

METHODOLOGY FOR EXTENSIVE EVALUATION OF
SEMIAUTOMATIC AND INTERACTIVE SEGMENTATION
ALGORITHMS USING SIMULATED INTERACTION MODELS

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
S M Rafizul Haque

©S M Rafizul Haque, August 2016. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

Performance of semiautomatic and interactive segmentation(SIS) algorithms are usually evaluated by employing a small number of human operators to segment the images. The human operators typically provide the approximate location of objects of interest and their boundaries in an interactive phase, which is followed by an automatic phase where the segmentation is performed under the constraints of the operator-provided guidance. The segmentation results produced from this small set of interactions do not represent the true capability and potential of the algorithm being evaluated. For example, due to inter-operator variability, human operators may make choices that may provide either overestimated or underestimated results. As well, their choices may not be realistic when compared to how the algorithm is used in the field, since interaction may be influenced by operator fatigue and lapses in judgement. Other drawbacks to using human operators to assess SIS algorithms, include: human error, the lack of available expert users, and the expense. A methodology for evaluating segmentation performance is proposed here which uses simulated Interaction models to programmatically generate large numbers of interactions to ensure the presence of interactions throughout the object region. These interactions are used to segment the objects of interest and the resulting segmentations are then analysed using statistical methods. The large number of interactions generated by simulated interaction models capture the variabilities existing in the set of user interactions by considering each and every pixel inside the entire region of the object as a potential location for an interaction to be placed with equal probability. Due to the practical limitation imposed by the enormous amount of computation for the enormous number of possible interactions, uniform sampling of interactions at regular intervals is used to generate the subset of all possible interactions which still can represent the diverse pattern of the entire set of interactions.

Categorization of interactions into different groups, based on the position of the interaction inside the object region and texture properties of the image region where the interaction is located, provides the opportunity for fine-grained algorithm performance analysis based on these two criteria. Application of statistical hypothesis testing make the analysis more accurate, scientific and reliable in comparison to conventional evaluation of semiautomatic segmentation algorithms. The proposed methodology has been demonstrated by two case studies through implementation of seven different algorithms using three different types of interaction modes making a total of nine segmentation applications to assess the efficacy of the methodology. Application of this methodology has revealed in-depth, fine details about the performance of the segmentation algorithms which currently existing methods could not achieve due to the absence of a large, unbiased set of interactions. Practical application of the methodology for a number of algorithms and diverse interaction modes have shown its feasibility and generality for it to be established as an appropriate methodology. Development of this methodology to be used as a potential application for automatic evaluation of the performance of SIS algorithms looks very promising for users of image segmentation.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisors Professor Mark Eramian and Professor Kevin Schneider with whom I have been fortunate to work with for the last few years. I would like to pay my heartiest thanks to them for their continuous support, guidance and motivation. They were always actively engaged in my research by providing advice and ideas. I am grateful to them for being always available to discuss any problem and providing effective suggestions. They were always updated and conscious about the progress by continuously monitoring the results and direction of my research. I could not have imagined having better advisors and mentors for my PhD study.

Besides my supervisors, I would like to thank my rest of the committee members: Professor Eric Neufeld, Professor Carl Gutwin, Professor Nathaniel Osgood and Professor Roger Pierson, for their insightful comments and encouragement. Their critical questions and valuable advice have motivated me to widen my research from various perspectives.

I am grateful to my supervisor Professor Mark Eramian and my fellow lab mates Arvie and Brittany Chan for their help to understand and solve many issues with image processing library 'LibImage' which was actually developed by them over the years.

I thank my fellow lab mates Jianning Chi and Xin Yi for the stimulating discussions, help for programming problems and for all the fun we have had in the last few years.

I am especially grateful to one of my supervisors Prof Kevin Schneider for his continued financial support which was a great help for me to concentrate on my research.

Last but not the least, I would like to thank my family: my parents, my wife, my daughters and to my sister and brother for supporting me mentally throughout my PhD study.

