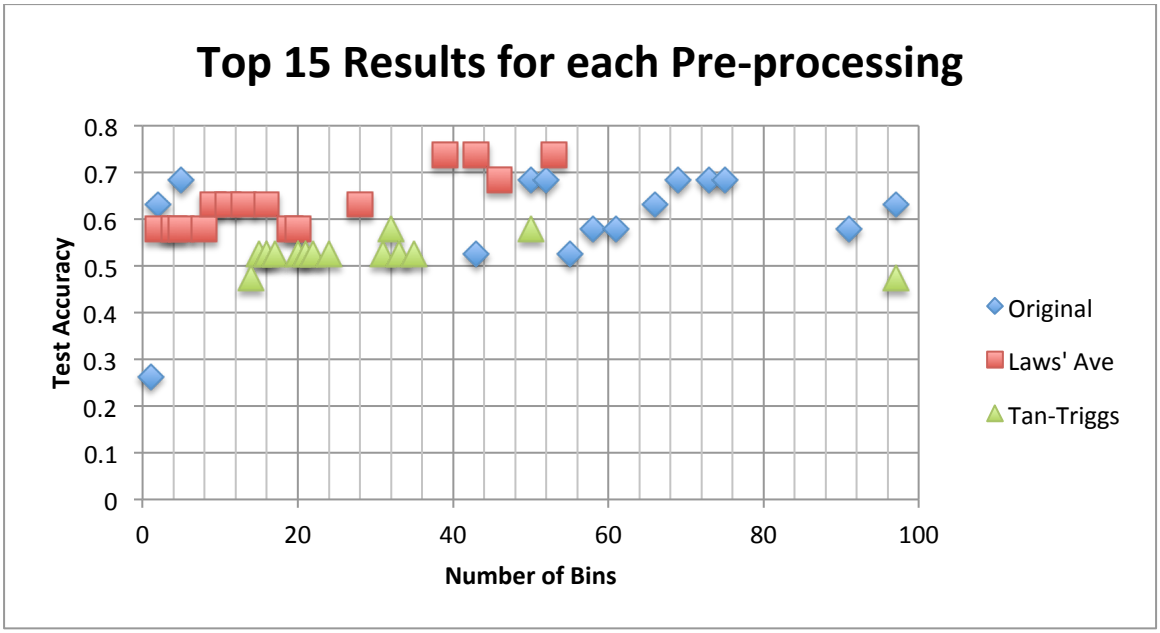


Figure 5-9. Best 15 Number of bins for each pre-processor

The window size used by Laws' Averaging pre-processor is found by sweeping window size from 1 to 50 and finds the best value similar to the procedure done in the number of bins for laws' filters. Table 5-8 summarizes the top results from the window size sweep. It can clearly be seen that window size should either be 3,6,18,39,46. However, 39 and 46 can be ruled out as using this window size (i.e. 39 x 39 and 46 x 46) is almost as big as the image. From the choices of 3,6 and 18, the window size of 3 is chosen.

Table 5-8. Top Values of Window Size

Window Size (n x n)	Train and Test Accuracy Difference	Testing Accuracy
3	5.96%	73.68%
6	6.43%	73.68%
9	0.61%	68.42%
18	5.08%	73.68%
39	5.08%	73.68%
44	5.92%	68.42%
45	5.92%	68.42%
46	0.65%	73.68%
49	4.15%	68.42%

5.5 Local Binary Pattern Global Histogram (LBPGH) Optimization

This section will provide the results for finding the optimal parameters for Local Binary Pattern Histogram. Similar to the previous section, the number of bins is swept from 1 to 100 then search for the optimum results. Based on the findings from the previous section, the SVM prediction model is utilized. The accuracy estimation will also be compared if an additional preprocessor is added, specifically Tan-Triggs pre-processor.

The results of the parameter sweep are shown in Figure 5-10. It can first be concluded that Tan-Triggs preprocessor does not improve the overall accuracy for images of Breast Self-examination. As Local Binary Pattern Features are inherently invariant to monotonic gray-level transformations [94], [95], the LBPGH with no preprocessor with 41 bins is utilized. Monotonic gray-level invariance means illumination invariance under certain conditions.

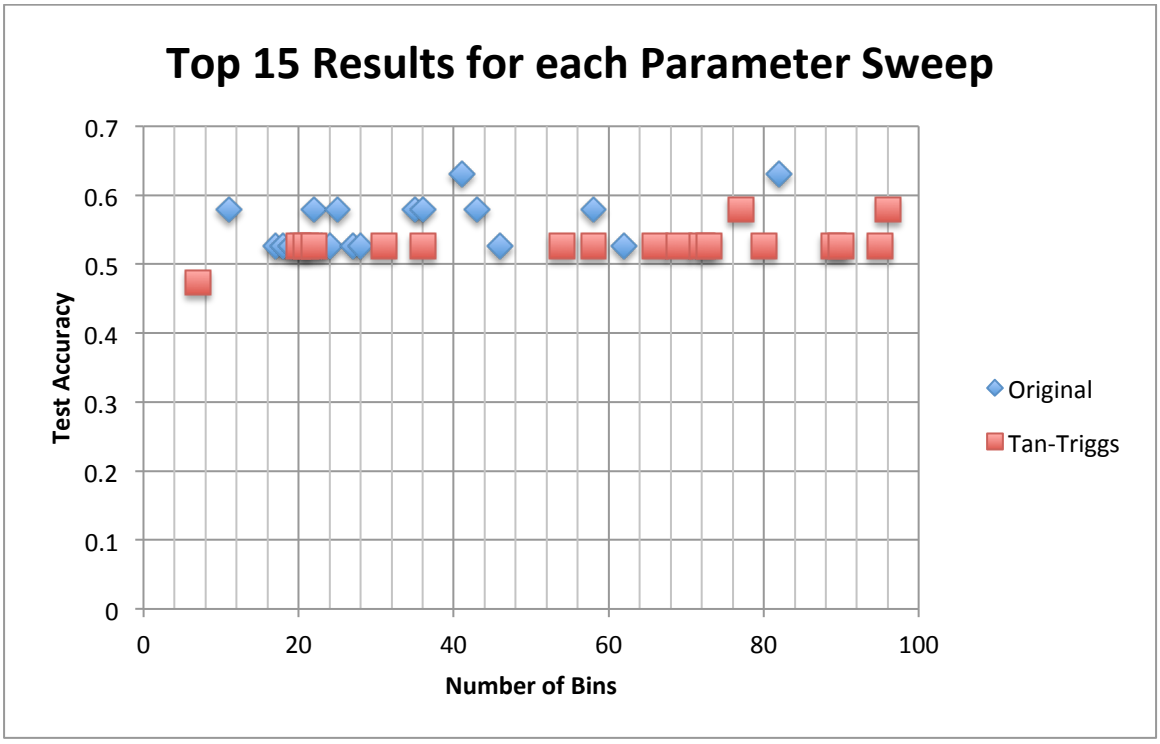


Figure 5-10. Parameter Sweep for LBPGH

5.6 Model Selection

Only the two simplest features are used which is the image entropy from section 4.6.2 and normalized area of the total shadow from section 4.6.1. The rationale is to simplify the feature space and complexity of the models and focuses only on selecting the best models that predicts depth.

There are four models to be selected upon: (1) Multiple Linear regression, (2) Artificial Neural Network, (3) Support Vector Machine, and (4) Gradient Boosted Trees.

For training Multiple Linear Regression (REG), the input features utilized are outlined in section 4.6.1 and 4.6.2. Then, normalized using L1 normalization. The training and testing accuracy of this model for all available quadrants are shown in Table 5-9 as column 1 and 5, respectively.

In training the Artificial Neural Network (ANN) model, the outline in Section 4.7.4 is followed. After extracting the features, image entropy (4.6.2) and normalized shadow area (4.6.1), normalize it using L1 norm then feed it to the model. The training and testing accuracy is shown column 2 and 6 of Table 5-9, respectively.

For Support Vector Machines (SVM), the training scheme is outlined in section 4.7.2. Similarly, the input features are only normalized shadow area (4.6.1) and image entropy (4.6.2). The training and testing accuracy is shown in Table 5-9 as column 3 and 7, respectively.

The last statistical model from the choices is the Gradient Boosted Trees (GBT). The implementation utilized in this study is seen in section 4.7.3. Only two feature vectors are used in GBT. The features matrices are appended by ones vector and normalized by L1-Norm. The training and testing accuracy is shown in Table 5-9 as column 4 and 8, respectively.

Table 5-9. Model Selection Accuracy Assessment

Model Selection	Training Accuracy				Testing Accuracy			
	REG	ANN	SVM	GBT	REG	ANN	SVM	GBT
Cup A								
Left_Q2	68.30%	28.70%	68.30%	100.00%	66.70%	33.30%	38.90%	44.40%
Left_Q3	60.00%	24.20%	70.80%	97.50%	19.00%	47.60%	76.20%	23.80%
Right_Q2	68.30%	28.70%	68.30%	100.00%	66.70%	33.30%	38.90%	44.40%
Right_Q3	60.00%	24.20%	70.80%	97.50%	19.00%	47.60%	76.20%	23.80%
Cup B								
Left_Q2	31.70%	24.60%	56.30%	95.10%	12.00%	16.00%	56.00%	60.00%
Left_Q3	31.90%	24.80%	60.20%	97.30%	26.30%	21.10%	52.60%	31.60%
Right_Q2	56.70%	41.70%	55.80%	98.30%	66.70%	42.90%	57.10%	47.60%
Right_Q3	65.70%	17.60%	71.30%	98.10%	68.40%	5.30%	84.20%	84.20%
Cup C								
Left_Q2	54.10%	26.20%	59.00%	100.00%	54.50%	27.30%	54.50%	27.30%
Left_Q3	47.10%	34.10%	57.60%	100.00%	46.70%	46.70%	60.00%	33.30%
Right_Q2	50.70%	22.70%	61.30%	100.00%	69.20%	7.70%	61.50%	53.80%
Right_Q3	51.90%	48.10%	56.80%	100.00%	42.90%	35.70%	42.90%	21.40%

Table 5-9 shows the list of all training and test accuracies in all quadrants. Taking its weighted test average summarizes these results and shown in Figure 5-11. The weighted average test accuracy is shown in red bar. The figure is already sorted the model from highest test accuracy to the lowest test accuracy. The baseline accuracy is also shown for comparison. The weighted training accuracy (shown in blue bar) is also put side-by-side with their corresponding test accuracy for reference.

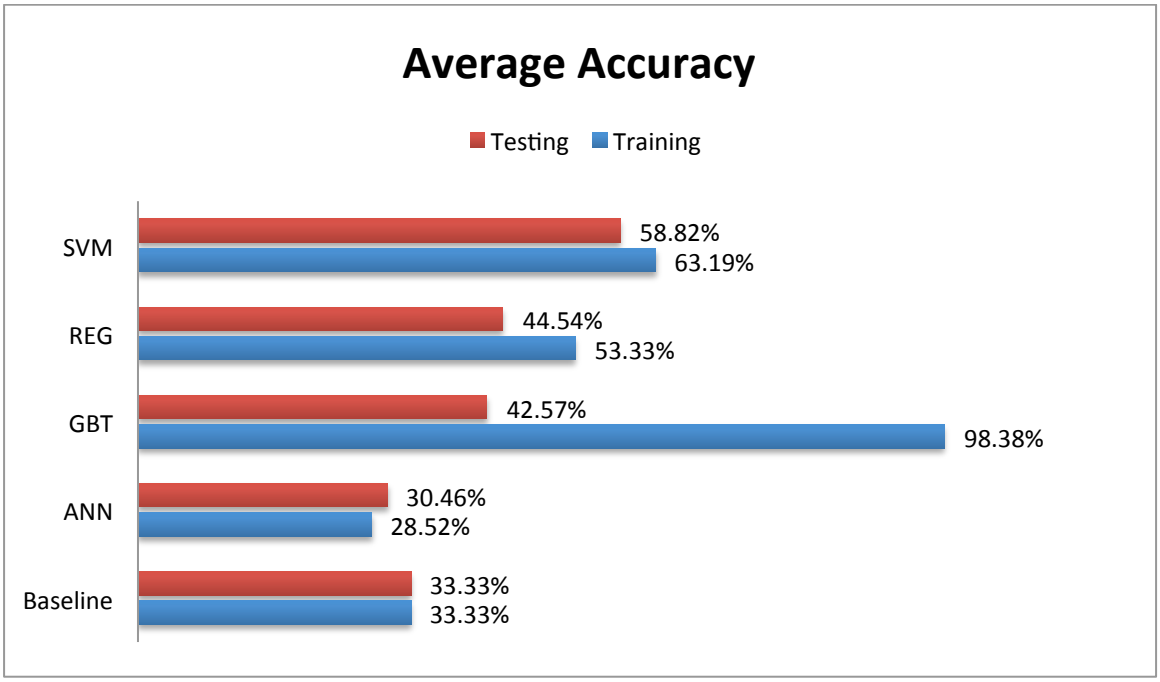


Figure 5-11. Model Selection Average Accuracy Assessment

5.7 Selection of optimal Feature Combinations

In this study, there are four features to choose from: (1) Image Entropy (Section 4.6.2), (2) Normalized Shadow Area (Section 4.6.1), (3) Laws' Textures Histogram (Section 4.6.3), and Local Binary Pattern Global Histogram (Section 4.6.4). However, it is not yet known which combination of these features provide the optimal accuracy. In this section, the results of selecting the best combination of features are shown.

Section 5.4 showed and discussed the optimal parameters for Laws' Features Histogram. While section 5.5 presented and discussed the optimal parameters needed for Local Binary Pattern Global Histogram. The findings of these sections have been utilized in this section. Meaning, Laws' Features Histogram and Local Binary Pattern Global Histogram of this section utilized the optimal feature extraction schemes already.

It should be noted that the prediction model utilized is based on the results of Section 5.6. The results from Section 5.6, which is thoroughly discussed in Section 6.4.1, show that the best prediction model to utilize for this study is Support Vector Machine (SVM). That is why, all accuracy assessed in this section are all based on the SVM prediction model.

This section shall first discuss the accuracy obtained when each of the features are utilized alone. Afterwards, it will show the results of the investigation of the different combinations of the features.

5.7.1 Accuracy Assessment of Each Features

When there are four features and only one has to be selected, then there are $4C1 = 4$ combinations of features. In this subsection, each feature sets will be the input to a separate SVM model. Hence, for each features, the goal is to determine accuracy it will provide to the model. The idea of this subsection will be proven useful for the analysis of feature combination, which is discussed in Section 6.4.2.

Concretely, for each quadrant of each cup size, from the four feature extraction schemes: Normalized Shadow Area (Section 4.6.1), Image Entropy (Section 4.6.2), Laws' Textures Histogram (Section 4.6.3), and Local Binary Pattern Global Histogram (Section 4.6.4), choose the first feature, and follow the outline of the SVM training scheme as discussed in Section 4.7.2. Then assess the training and testing accuracy using the confusion matrix (Section

4.8). After finishing the processing of the first feature, do the same procedure for the second, third and fourth feature. Afterwards, repeat all the procedures for all the available quadrants for each cup size. Note that each trained SVM model utilized the optimal parameters for the said prediction model.

Table 5-10 shows the results of training the SVM model for each of the available quadrants and for each feature. Each row shows the results for the given cup size and quadrant. For instance, row 2 is the 2nd quadrant of the left breast of Cup A. The first four columns shows the training accuracy for the four features while last four columns shows its corresponding testing accuracy. In this symbol Ent means entropy, Sha means normalized Shadow Area, Law means Laws' Features Histogram, and LBP means Local Binary Pattern Global Histogram.