I dedicate my thesis to my parents S M Abdur Razzaque and Fazilatunnesa, for their endless love, support and encouragement

CONTENTS

Permission to Use	i
Abstract	ii
Acknowledgements	iii
Contents	v
List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Introduction	1
1.1.1 Image segmentation	1
1.1.2 Need for semiautomatic and interactive segmentation	2
1.2 Evaluation of SIS algorithms: standard approach	2
1.3 Potential deficiencies with accounting for humans in evaluation	3
1.4 Problem statement	4
1.5 Hypothesis	4
1.6 Contributions	5
1.7 Outline	6
2 Background and Literature Review	8
2.1 Background	8
2.1.1 Image Segmentation	8
2.2 Types of segmentation algorithms	8
2.2.1 Automatic segmentation	9
2.2.2 Semiautomatic and Interactive segmentation	9
2.2.3 Interaction in image segmentation	10
2.3 Types of input provided by user	10
2.3.1 Setting parameter values	11
2.3.2 Direct graphic input on the image	11
2.3.3 Selecting from pre-set options in a menu	16
2.4 Evaluation of interactive segmentation methods	16
2.4.1 Accuracy	16
2.4.2 Reproducibility	24
2.5 Review of related works	27
2.6 Review of variabilities in user interactions	28
3 Simulated Interaction Models	32
3.1 Overview of Methodology	32
3.1.1 Programmatic Generation of Interactions	34
3.1.2 Categorization of interactions	39
3.2 Evaluation methodologies	42
3.2.1 Traditional Methods	42
3.2.2 Mean segmentation accuracy within categories	43
3.2.3 Coefficient of variation for lightweight evaluation of reproducibility potential	43

4	Segmentation of follicles in ultrasound images: A case study of the proposed methodology using position of interaction for categorization	45
4.1	Interaction models for generating interactions within follicles	46
4.2	Categorization of interactions	48
4.3	Analysis of the segmentation results	49
4.3.1	Mean Segmentation Accuracy Within Interaction Categories	49
4.3.2	Analysis of Variance of Segmentation Accuracy Between Interaction Categories	55
4.3.3	Coefficient of Variation within Interaction Categories	59
4.3.4	Significance test of CV for pairs of interaction groups	64
4.4	Results and Discussion	66
4.5	Summary of results	70
5	Segmentation of follicles in ultrasound images: A case study of the proposed methodology using local image properties for interaction categorization	74
5.1	Image Texture Properties	74
5.1.1	Energy	75
5.1.2	Contrast	76
5.1.3	Correlation	76
5.1.4	Homogeneity	76
5.2	Evaluation of SIS algorithms following the methodology	77
5.2.1	Segmentation and Generation of Interactions	77
5.2.2	Categorization of interactions	77
5.2.3	Analysis of the results using statistical methods	80
5.2.4	Significance test of the results for the bins of image properties	87
5.3	Summary of results	89
6	Discussion	98
6.1	Significance of the results and observations	98
6.2	Efficacy of the methodology	99
6.2.1	Seed point example	99
6.2.2	Brush stroke example	101
6.2.3	Closed contour example	102
6.3	Relation between accuracy and spatial position of interactions	103
6.3.1	Impact of position of interactions on accuracy	103
6.3.2	Significance of this impact	104
6.4	Relation between accuracy and image properties	104
6.4.1	Existence of association between image properties and accuracy	104
6.4.2	Significance of this association	106
6.4.3	Open questions	106
6.5	Training human operators to use SIS algorithms	107
6.5.1	Effect of the methodology on users	107
6.5.2	Significance of this effect	107
6.6	Selection of interaction modes and SIS algorithms to be used in an application	109
6.6.1	Choice of algorithm and interaction mode	109
6.6.2	Open questions	109
6.6.3	Future work	109
6.7	Background interactions	110
6.7.1	Need for automatic generation	110
6.7.2	Future work	110
7	Conclusion	112
7.1	Contributions	112
7.2	Challenges	113
7.3	Open questions	114

LIST OF TABLES

2.1	Interactions inside the foreground object provided by each individual user and corresponding accuracy	31
4.1	List of the algorithms and interaction modes used in our experiments	45
4.2	P values of Dice, RMSD and HD regressions of three groups of interactions for all nine algorithms	56
4.3	P values of Dice, RMSD and HD CV between different groups of interactions for all nine algorithms	66
5.1	Features and bins used for seed point interaction	78
5.2	Features and bins used for brush stroke interaction	79
5.3	Features and bins used for closed contour interaction	79
5.4	Features and bins used for iso-contour interaction	79
5.5	Dice values of the segmentations generated by algorithms <i>GSC</i> , <i>GSCSeq</i> , <i>TRC</i> , <i>Onecut</i> and <i>GCBS</i> for brush stroke interactions categorized according to the thresholds of the eight feature values.	81
5.6	Segmentation result RMSD for algorithms that used brush strokes interactions	82
5.7	Segmentation result HD for algorithms that used brush strokes interactions	83
5.8	Segmentation results for <i>DRLSE</i> algorithm that uses closed contour interactions	84
5.9	Segmentation results for DRLSEIC algorithm that uses closed iso-contour interactions	85
5.10	Segmentation results for algorithms GCSP and GCnoSP that use seed point interactions	86
5.11	P values of the categories of feature values for brush stroke interaction	88
5.12	P values of the bins of feature values for closed contour interaction	88
5.13	P values of the bins of feature values for seed point interaction	89
6.1	Comparison between the proposed methodology and the standard method for seed point interactions.	100
6.2	Comparison between the proposed methodology and the standard method brush stroke interactions.	101
6.3	Comparison between the proposed methodology and the standard method.	102