Table 5-10. Feature Selection Accuracy

Features Accuracy	Training Accuracy				Testing Accuracy			
	Ent	Sha	Law	LBP	Ent	Sha	Law	LBP
Cup A								
Left_Q2	70.30%	61.39%	64.36%	70.30%	66.67%	61.11%	77.78%	61.11%
Left_Q3	75.83%	59.17%	67.50%	84.17%	66.67%	42.86%	52.38%	61.90%
Right_Q2	90.48%	43.81%	56.19%	73.33%	38.89%	61.11%	66.67%	66.67%
Right_Q3	67.00%	49.00%	52.00%	82.00%	52.94%	47.06%	58.82%	76.47%
Cup B								
Left_Q2	70.42%	44.37%	52.11%	79.58%	40.00%	64.00%	56.00%	68.00%
Left_Q3	79.65%	46.90%	49.56%	61.06%	57.89%	47.37%	73.68%	63.16%
Right_Q2	82.50%	54.17%	54.17%	85.83%	66.67%	52.38%	71.43%	61.90%
Right_Q3	80.56%	71.30%	71.30%	85.19%	84.21%	84.21%	78.95%	89.47%
Cup C								
Left_Q2	80.33%	44.26%	50.82%	62.30%	54.55%	45.45%	54.55%	72.73%
Left_Q3	74.12%	58.82%	47.06%	87.06%	33.33%	60.00%	60.00%	80.00%
Right_Q2	66.67%	53.33%	54.67%	80.00%	46.15%	46.15%	84.62%	69.23%
Right_Q3	81.48%	51.85%	55.56%	75.31%	50.00%	35.71%	78.57%	85.71%

Figure 5-12 summarizes the training and testing accuracy of Table 5-10 by calculating its weighted average for each feature. The blue columns are the

accuracies for Entropy Features (Ent), the red columns are the accuracies for normalize Shadow Area Feature (Sha), the green columns are for Laws' Features Histogram (Law), and the violet column is for Local Binary Pattern Global Histogram (LBP). The four columns on the left of the figure are the weighted average training accuracies while the four columns on the right are the weighted average testing accuracies. Note that the numerical value of the accuracies is displayed on top of each respective column.

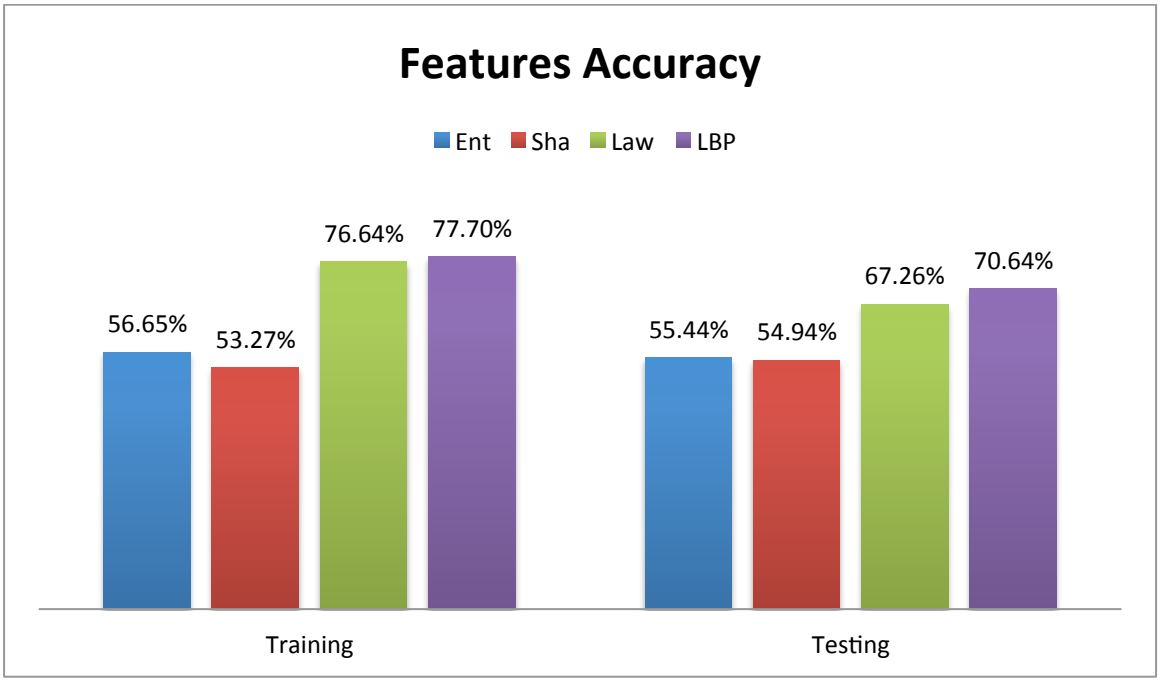
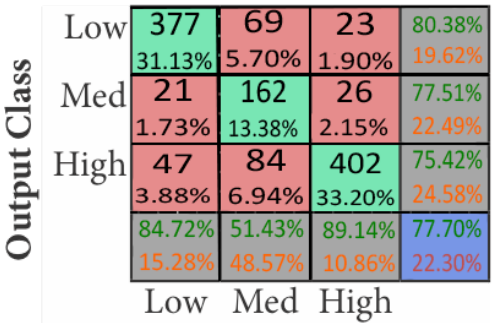
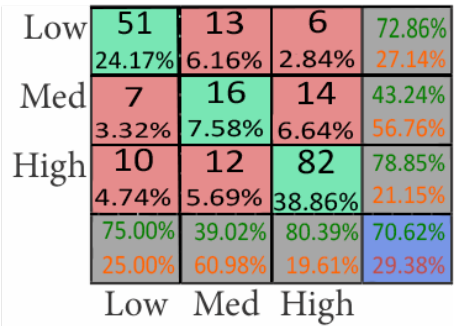
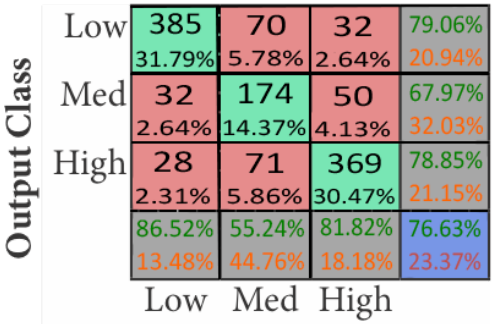
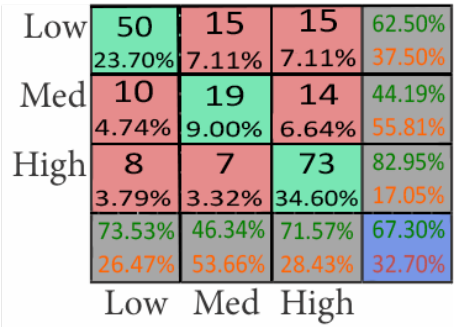
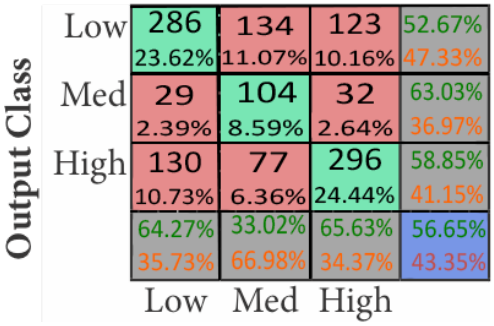
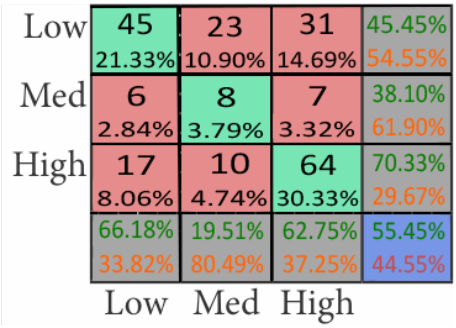
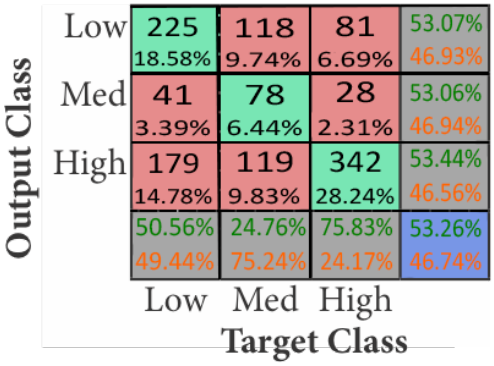
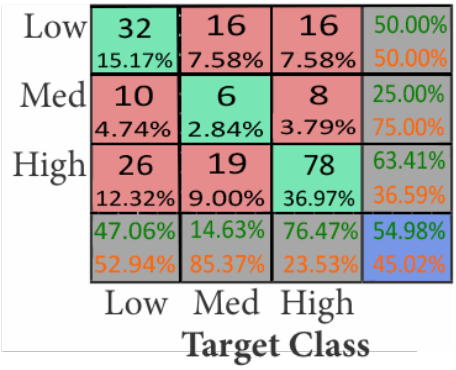


Figure 5-12. Feature Selection Average Accuracy

In order to study behavior of each feature extraction schemes, their corresponding confusion matrices are shown in Table 5-11. It should be noted that all confusion matrices presents the overall accuracy in all quadrants and cup size. This type of table for data presentation is uncommon so it should be explained with great details. This table is composed of four rows and two columns. The first column presents the confusion matrices for the training set while the column presents the confusion matrices for the test set. The four rows are labeled as: A, B, C, and D. These labels are written below the specified row.

Table 5-11. Confusion Matrix (A) LBP, (B) Law, (C) Entropy, (D) Shadow

Training	Testing
	
A)	
	
B)	
	
C)	
	
D)	

Each row represents different features. Row A represents LBP feature (Section 4.6.4), row B represents Laws Features (Section 4.6.3), row C represents Shadow Features (Section 4.6.1) and row D represents Entropy Features (4.6.2). Hence, each row presents the confusion matrices for the specified feature. For convenience, the arrangement of rows is sorted from the feature that provides highest test accuracy to the lowest test accuracy.

5.7.2 Accuracy Assessment for all combinations of Features

In the previous subsection, it shows the assessed accuracy for each of the four features in the SVM prediction model. The final step is to check all the combinations of the four features and check which is of them provides the optimal results. So given that are four features to choose from and we want to gather all combinations, then it means there are $4C2 + 4C3 + 4C4 = 12$ combinations. Intuitively, the best combination is when all input features are utilized to the prediction model. However, sometimes, some features might not be a good partner with others. It is the motivation of this section to know which combinations provides the optimal results and which features contribute the greatest.

Figure 5-13 shows the overall training and testing accuracy for all 12 combinations. It is already arranged from the combinations that provide the highest test accuracy to the lowest test accuracy. This figure plots the training accuracy and testing accuracy as blue bar and red bar, respectively. The x-axis shows the values of the accuracy. The y-axis represents the combination names. Ent refers to Image Entropy (Section 4.6.2), Sha refers to Normalized Shadow Area (Section 4.6.1), Law refers to Laws' Textures Histogram (Section 4.6.3), and LBP refers to Local Binary Pattern Global Histogram (Section 4.6.4). It is also notable that the numerical value of the test accuracy is also displayed in front of each red bars in the figure. The SVM model utilized for training each

combination utilized the optimal parameters that provide the highest test accuracy.

In order to study the behavior of the highest combination, Table 5-12 lists the confusion matrices of the highest test accuracy combinations. Note that as seen in Figure 5-13, EntLawLBP, lawLBP, and ShaLawLBP provides equal test accuracies. That is why Table 5-12 listed the confusion matrices of these three combinations.

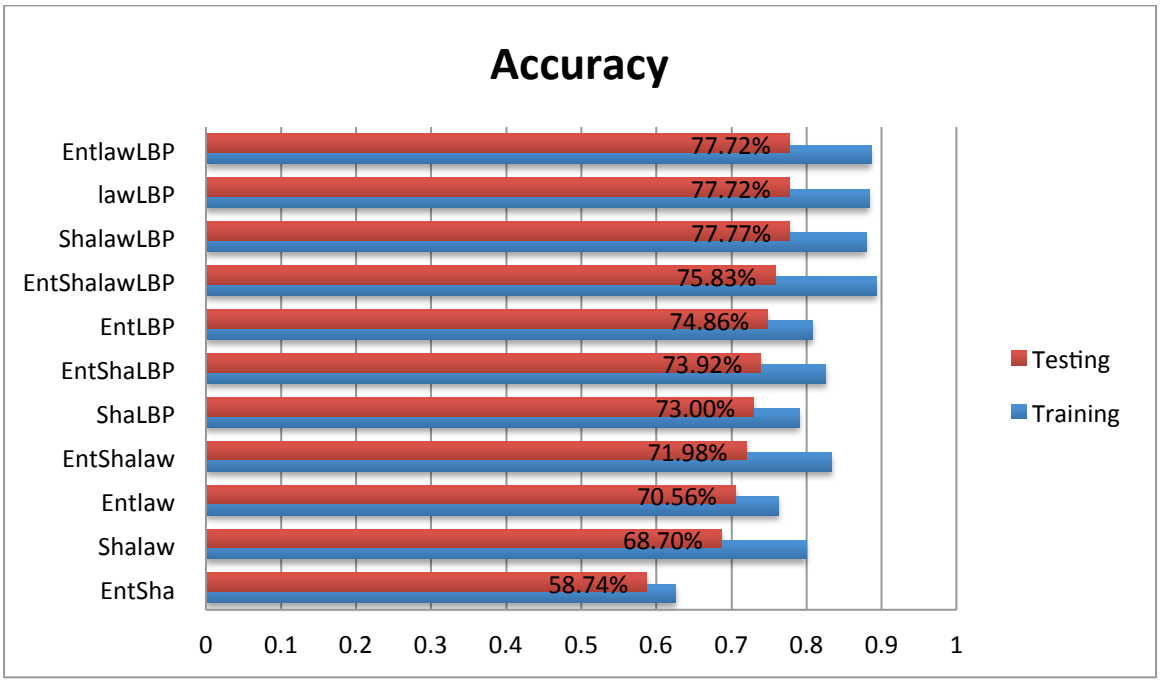
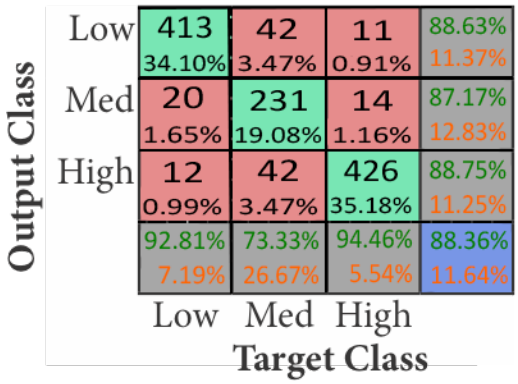
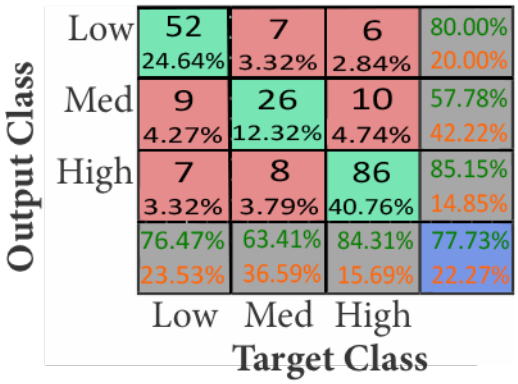


Figure 5-13. Feature Combination Average Accuracy

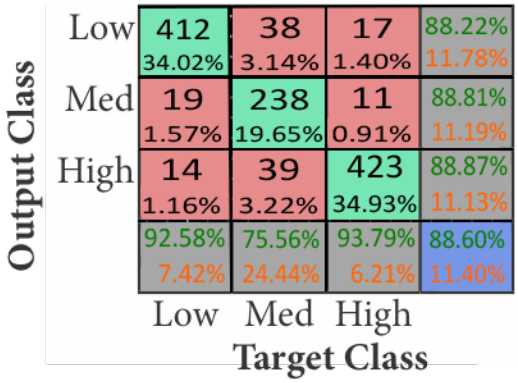
Table 5-12. Confusion Matrix: (a) LawLBP, (b) EntLawLBP, (c) ShaLawLBP

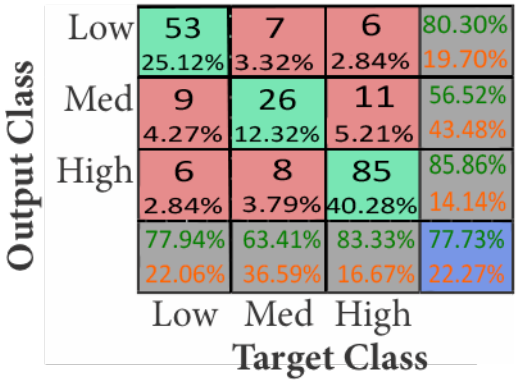
Training	Testing



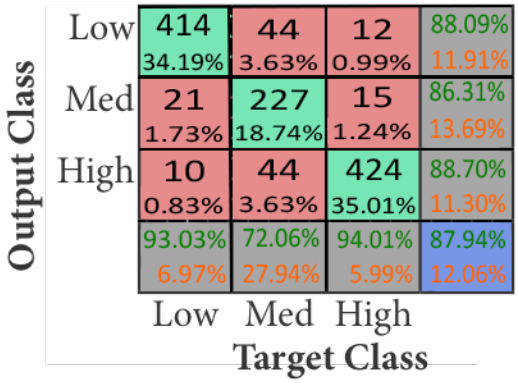


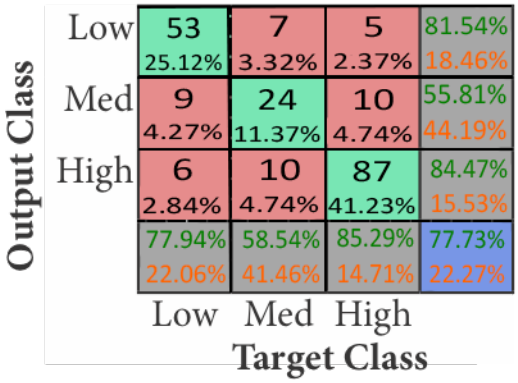
a)





b)





c)

CHAPTER 6
Discussion And Analysis

6.1 RGBD BSE dataset

RGBD BSE dataset provided an avenue to quantify depth palpation as shown in section 5.1. Using the Depth Map provided by Kinect and the method to scalarize and quantize as discussed in Section 4.4, multiple variations of depth level estimation algorithm was constructed as shown in Section 5.6 - 5.7.