LIST OF FIGURES

1.1	(a) An image containing objects of interest and (b) Segmented objects	1
2.1	(a) Demonstration of TP, FP, TN, FN where the segmentations overlap (b) Object of interest is totally surrounded by the segmented object. (c) Object of interest and segmented object are totally disjoint	17
2.2	Image A is very different from image B and very similar to image C.	24
2.3	Seed points provided by 19 individual users represented in different colours	29
2.4	Accuracy values for three different images for the seed points provided by 19 individual users	29
3.1	Some objects with star shapes obtained from [139]	32
3.2	An object with grids inside the region where each intersection point of the grids represents the position of a seed point.	35
3.3	(a) Straight brush strokes within object region	36
3.4	Example of a baseline with two ellipses having the centroids on it	38
3.5	Ellipses of different orientations obtained by rotating the axes	38
3.6	(a) An object with few generated closed contours inside the object region (b) An object with some of the generated iso contours inside the object region	39
3.7	Mean and standard deviation (sample) of Dice, RMSD and HD for all nine algorithms	42
4.1	(a) Piecewise cubic polynomial function used for determining the number of seed points for each follicle. (b) Programmatically generated seed points inside each follicle in an ultrasound image. (c) The large follicle is segmented once with each of the seed points shown while the remaining follicles' seed points (at the centroids of their regions) are held constant. This process is repeated for each other follicle in the image.	46
4.2	Straight brush strokes inside the follicle region	47
4.3	(a) Curved brush strokes inside the follicle (b) Curved brush strokes of three categories shown in different colours (c) Each follicle is segmented using each brush stroke from its set of all brush strokes exactly once while the brush strokes for any other follicle in the image are held constant	48
4.4	(a) Closed contour and (b) Iso-contour inside the follicle region	48
4.5	Mean and Standard Deviation of Dice with error bar for the overall, central, intermediate and peripheral group of interactions from the left to right order respectively in each row for nine algorithms <i>GCSP</i> , <i>GCBS</i> , <i>GSC</i> , <i>GSCSeq</i> , <i>TRC</i> , <i>Onecut</i> , <i>DRLSE</i> , <i>DRLSEIC</i> and <i>GCnoSP</i> from the top to bottom row order respectively.	51
4.6	Mean and Standard Deviation of RMSD with error bar for the overall, central, intermediate and peripheral group of interactions from left to the right order respectively in each row for nine algorithms <i>GCSP</i> , <i>GCBS</i> , <i>GSC</i> , <i>GSCSeq</i> , <i>TRC</i> , <i>Onecut</i> , <i>DRLSE</i> , <i>DRLSEIC</i> and <i>GCnoSP</i> from top to bottom row order respectively.	52
4.7	Mean and Standard Deviation of HD with error bar for the overall, central, intermediate and peripheral group of interactions from left to right order respectively in each row, for nine algorithms <i>GCSP</i> , <i>GCBS</i> , <i>GSC</i> , <i>GSCSeq</i> , <i>TRC</i> , <i>Onecut</i> , <i>DRLSE</i> , <i>DRLSEIC</i> and <i>GCnoSP</i> from top to bottom row order respectively.	53
4.8	Histogram of p values obtained from Kruskal-Wallis Tests for Dice Coefficient for nine algorithms	57
4.9	Histogram of p values obtained from Kruskal-Wallis Tests for RMSD for nine algorithms	57
4.10	Histogram of p values obtained from Kruskal-Wallis Tests for HD for nine algorithms	58
4.11	Histogram of p values (aggregation of the same data presented in Figures 4.8 to 4.10) obtained from Kruskal-Wallis Tests for Dice Coefficient, RMSD and HD for all nine algorithms	58
4.12	Coefficient of variation of Dice, RMSD and HD for four groups of seed points	62
4.13	Mean and standard deviation of Dice, RMSD and HD for all groups of interactions for all algorithms	72