However, based on the results of Section 5.1, there is a limitation with dataset. First, it is the use of near mode operation of Kinect. This mode of operation reduces the accuracy of Kinect. A comprehensive study of kinect is done by Khoshelham and Elberink[67]. But this study only focused from [67] a minimum distance 1m to 3m distance. In an official Microsoft webpage, it has also been stated that the recommended distance is 1m to 3 [96]. But it also told that Kinect can operate from 0.5 to 1m. This mode of operation is called near mode. It doesn't provide the best quality compared when captured from 1m to 3m. But it is reliable enough for some Kinect based application.

Apparently, the research chose a trade-off between lower reliability and lower resolution of RGB and breach of privacy. It is shown in Table 5-2, and shown here as Table 6-1, the list of final image resolution of each image ROI for each of the available quadrants. Clearly, the list shows a very low resolution for all quadrants. The highest resolution is at 84 x 55 on cup B right brest quadrant 3. One can imagine that increasing the distance of Kinect from the subject to at least 1m will reduce the resolution to further lower resolution from ¼ to 1/2. Given in Table 6-1 that the resolution is already too low to capture finer details, reducing from ¼ to ½ might not be good decision. Now, even if it the lower

resolution is good enough there would be a problem with privacy as the goal is not to see the faces of the subject in the recorded video.

Table 6-1. Box Mask Resolution

Box Resolution	Cup A	Cup B	Cup C
Left_Q2	38 x 33	49 x 59	68 x 58
Left_Q3	46 x 30	77 x 55	74 x 43
Right_Q2	44 x 47	67 x 61	56 x 37
Right_Q3	46 x 38	84 x 55	60 x 39

6.2 Quantitative Evaluation

Part of the objective of this thesis is to perform a quantitative comparison with the state-of-the-art depth estimation algorithm for BSE. Apparently, the state-of-the-art for this type of research is from Chen et al[19] where they use image entropy to estimate depth estimation. However, they only provided a graph qualitatively showing change in entropy value whenever the finger palpates the breast vertically. They did not utilize any quantitative evaluation scheme to check its accuracy. Without a proper evaluation scheme, it is hard to benchmark their algorithm with the algorithm of other researchers.

The evaluation scheme developed in this study provides a framework to calculate the quantitative accuracy of the developed depth estimation algorithm and the quantitative accuracy of the depth estimation algorithm for the previous studies and for future studies. The quantization scheme is very simple, as it requires only two input variables to characterize the fuzzy model.

Table 6-2. Fuzzy Parameters

Fuzzy Model Parameters	Cup A		Cup B		Cup C	
	MIN	MAX	MIN	MAX	MIN	MAX
Left_Q2	762 mm	794 mm	607 mm	678 mm	568 mm	609 mm
Left_Q3	744 mm	771 mm	603 mm	617 mm	563 mm	607 mm
Right_Q2	772 mm	790 mm	619 mm	684 mm	591 mm	668 mm
Right_Q3	771 mm	801 mm	614 mm	658 mm	597 mm	673 mm

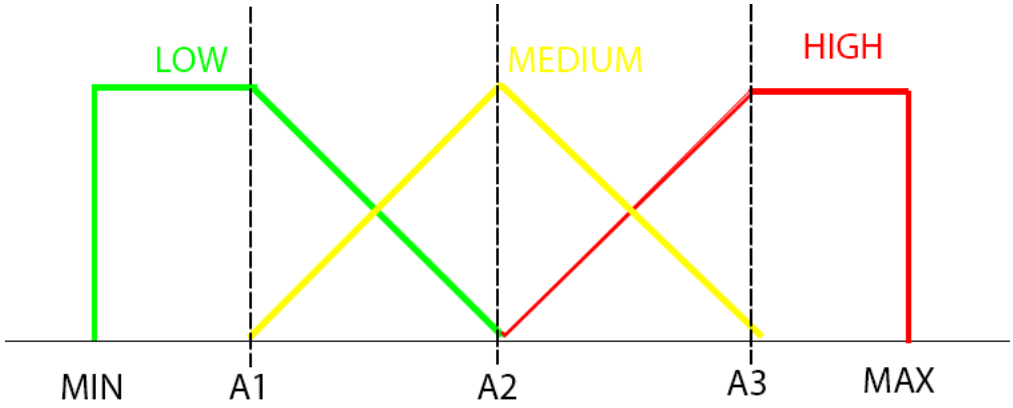


Figure 6-1. Model used for depth quantization

However, one can see a limitation on this quantization scheme. First, by simply observing the variable MIN and MAX from Table 5-3, it can clearly be seen that it varies between subjects. Different distances between the camera and the subject cause it during the actual recording. Knowing that only the torso should be the only one recorded, it can be shown that the distance between the camera and the subject will become dependent on the height of the subject, elevation angle of the camera, and the zoom of the camera. Hence, to fixed this issue, one alternative is to make the zoom and elevation angle of the camera constant, then get the relationship of variable MIN and MAX to the height of the subject. In an actual BSE, the fixed zoom and elevation angle may be instructed to the user prior to the start of BSE. Hence, this simple quantization scheme can still be utilized for BSE in all types of women.

6.3 Benchmarking with state-of-the-art

In order to benchmark the algorithm with the state-of-the-art, there is a need first to know the accuracy of the state-of-the-art. The accuracy on each quadrant is shown in Table 5-5. Its average is seen in Figure 5-6 and shown here as Figure 6-2. As it can be seen in the figure, the test accuracy is 30.33%. This accuracy is almost similar to the baseline accuracy. This means that in actual Breast Self Examination procedure, its depth prediction accuracy is as

good as selecting it at random. Hence, to create a better depth estimation algorithm, the latter should estimate depth better than random classification.

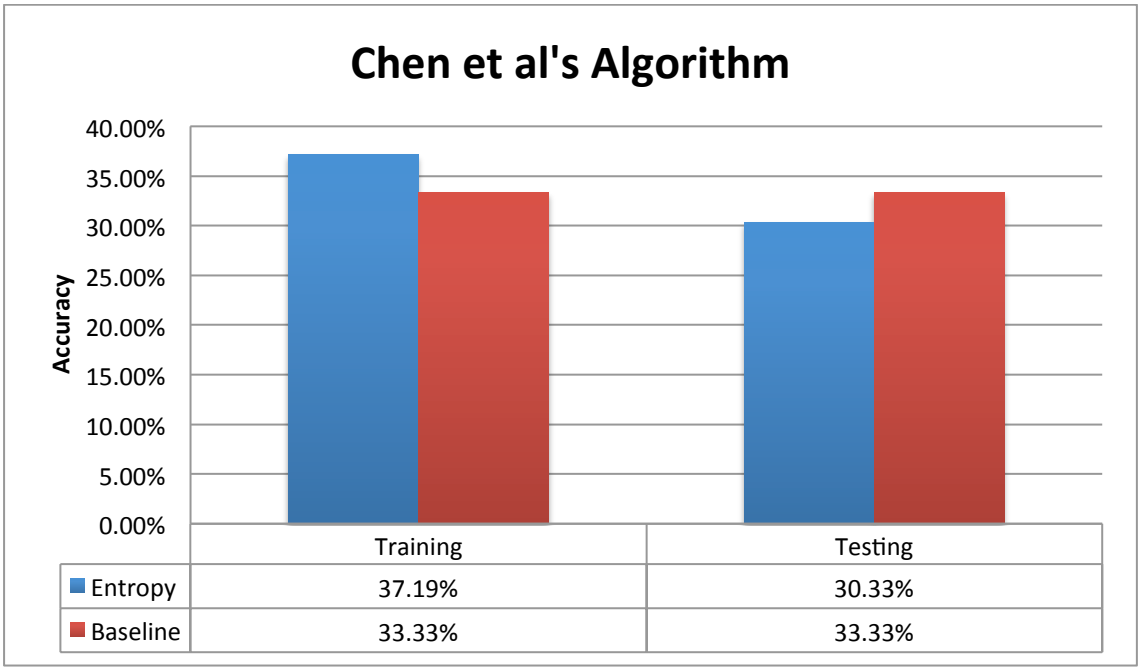


Figure 6-2. Chen et al's algorithm

6.4 Depth Classifier

6.4.1 Prediction model selection and analysis

The results of Section 5.6 can be summarized by the average accuracies of all models that is shown as Figure 5-11 and shown here as Figure 6-3. Again, the figure shows the training and testing accuracy seen as blue bar and red bar, respectively. The numerical value of each bars are also shown beside their corresponding bars. The sequence of models is arranged from the highest test accuracy to the lowest accuracy. In the figure, Support Vector Machine is symbolized as SVM, Multiple Linear Regression as REG, Gradient Boosted Trees as GBT, and Artificial Neural Network as ANN.

Figure 6-3 provides a clear indication that Support Vector Machine should be chosen as the prediction model. Having the highest test accuracy of

58.3%. Its training accuracy of 63.1% provides a good indication of optimality since this value is near to the test accuracy of 58.3%. Although the training and test accuracy of SVM is concluded to be superior compared to others. Other details and factors for each model is considered and discussed in the succeeding paragraphs.

Based on the figure, only SVM, Multiple Linear Regression, and GBT provide test accuracy higher than the baseline accuracy. Again, the baseline accuracy is the accuracy for random classification on a 3-class classifier. ANN seems to be optimal already as the training accuracy (30.4%) and testing accuracy (28.8%) has value near with each other. It only has 1.6% difference with each other. Hence, ANN should be ruled out from the choices of prediction model.

Gradient Boosting Trees algorithm provides the best training accuracy of 98.7%. When using training accuracy as a basis alone, GBT is the best model. However, Gradient Boosting Trees model has lowered its test accuracy by half compared to its training error. Knowing that the training accuracy is significantly higher than the test accuracy means that training scheme makes the model underfit as explained in section 5.4. Knowing the theoretical structure of GBT (Section 3.9.2) where it inherits the property of decision trees algorithm, the most probable reason shall be the lack of extensive training set. But the focus of this study is to develop a depth level estimation algorithm rather than making an extensive dataset for BSE. Hence, utilization of GBT as the prediction model shall be ruled out and the issue stated above can be a topic of the next research which will be discussed in section 7.2.

Now, for the multiple linear regression model, it provides a training and test accuracy of 53.9% and 46.5%, respectively. Observing the difference of training and test accuracy that is 7.4%, it means that this model is almost

optimal already. It is also notable that the combination of using both image entropy features and normalized shadow area features is superior compared on using one of the features alone. It can be vividly seen in Figure 6-4. It can be seen that the test accuracy of utilizing both features provided a higher test accuracy of 46.5% while using image entropy or normalized shadow area alone provided 30.33% and 39.9% accuracy, respectively.

Utilization of both image entropy and normalized shadow area features is shown to provide better results than using both features alone. This is shown using the linear regression model. However, using all consideration above and the results shown in Figure 6-3, the linear regression might not seem the best choice. From the discussion in the preceding paragraphs, there are two choices left: Regression and SVM. Knowing that both SVM and regression are already approximately optimal. Hence, only one factor should be considered, which is, the model that provides better test accuracy. Hence, the SVM model is the best choice among the four models.

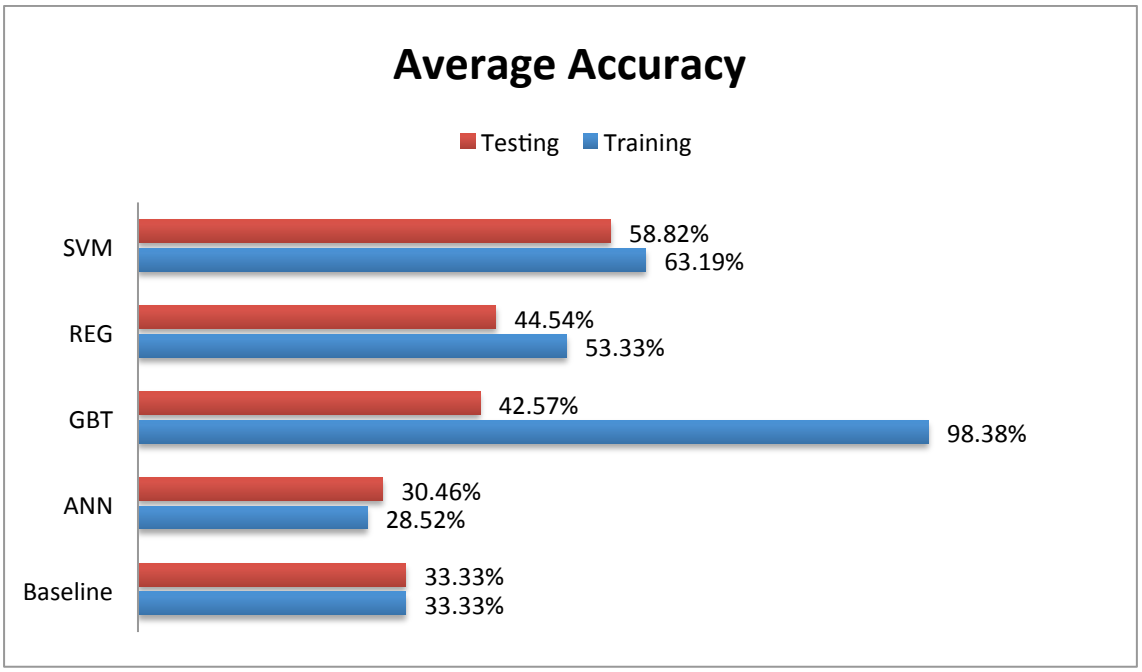


Figure 6-3. Prediction Model Selection Average Accuracy

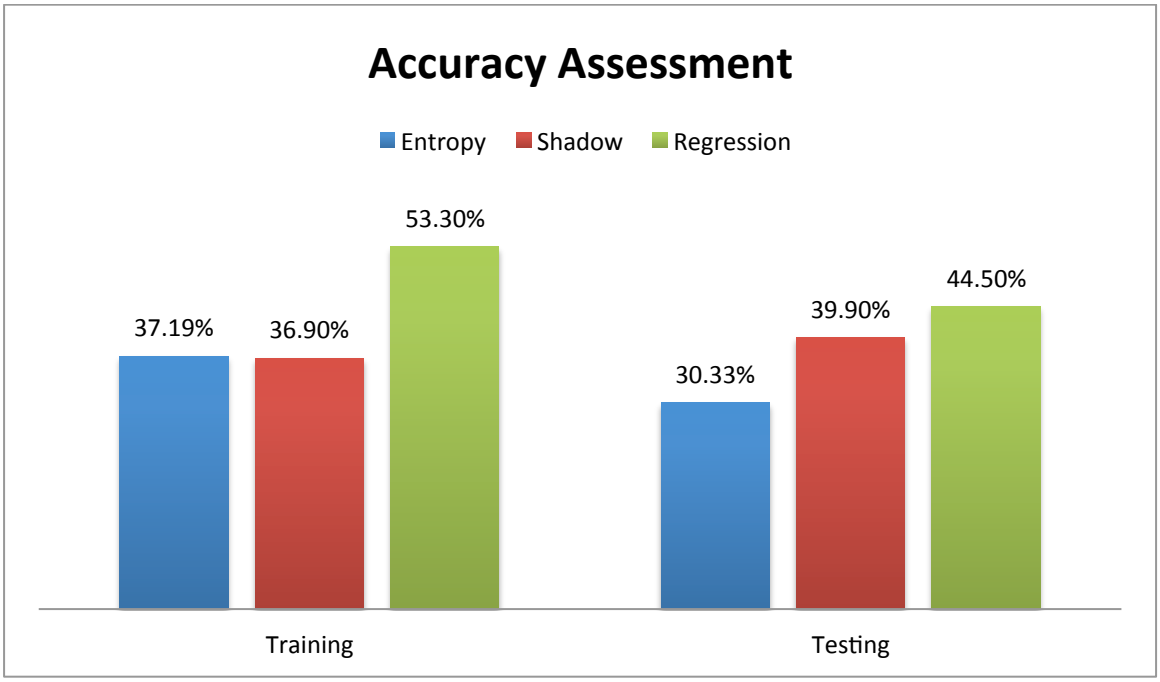


Figure 6-4. Comparison of Accuracy between different regression schemes

6.4.2 Features selection and analysis

The overall training and testing accuracies is shown in Figure 5-12 as shown here as Figure 6-5. Now, it should be noted again that the prediction model utilized for this section as discussed in the previous section (6.4.1) is the SVM. The laws' features utilized is the optimized laws' features as found by Section 5.4, and the LBP feature utilized is the optimized LBP features as found by Section 5.5. For brevity, the following symbols shall be used for this succeeding discussion:

- Law = Laws' Textures Histogram
- LBP = Local Binary Pattern Global Histogram
- Ent = Image Entropy
- Sha =Normalized Shadow Area

By observing Figure 6-5, there is a clear indication that the strongest features that predict depth are the Local Binary Pattern Global Histogram (LBP)

and Laws' Textures Histogram (Law). They provide a test accuracy of 70.64% and 67.26%, respectively. Again, as discussed in the preceding paragraph, each of these features and the SVM model is already optimized. Using the same idea from Section 5.4 and 5.5 for optimizing these features, the closeness of the values of training accuracy with their respective test accuracy shows that it is expected that these models be already optimized for providing the highest test accuracy.