4.14	Percentage of follicles having significant p values for Dice, RMSD and HD for all algorithms .	72
4.15	Percentage of CV values in the range [0-0.0025] for Dice, RMSD and HD for all algorithms .	73
5.1	A 4×4 image and its GLCM for displacement $d = (1, 0)$	75
5.2	Dice values for five algorithms that used brush stroke interactions for three ranges of different feature values	90
5.3	RMSD values for five algorithms that used brush stroke interactions for three ranges of different feature values	91
5.4	HD values for five algorithms that used brush stroke interactions for three ranges of different feature values	92
5.5	Dice values for two algorithms that used contour interactions for three ranges of different feature values	93
5.6	RMSD values for two algorithms that used contour interactions for three ranges of different feature values	94
5.7	HD values for two algorithms that used contour interactions for three ranges of different feature values	94
5.8	Dice values for two algorithms that used seed point interactions for three ranges of different feature values	95
5.9	RMSD values for two algorithms that used seed point interactions for three ranges of different feature values	96
5.10	HD values for two algorithms that used seed point interactions for three ranges of different feature values	97
6.1	An object with seed points generated by the (a) simulated interaction models (b) human operators. Blue, green and red seed points are peripheral, intermediate and central seed points respectively.	99
6.2	A histogram of accuracy values for the seed points generated by the simulated interaction model.	100
6.3	An object with brush strokes generated by the (a) simulated interaction models (b) human operators	101
6.4	An object with closed contours generated by the (a) simulated interaction models (b) human operators	102
6.5	Two images with two rectangular areas, for each, showing the qualitative feature values for these areas.	108

CHAPTER 1

INTRODUCTION

1.1 Introduction

1.1.1 Image segmentation

Image segmentation is the process of partitioning the image into multiple segments for isolating the objects of interest from the rest of the image. Figure 1.1(a) shows an image of two deer as the objects of interest and Figure 1.1(b) shows those two deer, separated from the rest of the image i.e. segmented from the background of the image.

Segmentation can be manual, automatic, or semiautomatic and interactive. Automatic segmentation doesn't require human interaction but semiautomatic and interactive segmentation (SIS) needs human interaction for providing expert knowledge to the algorithm, where this interactive phase is followed by the automatic phase.

Automatic segmentation is always preferable due to the labor-intensive and time-consuming nature of manual segmentation. Some segmentation problems, however, are still very difficult to solve with fully automatic methods especially when the number, size, and/or shape of objects of interest are arbitrary and need to be both detected and segmented or where the imaging modality and acquisition protocols result in indistinct boundaries between neighbouring objects. Difficult segmentation problems such as these occur frequently in medical image analysis, for example, identification and segmentation of ovarian follicles in ultrasonographic images [5, 114, 104], brain tumours and edema from MRI images [98, 145], and segmentation of micro-calcifications in mammography [16, 23]. The best performance, to date, for an automatic method for

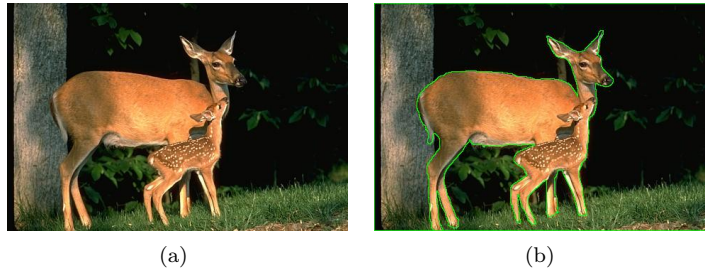


Figure 1.1: (a) An image containing objects of interest and (b) Segmented objects

segmenting ovarian follicle in ultrasound images has a detection rate of 79%, a misidentification rate of 29% with average (std. dev.) boundary deviation of $1.1 \pm 0.4\text{mm}$ [104] and a false detection rate of 22.51% [22]. Better detection and misidentification rates have been reported, but with no study of boundary accuracy measures [63]. The detection problem contributes significantly to the difficulty of these problems compared with problems where the number, shape, and position of objects of interest are known and/or very consistent, e.g. brain caduate [150], lateral ventricle segmentation [150], and prostate segmentation [119].