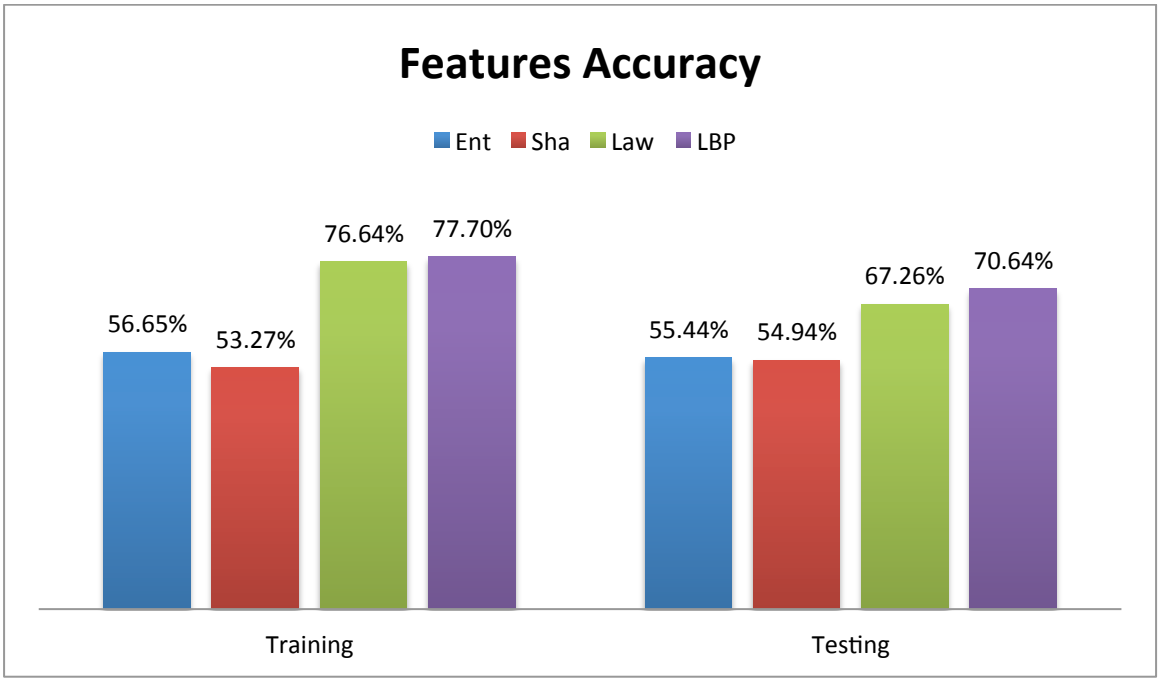


Figure 6-5. Training and Testing Accuracy of each Features

By knowing that Law and LBP provide the highest test accuracy features, it is expected in the succeeding discussion that these features should also give high performance. Now, Figure 6-6 shows the 12 combinations of features from the four available features.

The said figure shows that the best combinations are the EntLawLBP, lawLBP, and ShaLawLBP. Given that these 3 combinations have equal test accuracy and considered optimal already. Hence, the combinations to be

chosen should then be the feature combination that provides the least number of features. Since these three combinations contains the Laws' Features (Section 5.4) and LBP features (Section 5.5), then it can be concluded that appending either Ent or Sha does not provide significant contribution to the Law and LBP combination. The succeeding paragraphs shall discuss more details regarding the behavior of these combinations.

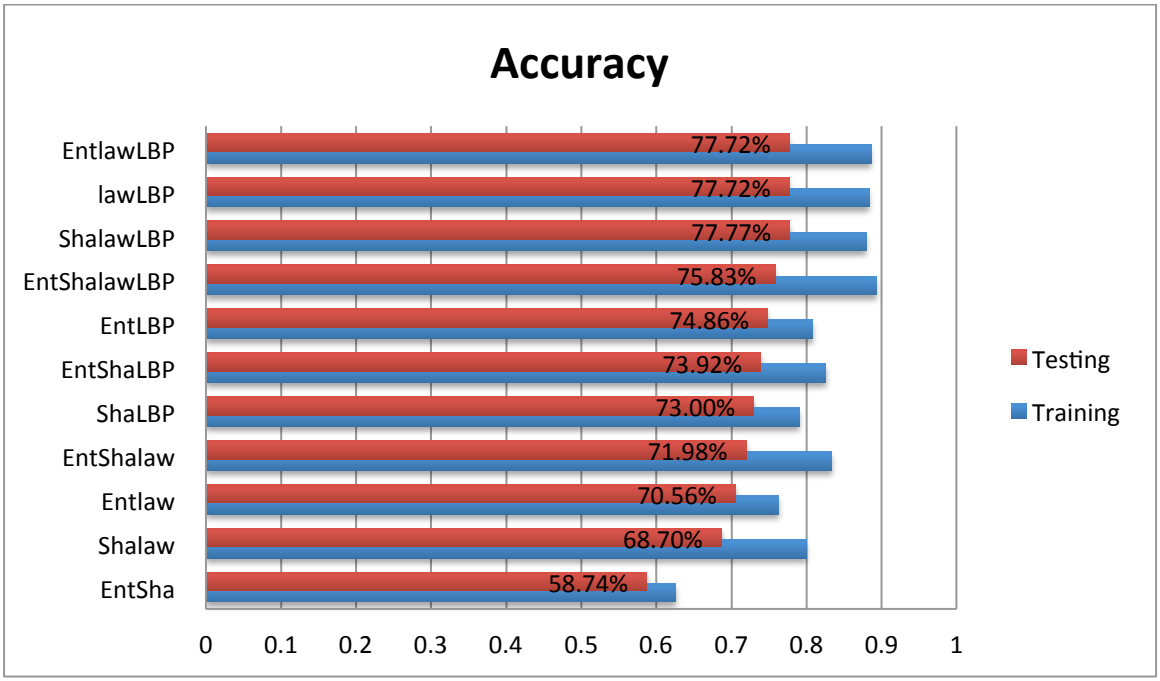


Figure 6-6. Feature Combination Average Accuracy

As discussed in the analysis of Figure 6-5, the Law and LBP are the best features that describe depth for this study. These are the "strong" features. Consequently, the Ent and Sha can be called the "weak" features. However, weak features cannot be disregarded immediately as other algorithms have relied on "weak" features like Haar-like features by Viola-Jones[77] and Mohammadi et al[84]. It is the aim of this subsection whether these features should be really disregarded or not. Figure 6-7 compares the accuracy of Laws' Features alone and Laws' Feature together with the combinations of the weak features. It can be seen that appending one weak feature (i.e. Ent or Sha) to

Law provides only a slight increase to the accuracy. However, if both of the weak features are appended, then it provides the highest increase to the accuracy. The Laws' Features provided 67.26% accuracy. Appending Sha added only 1.44% increase while Ent added 3.3%. Appending both of them provided 4.72% increase. Hence, adding Ent and Sha complements Laws' Features for estimating depth.

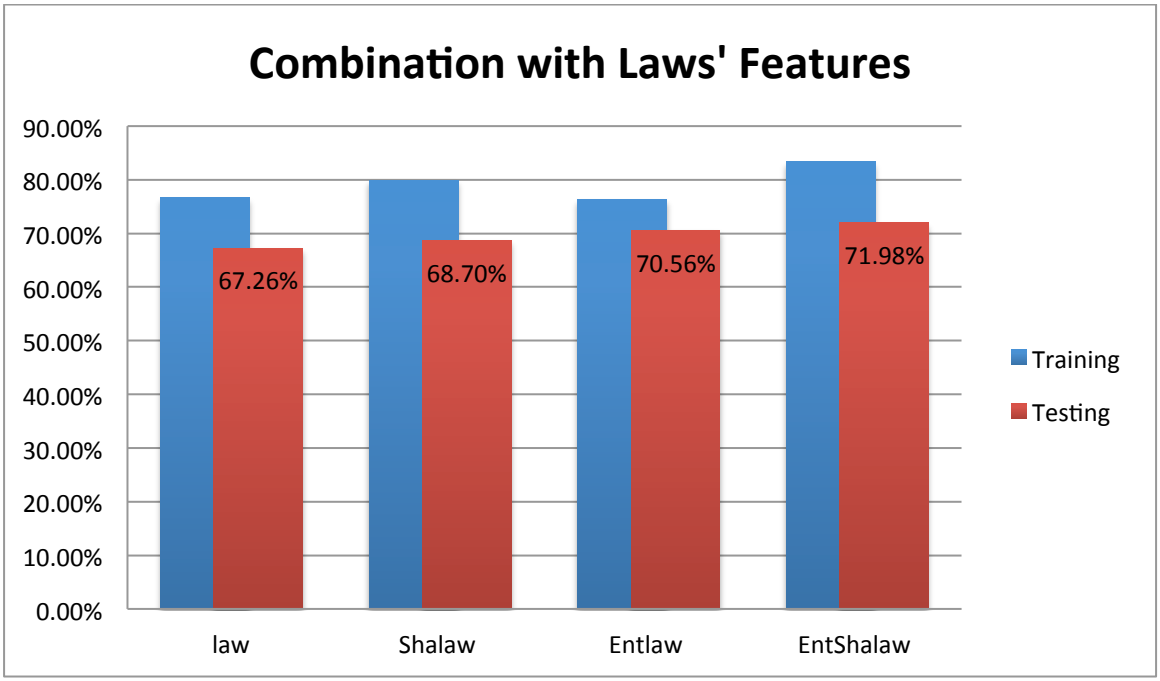


Figure 6-7. Comparison of different combinations with Laws' Features

Similar to Law, Figure 6-8 compares the accuracy between utilizing LBP alone and LBP appended with the combinations of the weak features. However, LBP behaves differently compared to Law. When only one weak feature is appended, then it provides a slight increase in accuracy. But, if both weak features are appended, it decreases the accuracy. When Sha was appended, it added 2.36% increase, while Ent added 4.22% increase. However, when both of them are appended, it added 3.28% increase. Given that 3.28% is only a slight deviation from 4.22% or 2.36%, it can be concluded that adding both of

the features to LBP does not provide an accuracy higher than using one of the weak features alone.

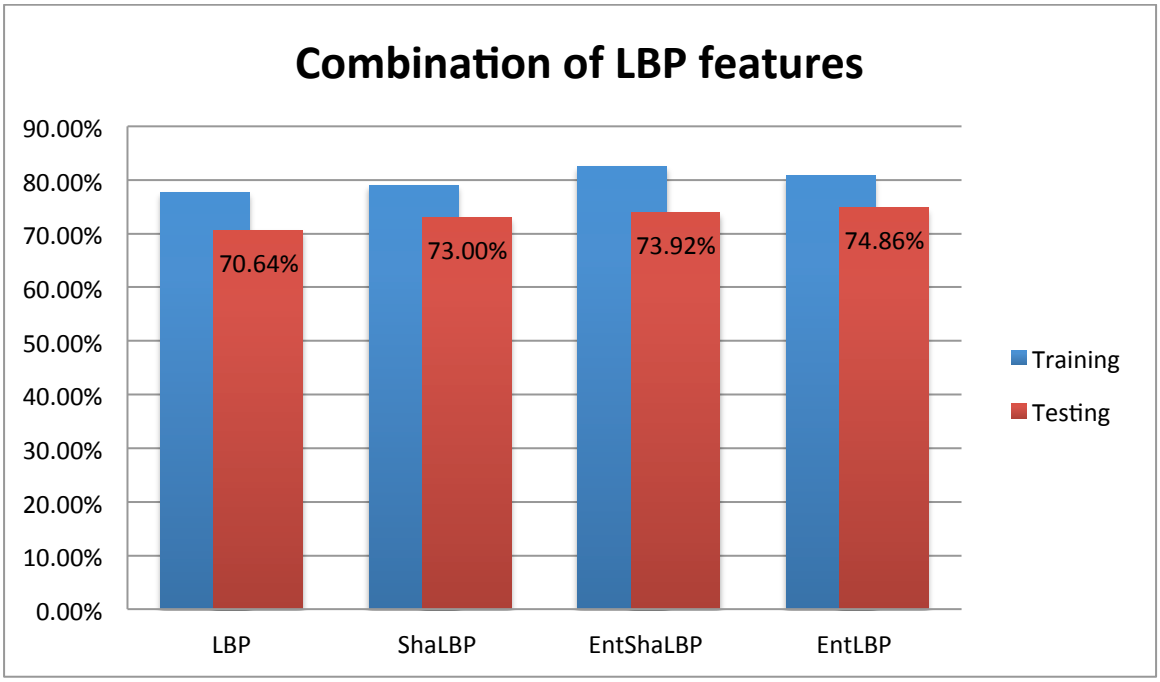


Figure 6-8. Comparison of different combinations with LBP features only

From the analysis of Figure 6-7 and Figure 6-8, it is known that utilizing both weak features is not good for LBP. Hence, the good option is to append only one weak feature especially when Law and LBP are combined. This conclusion is verified in Figure 6-9. It should first be noted that the results here are extracted from Figure 6-6. When both Ent and Sha are appended to LawLBP combination, it decreased the accuracy compared when utilizing only the combination of Law and LBP. But the interesting part is when either Ent or Sha is appended. It does not provide a significant increase in accuracy. This is in contrast when either Ent or Sha is appended to either Law alone or LBP alone. Moreover, their confusion matrices for the training and test as seen in Table 5-12 does not vary much. The false positive rate of each class and true negative of each class is similar to all the three combinations.

One good reason for this is the ratio of inserted features LawLBP to Ent/Sha. When Law is inserted, using the optimized results from Section 5.4, it represents, 9*39 = 351 input features of the SVM model. When LBP is inserted, using the results from Section 5.5, it represents 41 input features. Hence, when both Law and LBP are utilized, then it represents 351 + 41 = 392 input features of SVM. Hence, since Ent/Sha provides only one input features, it provides a 392:1 ratio of input features for LawLBP to Ent/Sha.

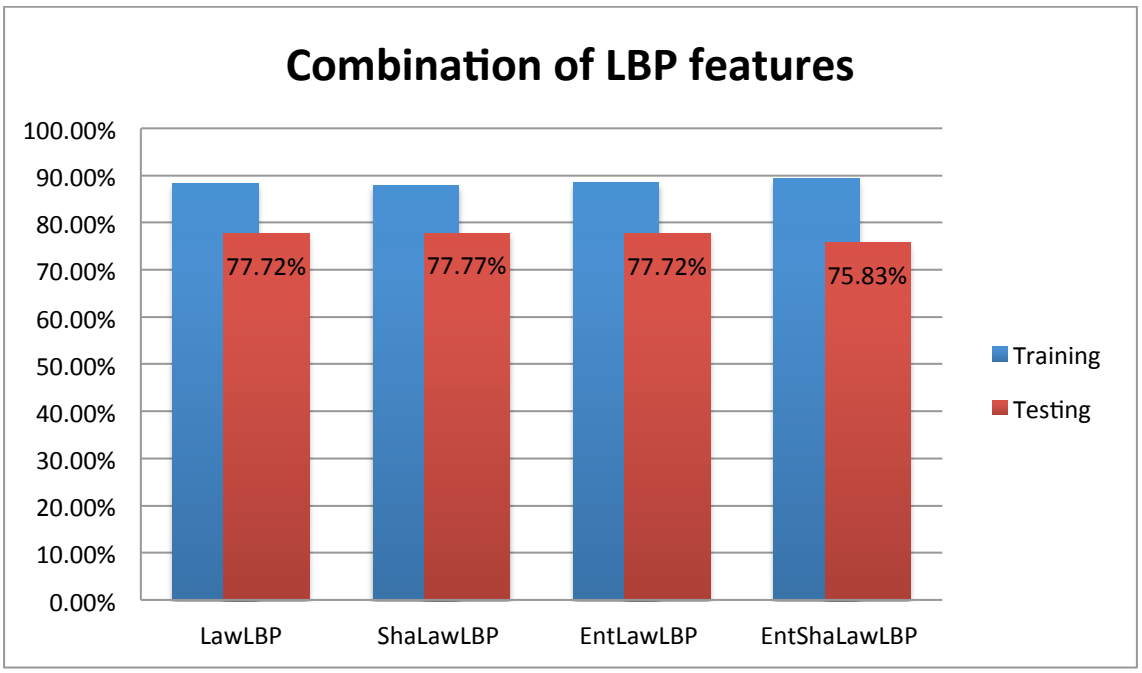


Figure 6-9. Comparison with different combinations for LawLBP

Therefore, using Figure 6-6 and Figure 6-9, the best combination is Laws' Textures Histogram plus Local Binary Pattern Global Histogram. Adding Image Entropy or Normalized Shadow Area does not significantly improve the accuracy. Hence, it is better not to use either of the weak features.

6.4.3 Overall analysis

Hence, using the all of the results from Chapter 5 and analysis of Chapter 6, the depth level estimation algorithm proposed by this study shall be a support vector machine that is implemented by OpenCV. The feature

extractions scheme to use are the Laws' Textures Histogram (Section 4.6.3) utilizing the parameters found in Section 5.4 plus the Local Binary Pattern Global Histogram (Section 4.6.4) utilizing the parameters found in Section 5.5. Among all combinations of four predictions models, four feature extraction schemes, this combination provides the highest accuracy. It provided test accuracy 77.71% in the recorded RGBD BSE Dataset evaluated by the proposed evaluation scheme.

This depth estimation algorithm is superior to the algorithm of the previous study of Chen et al[19]. It should be noted that their algorithm as benchmarked in the RGBD BSE dataset provides only 33.33% accuracy. This accuracy should be expected as the scope of their study includes only palpation of the breast that has a direction towards the breast. In actual BSE, the women typically palpates the breast with circular motion. Even if they are simply pressing towards the chest, the finger movement includes leftwards or sidewards movements.

However, this algorithm did not add the limitation of Chen et al's algorithm. Rather, this algorithm is expected to work with 77.71% accuracy on actual BSE performance as long as the cup size, the MIN and MAX of depth level quantization of each quadrant is given to this algorithm prior to BSE palpation proper.

CHAPTER 7
Conclusion and Recommendation

7.1 Summary

A comprehensive Breast Self Examination guidance system requires that the user properly palpate all quadrants of the breasts. Up until now, BSE is taught to women as pressing each quadrant with three levels of pressure: low, medium, high [3]. Without the proper pressure level, she may miss lumps that can only be felt at deep level palpation. But pressure is very subjective to the user. There was no quantifiable method to estimate it. Fortunately, we can estimate pressure using depth level palpated by the user. A common depth camera can easily capture depth changes by at least 1mm. However, a depth camera is not accessible to common people as depth cameras are relatively new. What is accessible to most people is an RGB camera.

This is the purpose of this study. This thesis would like to develop depth level estimation algorithm for monocular Breast Self Examination image sequence. Meaning, with the use of RGB cameras, the algorithm will predict depth level whether it is low, medium or deep. The previous study [19], [97], as benchmarked by our evaluation scheme, have shown only an average accuracy of 30.33%. This accuracy is just about the same as randomly picking the depth level. A random 3-level classifier can has an accuracy of 33.33%. Our depth level estimation algorithm was able to classify with an overall test accuracy of 77.71%. Our algorithm provides a 250% higher accuracy than the state-of-the-art.

The second contribution of this study is the RGBD BSE dataset. Quantifying subjective tasks like pressure level requires values. A ground truth is also the main requirement for calculating Errors and misclassifications.

Previous studies cannot estimate their error, as they do not have ground truth. With the use of RGB-D BSE recording, quantifying the error of the proposed depth level estimation algorithm is possible. Furthermore, it was now also possible estimating the error of previous studies, which is connected to the first contribution.

Our last significant contribution is that a method of quantifying low, medium and deep pressure level was proposed. With the use of “Fuzzy-Like” membership relation and annotated Finger Mask Sequence and Box Mask Sequence, extracting ground truth of depth level is possible. With this evaluation scheme, this study would not be able to provide quantified accuracies and benchmarking with the previous studies.

7.2 Future Work and Recommendations

One future work is to improve the accuracy of the classifier. The Texture feature extraction scheme utilized a simple summary statistics. It only captures the global features of the ROI. It ignores the spatial features of the image ROI. One might explore Ahonen’s Local Binary Pattern Histogram[98] where it partitions the image into multiple windows and captures the LBP histogram for each window. Another similar yet more robust suggestion is the idea of Dalal-Triggs HOG[99]. But perhaps, the most robust would be the Markov Random Field where it represents the image similar to Ahonen but adds a relationship function between windows. It was the framework utilized by Saxena et al[20] for estimating monocular depth for outdoor scenes. One can also explore the extension of Ojala et al [94] for Local Binary Pattern feature to make it more robust.

Another means to improve the accuracy is to add non-texture features as predictor of depth. As discussed in section 2.4.2, one might explore shape from shading, structure from motion, etc. However, it should be warned that based

on an undocumented experimentation, feature extraction algorithm like optical flow, corner features have not been successful as a good feature for depth level for this thesis.

Another good recommendation is the exploration of Gradient Boosted Trees Algorithm. Based on our experiments, it provides an average of 98.7% training error. But, it was only able to capture 41.3% of test set. This is caused by the nature of tree-based learning tools. It requires a large amount of dataset and large features to provide optimum results. Although in this work, the SVM-based algorithm dominated average test errors, it is predicted that if a sufficiently large dataset is available, it will provide most robust results among the learning algorithms proposed.

The RGBD BSE dataset can also be improved by adding more volunteers with different breast structure, skin tone, etc. It provides a more robust trained model so that it can predict depth even if the subject is not very similar to the training set. One might request the help of breast cancer organization for helping the said idea.

Improving the proposed evaluation scheme can also be a good avenue. The said evaluation scheme requires the input of variable MIN and MAX. But these variables, as discussed in Section 6.2, is dependent on the height of the subject and the elevation angle of the camera. Although Section 6.2 provided a simple workaround, it will not be a good solution for generalized usage of the BSE mobile app. Some might not read or forgot the instruction during the BSE performance. Perhaps, one can utilize a better evaluation scheme or create an algorithm that automatically extracts the variable MIN and MAX.

Among all the recommendation, the most important one is to generalize depth estimation algorithm for all quadrants. This study tackled only the quadrants that provide noticeable palpation. As discussed in Section 5.1.3,

there are other quadrants that occlude the finger palpation. These are the “outer” quadrant (i.e. Q4 and Q5 of left and right breast). As it cannot gather textures of fingers, “pressed” area, and “unpressed” area, one might explore other cues like arm movements, deformation of breasts, etc.

Appendix

1. Number of low, Med, and high ground truth frames per quadrant

Quadrant	Training			Testing		
	LOW	MED	HIGH	LOW	MED	HIGH
Cup A						
Left_Q2	38	29	34	9	6	3
Left_Q3	33	55	32	2	5	14
Right_Q2	41	18	46	3	4	11
Right_Q3	47	40	13	9	6	2
Cup B						
Left_Q2	51	35	56	5	4	16
Left_Q3	41	28	44	10	4	5
Right_Q2	50	24	46	9	3	9
Right_Q3	19	12	77	1	2	16
Cup C						
Left_Q2	22	12	27	5	1	5
Left_Q3	40	20	25	5	3	7
Right_Q2	24	23	28	5	1	7
Right_Q3	39	19	23	5	2	7
Total	445	315	451	68	41	102