1.1.2 Need for semiautomatic and interactive segmentation

Limitations of automatic methods for difficult segmentation problems encourage the use of semiautomatic and interactive segmentation algorithms in which a human operator provides high-level contextual information, mostly about the approximate location of the objects of interest and their boundaries, in an interactive phase, which is followed by an automatic phase where the segmentation is performed under the constraints of the operator-provided guidance. The interactive phase improves segmentation accuracy for these types of problems at the cost of some amount of operator time, but much less than that required for manual segmentation. In order to be effectively used in practical applications, good accuracy of segmentation for any algorithm is not enough. High *reproducibility* of the algorithm is also essential. Reproducibility of a segmentation algorithm is its ability to produce consistent segmentation results across different operators under similar settings. Semiautomatic and interactive algorithms must be evaluated in terms of both reproducibility and accuracy to ensure that results are not only high quality, but also consistent within and between users. Achieving this consistency for SIS algorithms is more challenging because the resulting segmentation depends on the required parameters *and* information provided by the user through interactions. For almost all segmentation problem instances, the number of possible correct interactions is enormous and segmentation results are highly dependent on this diverse set of interactions. Thus, producing consistent segmentation is a real challenge for any semiautomatic and interactive segmentation algorithm due to the high variability in the patterns of interactions provided by different users. Hence, the evaluation of reproducibility becomes very important for especially SIS algorithms.

1.2 Evaluation of SIS algorithms: standard approach

Humans are part of the process, and must be accounted for in the evaluation. Evaluation of any SIS algorithm requires a large number of human operators who can use the segmentation application for segmenting the images repeatedly to ensure that a significantly large number of possible diverse interaction patterns are used. But to engage large numbers of human operators for an experiment is problematic due to the limitations imposed by the logistic support, funding, and, in some cases, the unavailability of enough individuals with sufficient specialized domain knowledge. That is why accuracy and/or reproducibility of SIS algorithms is typically evaluated through segmenting a number of cases by a small number of experts in the problem

domain who are well-trained in the use of the interactive segmentation system. In fact, this has been the case with numerous recent studies that analyze intra- and/or inter-observer variability [17, 19, 26, 33, 38, 44, 70, 83, 86, 103, 116, 124, 126]. Of these, only the studies of Stammberger et al. [124] and Claudia et al. [33] used more than 5 observers. The former used 7 observers, while the latter used 20 observers, but subdivided their data so that each case was only segmented by 5 different observers. In all other cases, no more than 5 observers segmented each case. Steger and Sakas’ 2012 study used only one observer and did not consider reproducibility for their proposed interactive segmentation tool [126]. Even as many as 12 or 20 examples of interactive segmentations of an object does not adequately sample the range of different possible observer interactions that would be expected to produce a correct segmentation – we call this the set of *correct interactions*. Even for the simplest kind of interaction mode where the user has to select a point known as a “seed point” or to draw a brush stroke somewhere within an object, it is not possible to robustly characterize the inherent variability in segmentation accuracy resulting from the variations in seed point placement (the underlying cause of inter- and intra-observer variability) using only a small number of sampled interactions.

Recently, some authors have turned to constructing simulated *observer models* to take into account more interactions per case, and to avoid observer variability. Moschidis et al. [92] has simulated two different patterns of user interactions in order to avoid human involvement, where the number of foreground seeds is varied from 1 to 30. For each initialization per number of seeds, 9 different perturbations of seeds produced from variable seed displacement operations were used to generate 9 segmentation outcomes and then reproducibility of segmentation was evaluated by computing pairwise Tanimoto coefficient and measuring the effect of this perturbation of the input seeds on the resulting segmentations. Only 9 sets of seeds does not represent a very diverse set of examples of possible correct interactions. Nickisch et al. [96] investigated “robot users” in the context of learning optimal parameters for interactive segmentation systems. The robot user emulates the process of a user iteratively correcting incorrectly segmented areas (false negatives and false positives) during which good parameters are learned. Robot users can be adjusted to exhibit different behaviours and the authors use a small number of different robot observers to segment each case. However, their robot users must be initialized with a fixed initial set of manually determined brush strokes which is not a very diverse sampling of correct interactions.