2. Large size Images of each Histograms in section 4.7.2

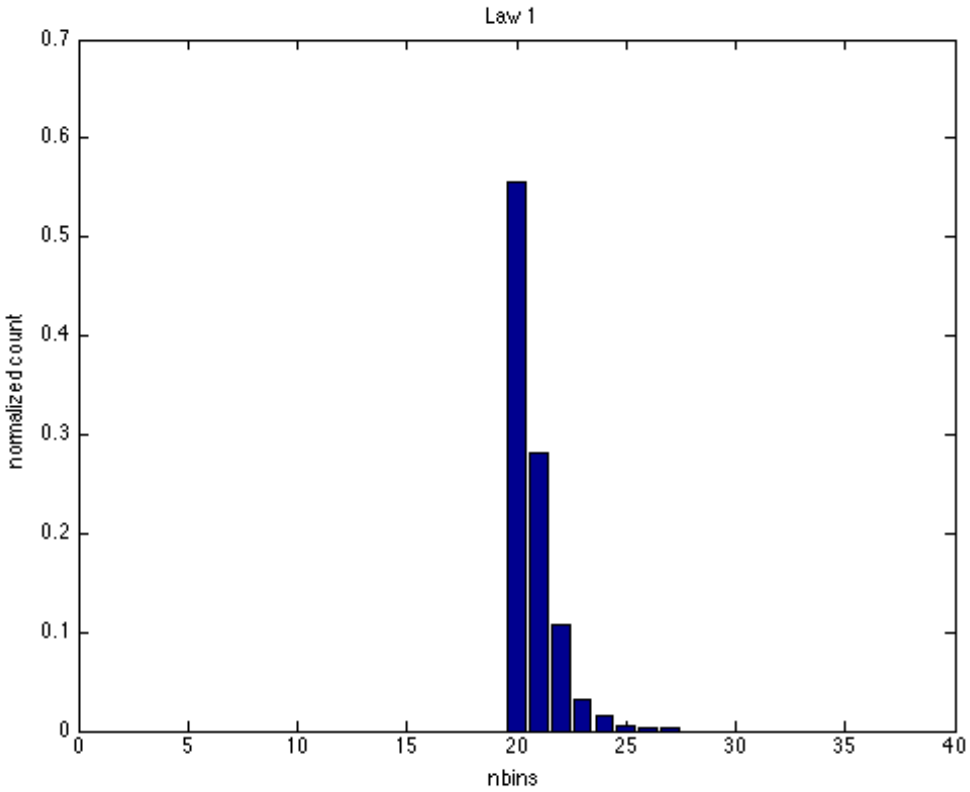


Figure a. Laws' 1st Texture Histogram

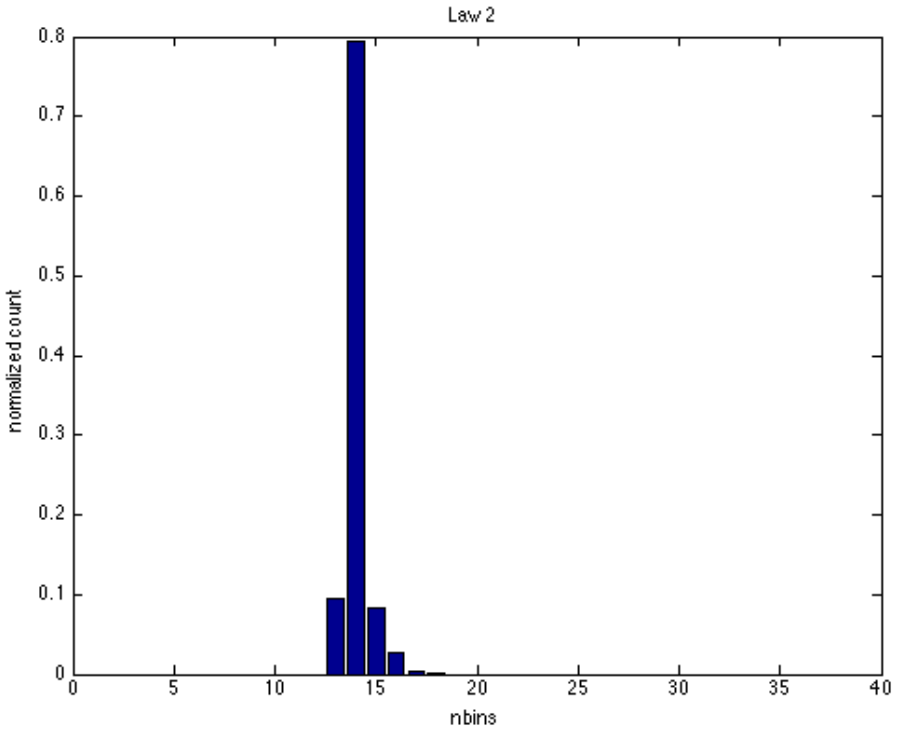


Figure b. Laws' 2nd Texture Histogram

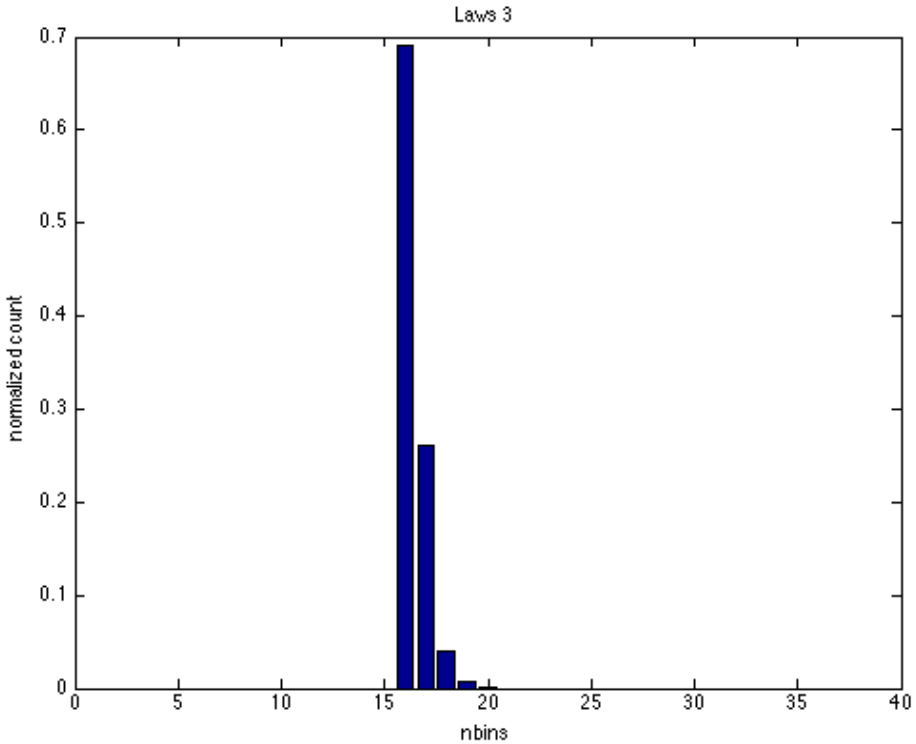


Figure c. Laws' 3rd Texture Histogram

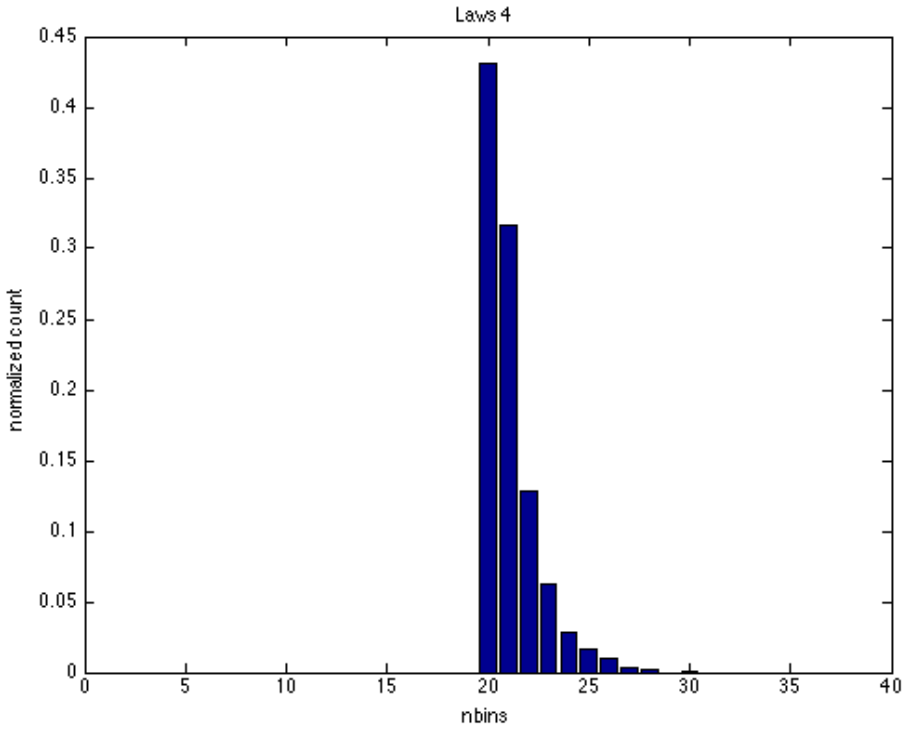


Figure d. Laws' 4th Texture Histogram

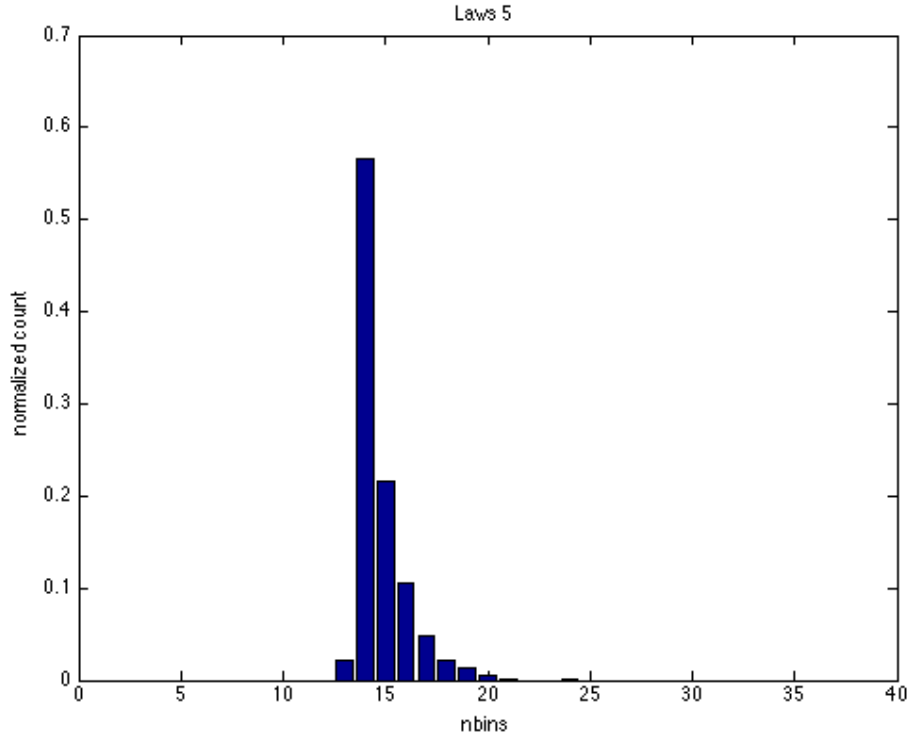


Figure e. Laws' 5th Texture Histogram

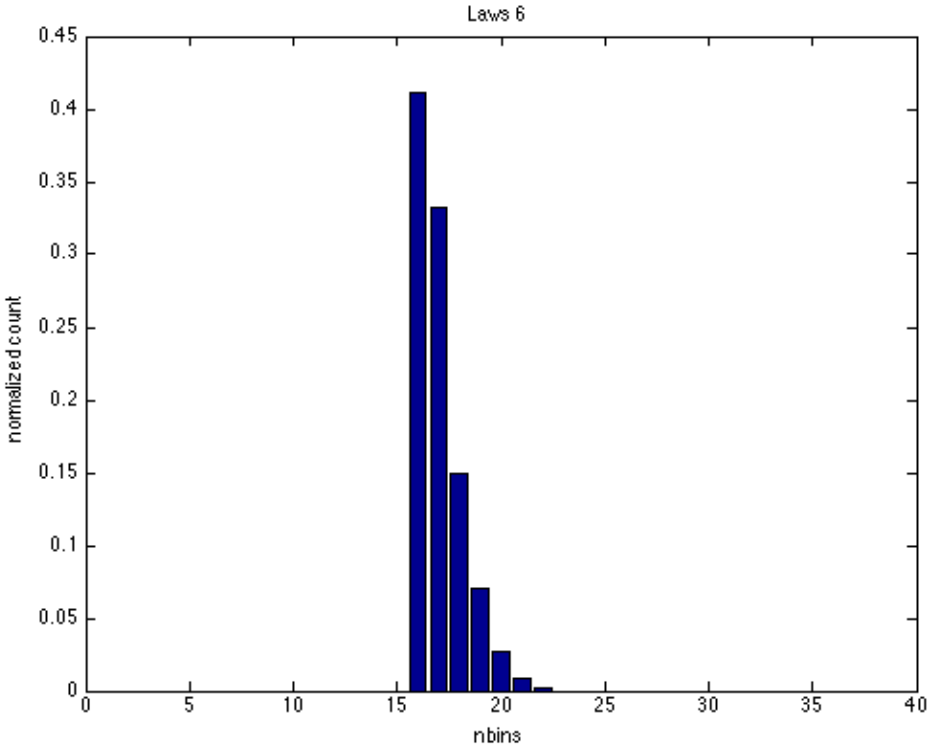


Figure f. Laws' 6th Texture Histogram

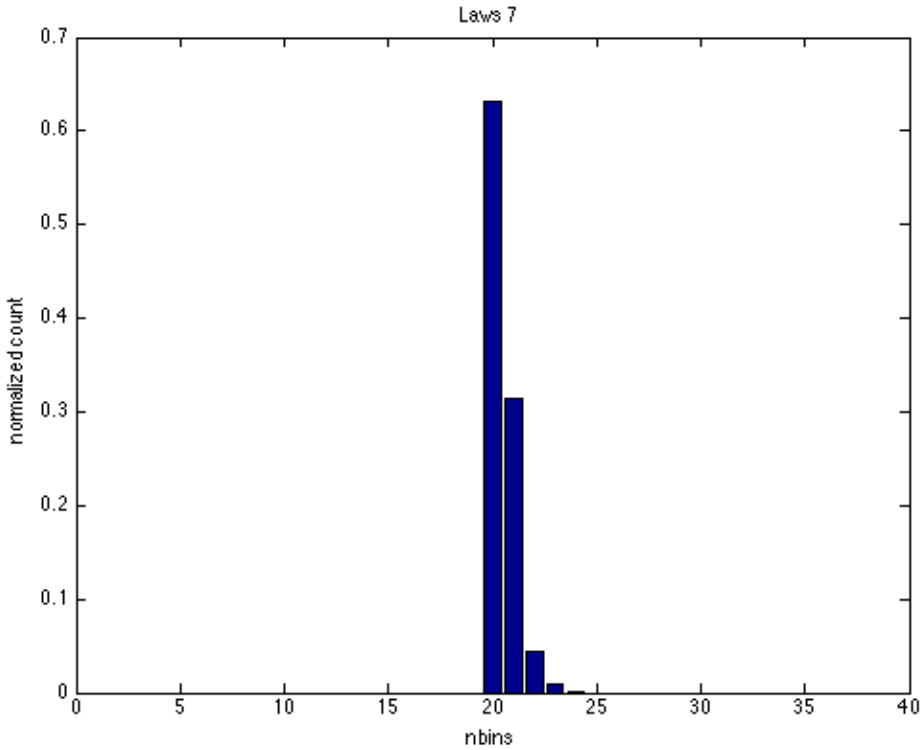


Figure g. Laws' 7th Texture Histogram

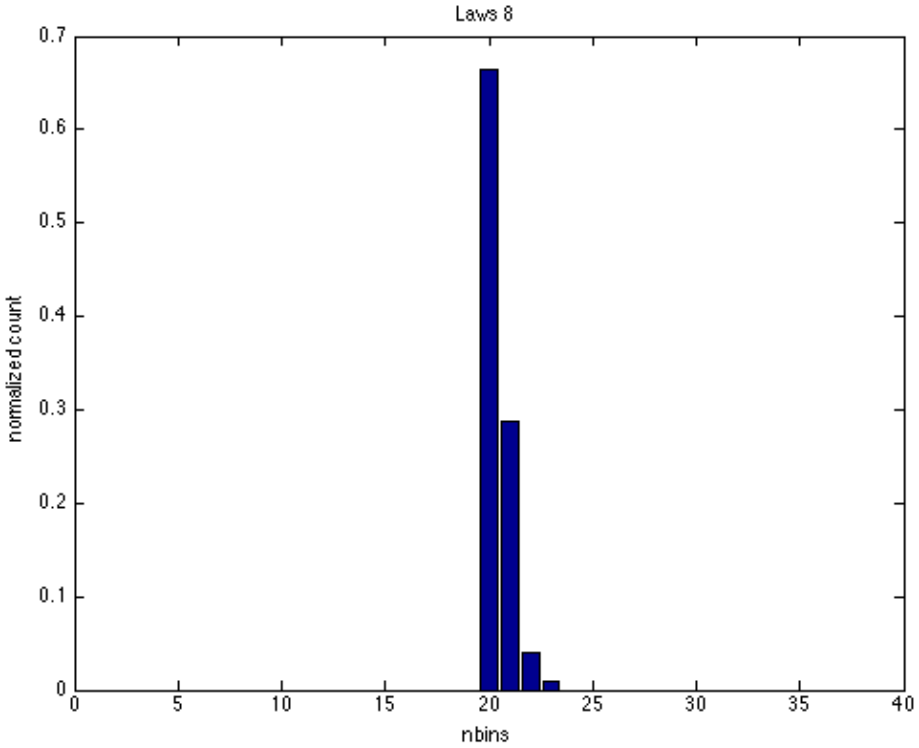


Figure h. Laws' 8th Texture Histogram

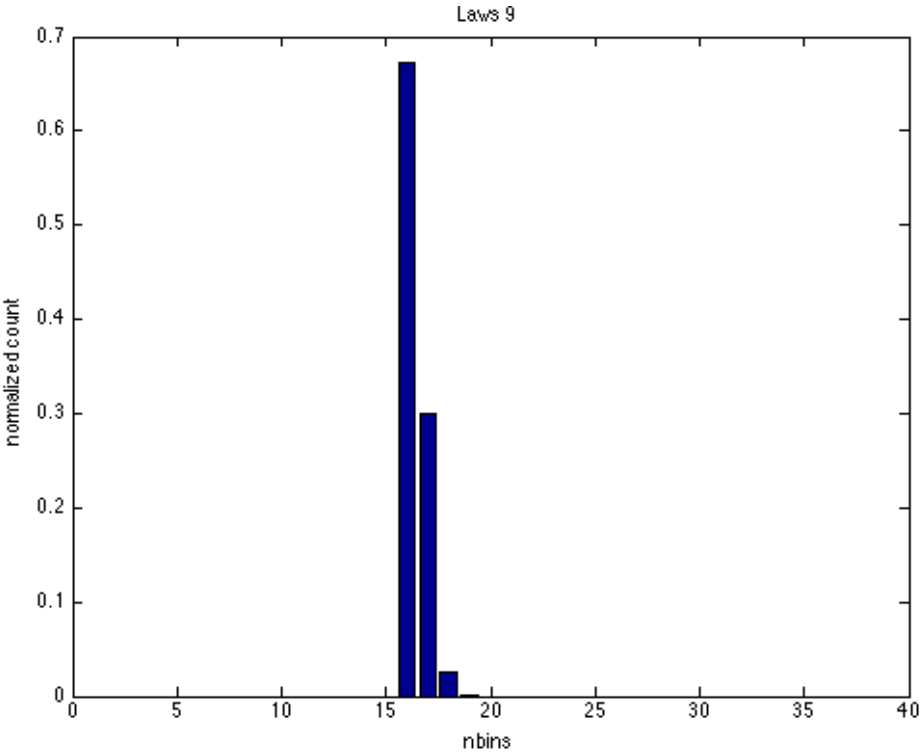


Figure i. Laws' 9th Texture Histogram

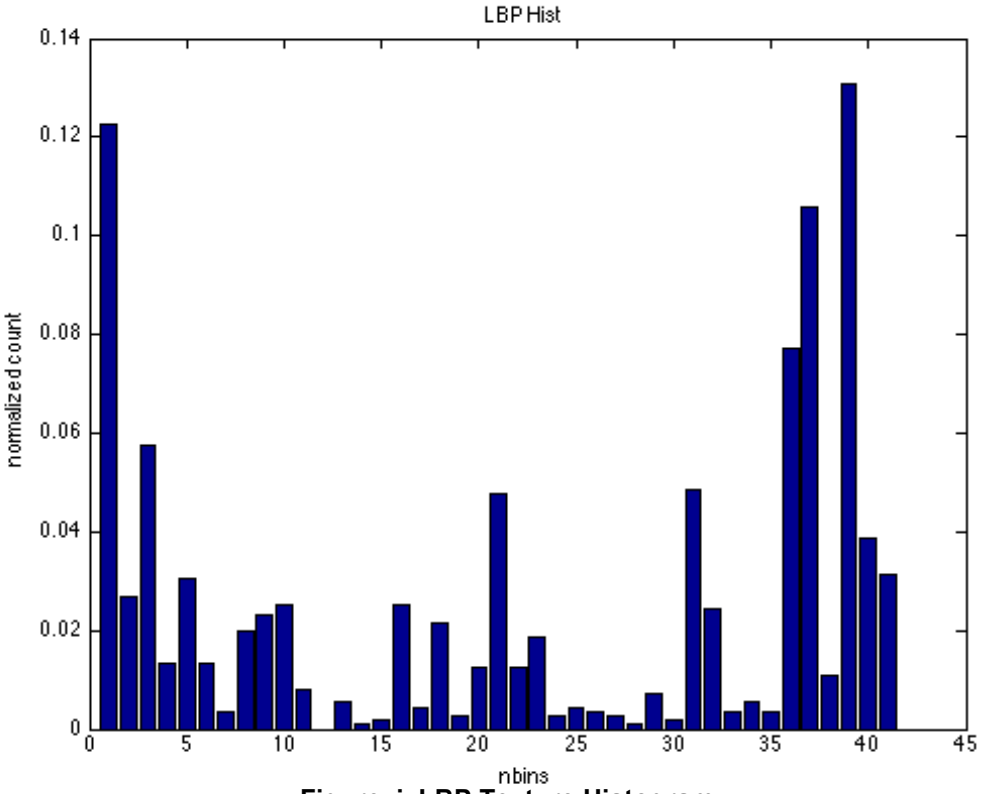


Figure j. LBP Texture Histogram

3. Actual Code Depth Extraction Scheme

```
HRESULT getDepthData(Mat* dest) {
    dest->create(height, width, CV_16UC1);
    HRESULT hr = S_OK;

    NUI_IMAGE_FRAME imageFrame;
    hr = sensor->NuiImageStreamGetNextFrame(depthStream, 0, &imageFrame);
    if (FAILED(hr))
    {
        return hr;
    }

    depthTimeStamp = imageFrame.liTimeStamp;
    BOOL nearMode=TRUE;
    INuiFrameTexture* pTexture;

    NUI_LOCKED_RECT LockedRect;
    imageFrame.pFrameTexture->LockRect(0, &LockedRect, NULL, 0);
    sensor->NuiImageGetColorPixelCoordinateFrameFromDepthPixelFrameAtResolution(NUI_IMAGE_RESOLUTION_640x480,
        NUI_IMAGE_RESOLUTION_640x480,
        width*height,
        (USHORT *)(LockedRect.pBits),
        width*height * 2,
        colorCoordinates);
    imageFrame.pFrameTexture->UnlockRect(0);

    hr = sensor->NuiImageFrameGetDepthImagePixelFrameTexture(depthStream, &imageFrame, &nearMode, &pTexture);
    if (FAILED(hr))
    {
        return hr;
    }
    pTexture->LockRect(0, &LockedRect, NULL, 0);
    USHORT *p;
    if (LockedRect.Pitch != 0) {

        NUI_DEPTH_IMAGE_PIXEL* pPixelRun = (NUI_DEPTH_IMAGE_PIXEL*)LockedRect.pBits;
            for (int j = 0; j < height; ++j) {      //Iterate by Rows
                p = dest->ptr<USHORT>(j);
                for (int i = 0; i < width; ++i) {       //Iterate by Cols
                    // Get depth of pixel in millimeters
                    USHORT realDepth = pPixelRun->depth;
                    p[i] = realDepth;
                ++pPixelRun;
            }

        }
```

4. Actual Code RGB Extraction

```
HRESULT getRgbData(Mat* dest) {

    HRESULT hr = S_OK;
    NUI_IMAGE_FRAME imageFrame;
    NUI_LOCKED_RECT LockedRect;
    hr = sensor->NuiImageStreamGetNextFrame(rgbStream, 0, &imageFrame);
    if (FAILED(hr))
    {
        return hr;
    }

    colorTimeStamp = imageFrame.liTimeStamp;
    dest->create(height, width, CV_8UC3);
    INuiFrameTexture* texture = imageFrame.pFrameTexture;
    texture->LockRect(0, &LockedRect, NULL, 0);
    if (LockedRect.Pitch != 0) {
        //const BYTE* start = (const BYTE*)LockedRect.pBits;
        const BYTE* curr = (const BYTE*)LockedRect.pBits;
        uchar *p;
        for (int j = 0; j < height; ++j) {
            p = dest->ptr<uchar>(j);
            for (int i = 0; i < width; ++i) {
                // Determine rgb color for each depth pixel
                int depthIndex = i + j * width;
                long x = colorCoordinates[depthIndex * 2];
                long y = colorCoordinates[depthIndex * 2 + 1];
                // If out of bounds, then don't color it at all
                if (x < 0 || y < 0 || x >= width || y >= height)
                    continue;
                else {
                    const BYTE* curr = (const BYTE*)LockedRect.pBits + (x + width*y) * 4;

                    for (int n = 0; n < 4; ++n) {
                        if (n == 3) curr++;
                        else p[3 * i + n] = *(curr++);
                    }

                }
            }
        }
    }
    texture->UnlockRect(0);
    sensor->NuiImageStreamReleaseFrame(rgbStream, &imageFrame);

    return hr;
}
```

5. SVM training Method

```
double MLalgos::trainSVM (const cv::Mat& feat, const cv::Mat& truthDepth,
std::string name,cv::Mat& confMat, FuzzyOptions opts) {

   CvSVMParams params;
   params.svm_type = opts.svm_type;
   params.C = opts.svmC;
   params.kernel_type = opts.svmKernel;
   params.gamma = opts.svmGamma;
   params.term_crit = opts.svmTermCrit;
   params.nu = opts.svmNu;
   params.p = opts.svmP;

   int nSamples = feat.rows;
```

```
    std::vector<double> para(2,0);

    //Fix Target output
    parseMinMax(opts, &para[0], &para[1]);
    cv::Mat outputs(nSamples,1,CV_32FC1);
    std::vector<MRFevaluate::LEVEL> vecTargs;
    for (auto i=0; i < nSamples; ++i) {
        double confLevelTarg;
        MRFevaluate::LEVEL targs = fuzzifyTriangular(truthDepth.at<double>(i),
para, &confLevelTarg);
        if (targs!=MRFevaluate::BADARGS) {
            outputs.at<float>(i) = (int)targs;
        }
        else {
            std::cout << "Fuzzification BADARGS!! check your code again!! " <<
std::endl;
            return 0;
        }
        vecTargs.push_back(targs);
    }
    //Training
    std::cout << "training SVM model..." << std::endl;
    CvSVM svm;
    cv::Mat featFloat, featMat;
    feat.convertTo(featFloat, CV_32F);
    cv::hconcat(cv::Mat::ones(nSamples, 1, CV_32F), featFloat, featFloat);
    if (!opts.isNormalized) {
        //Normalize it!
        featMat = featFloat.clone();
        for (auto i = 0 ; i < featFloat.rows; ++i) {
            cv::Mat temp;
            cv::normalize(featFloat.row(i), temp, opts.normType);
            temp.copyTo(featMat.row(i));
        }
    }
    else
        featFloat.copyTo(featMat);

    svm.train_auto(featMat, outputs, cv::Mat(), cv::Mat(), params);
//    svm.train(featMat, outputs, cv::Mat(), cv::Mat(), params);
    std::cout << "Finished training process" << std::endl;

    //04/22/15: Edited for Graphical Visualization
```

```
    //Get Training Error
    std::vector<MRFevaluate::LEVEL> vecEst;
    for (auto j = 0; j < nSamples; ++j) {
        float estClass = svm.predict(featMat.row(j)); //Either 1,2 or 3
        MRFevaluate::LEVEL estLevel;
        if (round(estClass) - MRFevaluate::LOW < FLT_EPSILON)
            estLevel = MRFevaluate::LOW;
        else if (roundf(estClass) - MRFevaluate::MEDIUM < FLT_EPSILON)
            estLevel = MRFevaluate::MEDIUM;
        else if (round(estClass) - MRFevaluate::HIGH < FLT_EPSILON)
            estLevel = MRFevaluate::HIGH;
        else {
            std::cerr  << "ERROR: estimating Pressure LEVEL." << std::endl;
            estLevel = MRFevaluate::BADARGS;
        }
        vecEst.push_back(estLevel);
    }

          svm.save(list.setModelName(name, "SVM").c_str());
    return plotConfusionMatrix(vecEst, vecTargs, confMat);
}
```

6. Gradient Boosted Trees Training Method

```
double MLalgos::trainGBoost (const cv::Mat& feat,
                             const cv::Mat& truthDepth,
                             std::string name,
                             cv::Mat& confuse,
                             FuzzyOptions opts) {

    CvGBTreesParams  params;
    params.loss_function_type = opts.GBTfunctType; // loss_function_type
    params.weak_count         = opts.GBTweak_count; // weak_count
    params.shrinkage          = opts.GBTshrinkage; // shrinkage
    params.subsample_portion  = opts.GBTsubsampleRatio; //
subsample_portion
    params.max_depth          = opts.GBTmax_depth; // max_depth
    params.use_surrogates     = opts.GBTuse_surrogates; // use_surrogates )
    int nSamples = feat.rows;
    std::vector<double> para(2,0);

    //Fix Target output
    parseMinMax(opts, &para[0], &para[1]);
    cv::Mat outputs(nSamples,1,CV_32FC1);
```

```
    std::vector<MRFevaluate::LEVEL> vecTargs;
    for (auto i=0; i < nSamples; ++i) {
        double confLevelTarg;
        MRFevaluate::LEVEL targs = fuzzifyTriangular(truthDepth.at<double>(i),
para, &confLevelTarg);
        if (targs!=MRFevaluate::BADARGS)
            outputs.at<float>(i) = (int)targs;
        else {
            std::cout << "Fuzzification BADARGS!! check your code again!! " <<
std::endl;
            return 0;
        }
        vecTargs.push_back(targs);
    }
    // Prepare Training Data
    std::cout << "training the Gradient Boosted Trees..." << std::endl;
    cv::Mat featFloat, featMat;
    feat.convertTo(featFloat, CV_32F);
    cv::hconcat(cv::Mat::ones(nSamples, 1, CV_32F), featFloat, featFloat);
    if (!opts.isNormalized) {
        //Normalize it!
        featMat = featFloat.clone();
        for (auto i = 0 ; i < featFloat.rows; ++i) {
            cv::Mat temp;
            cv::normalize(featFloat.row(i), temp, opts.normType);
            temp.copyTo(featMat.row(i));
        }
    }
    else
        featFloat.copyTo(featMat);

    // learn classifier
    CvGBTrees gbtrees;
    gbtrees.train( featMat, CV_ROW_SAMPLE, outputs, cv::Mat(), cv::Mat(),
cv::Mat(), cv::Mat(), params );

    //Get Training Error
    std::vector<MRFevaluate::LEVEL> vecEst;
    for (auto j = 0; j < nSamples; ++j) {
        float estClass = gbtrees.predict(featMat.row(j)); //Either 1,2 or 3
        MRFevaluate::LEVEL estLevel;
        if (round(estClass) - MRFevaluate::LOW < FLT_EPSILON)
            estLevel = MRFevaluate::LOW;
```

```
        else if (roundf(estClass) - MRFevaluate::MEDIUM < FLT_EPSILON)
            estLevel = MRFevaluate::MEDIUM;
        else if (round(estClass) - MRFevaluate::HIGH < FLT_EPSILON)
            estLevel = MRFevaluate::HIGH;
        else {
            std::cerr  << "ERROR: estimating Pressure LEVEL." << std::endl;
            estLevel = MRFevaluate::BADARGS;
        }
        vecEst.push_back(estLevel);
    }

    gbtrees.save(list.setModelName(name, "GBT").c_str());
    return plotConfusionMatrix(vecEst, vecTargs, confuse);
}
```

7. Artificial Neural Network Training Method

```
///@param feat - a matrix of nSamples x nFeature containing the feature vectors
double MLalgos::trainANN (const cv::Mat& feat, const cv::Mat&
truthDepth,std::string name,cv::Mat& confuse,FuzzyOptions opts) {

    //Set Initializiation
    int class_count = opts.class_count;
    int train_hr = 0;
    int method = opts.method;
    double method_param = opts.method_param;
    int nInnerPerceptrons = opts.nInnerLayerPerceptrons;
    int max_iter = opts.max_iter;
    int nSamples = feat.rows;
    int nFeatures = feat.cols+1;
    cv::Mat outputs(nSamples,class_count,CV_32FC1);
    std::vector<MRFevaluate::LEVEL> vecTarget, vecEstimate;

    int lsize[] = {nFeatures,nInnerPerceptrons,class_count};
    cv::Mat layer_size = cv::Mat((int)(sizeof(lsize)/sizeof(lsize[0])),1, CV_32S,
lsize).clone();


    //Set Parameters of MLP
    CvANN_MLP mlp;
    mlp.create(layer_size);

    //Load Data
```

```
    if (truthDepth.cols!=1) {
        std::cout << "ERROR: truthDepth should be a column vector!" << std::endl;
        return 1000;
    }
    std::vector<double> para(2,0);
    parseMinMax(opts, &para[0], &para[1]);

    for (auto i=0; i < nSamples; ++i) {
        double confLevelTarg;
        int targs = fuzzifyTriangular(truthDepth.at<double>(i), para,
&confLevelTarg);
        if (targs!=MRFevaluate::BADARGS) {
            outputs.at<float>(i,targs-1)++;
        }
    }

    //Train Data
    std::cout << "Training the ANN_MLP..." << std::endl;
    cv::Mat featFloat,featMat;
    feat.convertTo(featFloat, CV_32F);
    cv::hconcat(cv::Mat::ones(nSamples, 1, CV_32F), featFloat, featFloat);
          featFloat.copyTo(featMat);

    //Each input vector should be in each row and floating point.
    mlp.train(featMat, outputs,
cv::Mat(),cv::Mat(),CvANN_MLP_TrainParams(cvTermCriteria(CV_TERMCRIT
_ITER,max_iter,0.01),
                                                    method, method_param));

    //Predict ANN and calculate Training Error
    cv::Mat mlp_response(1,class_count,CV_32F);
    for (auto i = 0  ; i < nSamples; ++i) {
        mlp.predict(featMat.row(i), mlp_response);
        cv::Point maxPt;
        double best_class;
        cv::minMaxLoc(mlp_response, 0,0,0,&maxPt,cv::Mat());
        best_class = maxPt.x + 1;


        double confLevelTarg,confLevelEst;
        MRFevaluate::LEVEL targetLevel  =
fuzzifyTriangular(truthDepth.at<double>(i), para, &confLevelTarg);
        int r = fabs(best_class - targetLevel) < FLT_EPSILON ? 1 : 0;
```

```
        int bestInt = (int)round(best_class);
        MRFevaluate::LEVEL estLevel;
        if (bestInt - MRFevaluate::LOW < FLT_EPSILON)
            estLevel = MRFevaluate::LOW;
        else if (bestInt - MRFevaluate::MEDIUM < FLT_EPSILON)
            estLevel = MRFevaluate::MEDIUM;
        else if (bestInt - MRFevaluate::HIGH < FLT_EPSILON)
            estLevel = MRFevaluate::HIGH;
        else {
            std::cerr  << "ERROR: estimating Pressure LEVEL." << std::endl;
            estLevel = MRFevaluate::BADARGS;
        }

        vecEstimate.push_back(estLevel);
        vecTarget.push_back(targetLevel);
        train_hr += r;
    }
    double acc = plotConfusionMatrix(vecEstimate, vecTarget, confuse);
    mlp.save(list.setModelName(name, "ANN").c_str());
    return ((double)train_hr/nSamples);
}
```

8. Multiple Linear Regression Code (using Gurobi C++ Library)

```
std::vector<double> MRFevaluate::trainLineGurobi(std::vector<
std::vector<double> >& vecFeat, std::vector<double>& gndDepth) {

    for (auto i = 0 ; i < vecFeat.size(); ++i) {
        if (vecFeat[i].size() != gndDepth.size()) {
            std::cerr << "Features " << i << "doesn't have equal size to gndDepth!!"
<< std::endl;
        }
    }

    int nVars = (int)vecFeat.size() + 1;
    int nSamples = (int)vecFeat[0].size();

    std::vector<double> outputs;
    try {
        GRBEnv env = GRBEnv();
        GRBModel model = GRBModel(env);

        //Create Variables
```

```
        GRBVar *theta = new GRBVar[nVars];
        for (auto i =0; i < nVars; ++i) {
            theta[i] = model.addVar(-GRB_INFINITY, GRB_INFINITY, NULL,
GRB_CONTINUOUS);
        }
        GRBVar *err = model.addVars(nSamples);
        model.update();


        for (int i = 0 ; i < nSamples; ++i) {
            GRBLinExpr constrnt = 0.0;
            cv::Mat coeff = cv::Mat::ones(1, 1, CV_64FC1);
            for (auto k=0; k < nVars-1; ++k) {
                coeff.push_back(vecFeat[k].at(i));
            }
            cv::normalize(coeff, coeff, NORM_L1);
            for (auto j = 0; j < nVars; ++j) {
                constrnt += coeff.at<double>(j) * theta[j];
            }
            model.addConstr(constrnt - gndDepth[i] <=  1.0 * err[i]);
            model.addConstr(constrnt - gndDepth[i] >= -1.0 * err[i]);
        }

        // Integrate new variables
        model.update();

        GRBLinExpr obj = 0.0;
        for (auto i = 0 ; i < nSamples; ++i) {
            obj += err[i];
        }

        model.setObjective(obj,GRB_MINIMIZE);
        model.update();

        model.optimize();
        if (model.get(GRB_IntAttr_SolCount) > 0) {
            double *thi = new double(nVars);
            thi = model.get(GRB_DoubleAttr_X,theta,nVars);
            for (auto n=0; n < nVars; ++n) {
                outputs.push_back(thi[n]);
            }
            delete [] thi;
```

```
        }

        int optimstatus = model.get(GRB_IntAttr_Status);

        if (optimstatus == GRB_INF_OR_UNBD) {
            model.getEnv().set(GRB_IntParam_Presolve, 0);
            model.optimize();
            optimstatus = model.get(GRB_IntAttr_Status);
        }

        if (optimstatus == GRB_OPTIMAL) {
            double objval = model.get(GRB_DoubleAttr_ObjVal);
            std::cout << "Optimal objective: " << objval << std::endl;
        } else if (optimstatus == GRB_INFEASIBLE) {
            std::cout << "Model is infeasible" << std::endl;

            // compute and write out IIS

            model.computeIIS();
            model.write("model.ilp");
        } else if (optimstatus == GRB_UNBOUNDED) {
            std::cout << "Model is unbounded" << std::endl;
        } else {
            std::cout << "Optimization was stopped with status = "
            << optimstatus << std::endl;
        }

        delete [] theta;
        delete [] err;


    } catch(GRBException e) {
        std::cout << "Error code = " << e.getErrorCode() << std::endl;
        std::cout << e.getMessage() << std::endl;
    } catch (...) {
        std::cout << "Error during optimization" << std::endl;
    }
    //toni is handsome

    return outputs;

}
```

References

[1] E. Mohammadi, E. P. Dadios, L. A. G. Lim, M. K. Cabatuan, R. N. G. Naguib, J. M. C. Avila, and A. Oikonomou, "Real-Time Evaluation of Breast Self-Examination Using Computer Vision," *Int. J. Biomed. Imaging*, vol. 2014, 2014.

[2] R. A. A. Masilang, "Design and Development of a Computer Vision-based Breast Self-Examination Instruction and Supervision System," De La Salle University, 2014.

[3] American Cancer Society, "Breast Cancer," 2013. [Online]. Available: http://www.cancer.org/acs/groups/cid/documents/webcontent/003090-pdf.pdf.

[4] BreastCancer.org, "U.S. Breast Cancer." [Online]. Available: http://www.breastcancer.org/ symptoms/understand_bc/statistics.

[5] World Health Organization, "Breast cancer: prevention and control," 2013. [Online]. Available: http://www.who.int/cancer/ detection/breastcancer/en/index1.html.

[6] B. O. Anderson, R. Shyyan, A. Eniu, R. A. Smith, C.-H. Yip, N. S. Bese, L. W. C. Chow, S. Masood, S. D. Ramsey, and R. W. Carlson, "Breast cancer in limited-resource countries: an overview of the Breast Health Global Initiative 2005 guidelines.," *Breast J.*, vol. 12 Suppl 1, pp. S3–15, 2006.

[7] Cancercare, "cancercare.org," 2009. [Online]. Available: ttp://www.cancercare.org.

[8] "Breast Cancer," 2012. [Online]. Available: http://www.healthcentral.com/breast-cancer/c/question/186694/149651.

[9] Trading Economics, "Trading Economics," 2012. [Online]. Available: http://www.tradingeconomics.com/philippines/gdp-per-capita.

[10] A. M. Chiarelli, V. Majpruz, P. Brown, M. Thériault, R. Shumak, and V. Mai, "The contribution of clinical breast examination to the accuracy of

breast screening.," *J. Natl. Cancer Inst.*, vol. 101, no. 18, pp. 1236–43, Sep. 2009.

[11] N. S. Weiss, "Breast cancer mortality in relation to clinical breast examination and breast self-examination.," *Breast J.*, vol. 9, no. 2, pp. S86–S89, 2003.

[12] P. Pisani, D. M. Parkin, C. Ngelangel, D. Esteban, L. Gibson, M. Munson, M. G. Reyes, and A. Laudico, "Outcome of screening by clinical examination of the breast in a trial in the Philippines.," *Int. J. Cancer*, vol. 118, no. 1, pp. 149–54, Jan. 2006.

[13] T. McCready, D. Littlewood, and J. Jenkinson, "Breast self-examination and breast awareness: a literature review.," *J. Clin. Nurs.*, vol. 14, no. 5, pp. 570–8, May 2005.

[14] D. L. Gao, D. B. Thomas, R. M. Ray, W. W. Wang, C. J. Allison, F. L. Chen, P. Porter, Y. W. Hu, G. L. Zhao, L. Da Pan, W. Li, C. Wu, Z. Coriaty, I. Evans, M. G. Lin, H. Stalsberg, and S. G. Self, "Randomized trial of breast self-examination in 266,064 women in Shanghai," *Zhonghua Zhong Liu Za Zhi*, pp. 350–354, 2005.

[15] W. W. M. Lam, C. P. Chan, C. F. Chan, C. C. C. Mak, K. W. H. Chong, M. H. J. Leung, and M. H. Tang, "Factors affecting the palpability of breast lesion by self-examination.," *Singapore Med. J.*, vol. 49, no. 3, pp. 228–32, Mar. 2008.

[16] MedicineNet.com, "MedicineNet.com," 2004. [Online]. Available: http://www.medicinenet.com/script/main/art.asp?articlekey=9695.

[17] H. L. Howe, "Proficiency in performing breast self-examination," *Patient Couns. Health Educ.*, vol. 2, no. 4, pp. 151–153, 1980.

[18] M. Cabatuan and E. Dadios, "Computer vision-based breast self-examination stroke position and palpation pressure level classification using artificial neural networks and wavelet transforms," in *IEEE Engineering in Medicine and Biology Society*, 2012, pp. 6259–62.

[19] S. Chen, R. G. Naguib, and A. Oikonomu, "Hand pressure estimation by image entropy for a real-time breast self-examination multimedia system.," in *IEEE Engineering in Medicine and Biology Society*, 2005, pp. 1732–1735.

[20] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: learning 3D scene structure from a single still image.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–40, May 2009.

[21] S. Manoharan and P. Pugalendhi, "Breast Cancer: An Overview," *J. Cell Tissue Res.*, vol. 10, no. 3, pp. 2423–2432, 2010.

[22] S. G. Comen, "The Clinical Breast Exam: A Refresher for Family Medicine Residents," 2012. [Online]. Available: http://www.youtube.com/watch?v=1hya8RPM70k.

[23] A. Oikonomou, S. Amin, R. N. G. Naguib, A. Todman, and H. Al-Omishy, "Breast Self Examination Training Through the Use of Multimedia:A Prototype Multimedia Application," in *IEEE Engineering in Medicine and Biology Society*, 2003, pp. 1295–1298.

[24] A. Oikonomou and S. Amin, "IRiS: an interactive reality system for breast self-examination training," in *IEEE Engineering in Medicine and Biology Society*, 2004, pp. 5162–5165.

[25] Y. Hu and R. N. G. Naguib, "Search strategies for the automatic delineation of the breast area in a multimedia breast self-examination system," in *IEEE Engineering in Medicine and Biology Society*, 2003, pp. 1315–1318.

[26] Y. Hu and R. N. G. Naguib, "Hand motion segmentation against skin colour background in breast awareness applications," in *IEEE Engineering in Medicine and Biology Society*, 2004, pp. 3221–3224.

[27] D. Heeger, "Perception Lecture Notes: Depth, Size, and Shape," 2006. [Online]. Available: http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/depth/depth-size.html.

[28] P. a. Warren and S. K. Rushton, "Perception of scene-relative object movement: Optic flow parsing and the contribution of monocular depth cues," *Vision Res.*, vol. 49, no. 11, pp. 1406–1419, Jun. 2009.

[29] C. Wang, N. Komodakis, and N. Paragios, "Markov Random Field modeling, inference & learning in computer vision & image understanding: A survey," *Comput. Vis. Image Underst.*, vol. 117, no. 11, pp. 1610–1627, Nov. 2013.

[30] G. Palou and P. Salembier, “Occlusion-Based Depth Ordering on Monocular Images with Binary Partition Tree,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1093–1096.

[31] G. Palou and P. Salembier, “Monocular depth ordering using T-junctions and convexity occlusion cues.,” *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1926–1939, 2013.

[32] X. He and A. Yuille, “Occlusion boundary detection using pseudo-depth,” *Comput. Vision–ECCV 2010*, pp. 1–14, 2010.

[33] A. Saxena, S. Chung, and A. Ng, “Learning depth from single monocular images,” *NIPS*, 2005.

[34] A. Saxena, S. H. Chung, and A. Y. Ng, “3-D Depth Reconstruction from a Single Still Image,” *Int. J. Comput. Vis.*, vol. 76, no. 1, pp. 53–69, Aug. 2007.

[35] A. Saxena, M. Sun, and A. Ng, “Make3D: Depth Perception from a Single Still Image.,” *AAAI*, pp. 1571–1576, 2008.

[36] C. Jung and C. Kim, “Real-time estimation of 3D scene geometry from a single image,” *Pattern Recognit.*, vol. 45, no. 9, pp. 3256–3269, Sep. 2012.

[37] A. Wedel, U. Franke, J. Klappstein, T. Brox, and D. Cremers, “Realtime depth estimation and obstacle detection from monocular video,” *Pattern Recognit.*, pp. 475–484, 2006.

[38] a. Torralba and a. Oliva, “Depth estimation from image structure,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1226–1238, Sep. 2002.

[39] a. Mitiche and S. Hadjres, “MDL estimation of a dense map of relative depth and 3D motion from a temporal sequence of images,” *Pattern Anal. Appl.*, vol. 6, no. 1, pp. 78–87, Apr. 2003.

[40] H. Sekkati and A. Mitiche, “A variational method for the recovery of dense 3D structure from motion,” *Rob. Auton. Syst.*, vol. 55, no. 7, pp. 597–607, Jul. 2007.

[41] D. Sun, E. Sudderth, and M. Black, “Layered image motion with explicit occlusions, temporal consistency, and depth ordering,” *NIPS*, pp. 1–9, 2010.

[42] J. Michels, A. Saxena, and A. Ng, “High speed obstacle avoidance using monocular vision and reinforcement learning,” *22nd Int. Conf. Mach. Learn.*, 2005.

[43] E. B. Sudderth, a. Torralba, W. T. Freeman, and a. S. Willsky, “Depth from Familiar Objects: A Hierarchical Model for 3D Scenes,” *2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Vol. 2*, vol. 2, pp. 2410–2417, 2006.

[44] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[45] B. Liu, S. Gould, and D. Koller, “Single image depth estimation from predicted semantic labels,” *2010 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1253–1260, Jun. 2010.

[46] K. Karsch, C. Liu, and S. Kang, “Depth extraction from video using non-parametric sampling,” *Comput. Vision–ECCV 2012*, no. Sec 5, pp. 1–14, 2012.

[47] E. Delage and a. Y. Ng, “A Dynamic Bayesian Network Model for Autonomous 3D Reconstruction from a Single Indoor Image,” *2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Vol. 2*, vol. 2, pp. 2418–2428, 2006.

[48] O. Barinova, V. Konushin, and A. Yakubenko, “Fast automatic single-view 3-d reconstruction of urban scenes,” *Comput. Vision–ECCV …*, pp. 100–113, 2008.

[49] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: a factorization method,” *Int. J. Comput. Vis.*, vol. 154, pp. 137–154, 1992.

[50] C. J. Poelman and T. Kanade, “A paraperspective factorization method for shape and motion recovery,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 206–218, 1997.

[51] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross, "Scene reconstruction from high spatio-angular resolution light fields.," *ACM Trans. Graph.*, vol. 32, no. 4, p. 73, 2013.

[52] V. Nedovic, A. Smeulders, A. Redert, and J. Geusebroek, "Depth Information by Stage Classification.," *ICCV*, 2007.

[53] S. Battiato, S. Curti, M. La Cascia, M. Tortora, and E. Scordato, "Depth Map generation by image classification," in *SPIE*, 2004, pp. 95–104.

[54] M. Dimiccoli and P. Salembier, "Exploiting T-Junctions For Depth Segregation in Single Images," *IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 1229–1232, 2009.

[55] G. Palou and P. Salembier, "From local occlusion cues to global monocular depth estimation," *2012 IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 793–796, Mar. 2012.

[56] J.-M. Morel and P. Salembier, "Monocular Depth by Nonlinear Diffusion," *2008 Sixth Indian Conf. Comput. Vision, Graph. Image Process.*, pp. 95–102, Dec. 2008.

[57] R. C. Gonzales, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*, 2nd ed. Gatesmark Publishing, 2004.

[58] BorisFx, "Continuum Complete." [Online]. Available: http://www.borisfx.com/avid/bccavx/classic_features.php.

[59] R. Rojas, "Lucas-Kanade in a Nutshell." [Online]. Available: http://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/documents/tutorials/Lucas-Kanade2.pdf.

[60] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Imaging*, vol. 130, pp. 674–679, 1981.

[61] D. Fleet and Y. Weiss, "Optical Flow Estimation," in *Mathematical Models in Computer Vision: The Handbook*, Springer, 2005, pp. 239–258.

[62] MathWorks, "Documentation Center," 2012. [Online]. Available: http://www.mathworks.com/help/vision/ref/opticalflow.html#bqi5zaf-1.

[63] W. Hayt and J. Buck, *Engineering Electromagnetics*, 8th ed. McGraw-Hill, Inc., 2012.

[64] R. Szeliski, *Computer vision: algorithms and applications*. Springer, 2010.

[65] V. Eruhimov, “Stereo Matching Calibration.” 2010.

[66] OpenCV, “Real Time pose estimation of a textured object,” 2015. [Online]. Available: http://docs.opencv.org/master/dc/d2c/tutorial_real_time_pose.html. [Accessed: 06-May-2015].

[67] K. Khoshelham and S. O. Elberink, “Accuracy and resolution of Kinect depth data for indoor mapping applications.,” *Sensors (Basel).*, vol. 12, no. 2, pp. 1437–54, Jan. 2012.

[68] Xbox Support, “Kinect sensor for Xbox 360 components,” 2015. [Online]. Available: http://support.xbox.com/en-US/xbox-360/kinect/kinect-sensor-components. [Accessed: 13-Apr-2015].

[69] E. Salvador, A. Cavallaro, and T. Ebrahimi, “Cast shadow segmentation using invariant color features,” *Comput. Vis. Image Underst.*, vol. 95, no. 2, pp. 238–259, Aug. 2004.

[70] K. I. Laws, “Rapid Texture Identification,” in *Proc. SPIE 0238, Image Processing for Missile Guidance*, 1980, vol. 0238, pp. 376–381.

[71] L. Shapiro and G. Stockman, *Computer Vision*, vol. 2004, no. October. Prentice Hall, 2001.

[72] P. Wagner, “Local Binary Patterns,” 2011. [Online]. Available: http://www.bytefish.de/blog/local_binary_patterns/. [Accessed: 26-Apr-2015].

[73] I. . H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, Third. Elsevier, 2005.

[74] OpenCV, “Introduction to Support Vector Machines,” 2015. [Online]. Available: http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html. [Accessed: 26-Apr-2015].

[75] A. Ng, “Machine Learning,” 2013. [Online]. Available: https://www.coursera.org/learn/machine-learning.

[76] C. M. Bishop, *Pattern recognition and machine learning*, vol. 4, no. 4. Springer, 2006.

[77] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001*, vol. 1, pp. I–511–I–518, 2001.

[78] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Ann. Stat.*, pp. 1189–1232, 2001.

[79] L. Q. Doan, “Photo Gallery Breast Augmentation,” 2013. [Online]. Available: http://drluudoan.luudoancosmeticinstitute.com/?page_id=792.

[80] J. A. Perlman, “Breast Enlargement to a D cup size with silicone breast implants,” 2012. [Online]. Available: http://www.plasticsurgerybeverlyhills.com/images/Breast-enhancement-March11A.jpg.

[81] B. P. Dickinson, “Breast Augmentation Full C Small D,” 2014. [Online]. Available: http://www.drbriandickinson.com/blog/breast-augmentation-full-c-small-d_17.html.

[82] D. Revis, “Breast Augmentation Before & After Photos.” [Online]. Available: http://www.southfloridaplasticsurgery.com/before-after-photos/breast-augmentation.html.

[83] The Wizard of Bras, “Cup Size Chart.” [Online]. Available: http://www.wizardofbras.com/cupsizechart.aspx.

[84] E. N. Mohammadi, “Design and development of the computer vision algorithm for a real-time breast self-examination,” De La Salle University, 2014.

[85] N. Silberman and R. Fergus, “Indoor Scene Segmentation using a Structured Light Sensor,” in *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition*, 2011.

[86] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in *Computer Vision--ECCV 2012*, Springer, 2012, pp. 746–760.

[87] J. Xiao, A. Owens, and A. Torralba, “SUN3D: A database of big spaces reconstructed using sfm and object labels,” in *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2013, pp. 1625–1632.

[88] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, “A category-level 3d object dataset: Putting the kinect to work,” in *Consumer Depth Cameras for Computer Vision*, Springer, 2013, pp. 141–165.

[89] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, “Contextually guided semantic labeling and search for three-dimensional point clouds,” *Int. J. Rob. Res.*, p. 0278364912461538, 2012.

[90] Q.-Y. Zhou and V. Koltun, “Dense scene reconstruction with points of interest,” *ACM Trans. Graph.*, vol. 32, no. 4, p. 112, 2013.

[91] Manila Doctors Hospital, “Manila Doctors Hospital,” 2015. [Online]. Available: http://www.maniladoctors.com.ph/doctors/joson-reynaldo-o/. [Accessed: 30-Apr-2015].

[92] N. C. I. U. S. National Institutes of Health, “SEER Training Modules, Quadrants of The Breast.” [Online]. Available: http://training.seer.cancer.gov/. [Accessed: 13-Apr-2015].

[93] S. Douglas, “Using the Roto Brush Tool in Adobe After Effects CS5,” 2010. [Online]. Available: http://www.kenstone.net/fcp_homepage/rotobrush_ae_cs5_douglas.html. [Accessed: 30-Apr-2015].

[94] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[95] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” *IEEE Trans. Image Process.*, vol. 19, pp. 1635–1650, 2010.

[96] Kinect for Windows Team, “Near Mode: What it is (and isn’t),” *MSDN Blogs*, 2012. [Online]. Available:

http://blogs.msdn.com/b/kinectforwindows/archive/2012/01/20/near-mode-what-it-is-and-isn-t.aspx. [Accessed: 13-Apr-2015].

[97] S. Chen, C. Qicai, R. N. G. Naguib, and A. Oikonomou, “Hand Pressure Detection Among Image Sequence In Breast Self-Examination Multimedia System,” *2009 Int. Forum Inf. Technol. Appl.*, pp. 127–130, May 2009.

[98] T. Ahonen, a. Hadid, and M. Pietikainen, “Face Description with Local Binary Patterns: Application to Face Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, 2006.

[99] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 886–893, 2005.