1.3 Potential deficiencies with accounting for humans in evaluation

From a very recent unpublished study (described in section 2.6), it has been known that humans, in aggregate, and sometimes individually, distribute interactions uniformly throughout the object of interest i.e. they place seed points everywhere in aggregate, but individually they may or may not. So, small number of samples, provided by 3 to 5 typical human users, are not enough to capture the variabilities in the interaction patterns,

aggregated across large number of users, over a long period. Thus, the main deficiency of existing evaluation methods is that an insufficiently diverse sampling of the set of correct interactions are used to draw conclusions about overall segmentation accuracy and reproducibility. Therefore, one must consider a diverse set of correct interactions in order to compare algorithms fairly and take into account the consequences of poor choices that might arise from fatigue or lapses in judgement on the part of the operator. Insufficient numbers of correct interactions not only fail to include all types of interaction patterns but also is an obstacle for applying statistical methods due to the lack of uniformity and balance in the samples. Consequently, conclusions drawn based on these random and insufficient number of samples suffer from lack of confidence due to the inappropriate method of analysis. Employing human operators for segmenting the images and evaluating those segmentations, can give a rough idea about the segmentation performance but may not be able to convey the complete information regarding the performance of the algorithm.

1.4 Problem statement

Considering the potential deficiencies with accounting for humans in evaluation, this study focuses on these potential problems of the standard method for evaluating the SIS algorithms which can be briefly stated that this method of evaluation is inaccurate due to the very small number of samples and accordingly doesn't convey the complete information about the performance of the algorithm. As a result, comparison among the performances of different SIS algorithms based on the evaluation by standard method is not fair. It also fails to reveal the underlying in-depth details regarding the performance of a SIS algorithm and consequently, is not capable of discovering the subtle differences among the performances of different SIS algorithms. Thus, to overcome these limitations, performance of these algorithms, in terms of different features and input characteristics, must be scientifically explored.

1.5 Hypothesis

We hypothesize that **large numbers of user inputs drawn from a uniform distribution of possible inputs produces more accurate and richer comparisons of algorithms vs. Standard methods.**

In order to test this hypothesis, large numbers of user interactions are generated programmatically, i.e., without using any human operator. These interactions are simulated using a component, developed for this study, denoted as simulated interaction model, which is capable of generating large numbers of simulated user interactions that are uniformly distributed over the space of the region to be segmented. For each object, all the segmentations generated from the uniformly sampled set of possible interactions are evaluated in terms of accuracy for all such interactions and also using interesting subsets of such interactions to capture the potential variabilities in the set of interactions provided by the users.

Testing this hypothesis in this way is reasonable because it will be shown that a small numbers of random interactions produce variable results. Knowing that humans, in aggregate, produce interactions, distributed

uniformly across the object, but an individual human may or may not distribute interactions evenly over the object; supports the idea that, way more than 5 users are needed; so a large number of simulated interactions drawn from a uniform distribution is a reasonable approach. It will be shown that non-uniform distribution of interactions, such as interactions only near the central area of a region to be segmented, can turn out to be either better or worse than a uniform spatial distribution of interactions depending on the algorithm.

It will be shown that, to properly characterize the mean and variance of segmentation accuracy, lots of humans are needed. Our system can do that without employing lots of humans, can correctly characterize the mean and variation of segmentation accuracy that would be produced by humans, and figure out whether we should let humans be uniform random, or encourage them to place seed points in a different way for a particular problem.

In order to test the hypothesis, a methodology will be proposed for extensive evaluation of SIS algorithms. As a vehicle for the demonstration of the proposed methodology, two case studies will be conducted where a medical image dataset will be segmented by nine segmentation applications using seven different SIS algorithms and three different types of interaction modes. Then the resulting segmentations will be evaluated following the steps of the proposed methodology.

1.6 Contributions

The overall contribution of this thesis is to introduce a methodology for evaluating SIS algorithms comprehensively which provides a systematic approach to eliminate the drawbacks of the currently existing approaches. Specific contributions of this work are:

1. The use of simulated interaction models to generate interactions programmatically is a very important contribution of this study. It ensures that interactions are generated everywhere inside the object region. This overcomes the problem of small sample size of existing methods by generating a large number of interactions which is essential to ensure the presence of interactions throughout the object region. Consequently, it replaces the role of human operator in the process of segmentation evaluation which is an important achievement, since employing human operators has some drawbacks, including human variability, human error, unavailability of expert users and expense.
2. Generating interactions throughout an image and then using these interactions for segmenting images removes the variability of human users by considering all types of inputs including potentially bad inputs. As a result, segmentations generated using all types of interaction patterns are now included in the samples which are missing in existing methods. Consequently, analysis of the segmentation performance is more accurate, comprehensive and reliable which would not be possible at all while considering only a small set of segmentations produced from a set of potentially random interactions.
3. Due to the small number of interactions used in existing methods, the presence of interaction in all

regions of the object area is not ensured and categorization of interactions is not possible. Evaluation of the segmentation performance without categorizing the interactions may be misleading and unreliable because overall performance of two (or more) algorithms obtained from considering few interactions, may look similar due to the absence of diverse interactions in the sample. But, if performance of several algorithms are compared after categorizing the interactions, segmentation results for different groups of interactions may be different for these algorithms whereas overall performance may be still very close or similar. This fine difference between the performance of several algorithms could not be discovered without categorization of interactions and categorization of interactions could not be possible without considering a large number of interactions that ensure the presence of interactions everywhere inside the object region. This also discovers the properties of a segmentation algorithm in terms of the impact of different groups of interactions. For example, some of the algorithms may be found sensitive to the location of interactions whereas some other algorithms may not be that much sensitive, or some algorithms may be affected by some of the image properties while some algorithms may not be affected by those image properties that much, which we have already noticed.

4. A case study is used to demonstrate simulated interaction models. The case study adds credence and confidence to the results using established statistical methods, which could not be achieved with current algorithm evaluation practices. In order to investigate the impact of the position of interaction on the resulting segmentation, values of Dice, RMSD and HD for all groups of interactions have been tested to inspect whether there is any difference among the means of these values for different groups, i.e., whether values of these metrics for three groups of interactions come from the same distribution or not. This hypothesis has been tested using a proper statistical method which most of the existing methods do not follow. As well, six statistical methods have been applied to analyze the results to confirm the validity of the conclusions drawn based on the outcome of the analysis. Conversely, many of the existing methods did not apply proper statistical methods to support their findings and conclusions.

Hence the proposed methodology overcomes the flaws of existing methods and provides a scientific approach for in-depth performance analysis of the segmentation algorithms. Use of simulated Interaction models and sound statistical methods have made our proposed methodology different from existing methods by rectifying the effect of random nature of human observers and strengthening the method of performance evaluation.

1.7 Outline

The rest of the thesis is organized as follows. Chapter 2 presents background on image segmentation, different types of segmentation algorithms, interaction, different types of inputs used for user interaction, evaluation of segmentation algorithms, including number of metrics used for measuring the segmentation results in terms of accuracy, reproducibility and efficiency. In Chapter 3, the simulated interaction models which we

use to generate interactions are defined and explained. Chapter 4 describes the case study to demonstrate the proposed methodology using position of interaction as the criteria for categorizing of interactions. This chapter presents the steps of the methodology along with experimental results and analysis. In Chapter 5, we give another case study using image properties as the criteria for categorization of interactions to investigate the relation between the properties of image and the corresponding segmentation results. This chapter describes several texture properties of the image dataset, used in the experiment, and presents the comparative results with those image properties. An overall discussion on the significance of the results and observations, advantages of the proposed methodology over the standard method, in addition with some other issues related to the proposed methodology, has been presented in chapter 6. Chapter 7 presents the conclusion about the study which includes the contribution of this work, several challenges in this area, some open questions and directions for future research.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Background

As this study deals with extensive evaluation of semiautomatic and interactive segmentation (SIS) algorithms, basic information about the image segmentation and different types of segmentation algorithms need to be discussed. As user interaction is an integral part of some types of segmentation algorithms, different types of inputs used in the interaction taking place between the user and the algorithm will also need to be reviewed.

2.1.1 Image Segmentation

Image segmentation is the process of partitioning the image into multiple segments for isolating the objects of interest from the rest of the image. Segmentation changes the representation of an image from a grid of pixels into groups of pixels that form regions – a higher level of abstraction – which makes it easier to interpret and analyze [101]. The general goal of segmentation is to divide an image into objects or regions that are homogeneous in some sense or have some semantic connotation that can be useful for the subsequent stages of a particular image processing or vision application. Subsequent processing steps mainly use higher-level region information instead of having to scan all pixels. Thus, segmentation is a very important preprocessing step for high level image processing tasks such as measurement, visualization, registration, reconstruction, content-based retrieval etc. [100] where the ultimate results largely depend on the quality of the primary segmentation [142]. Segmentation algorithms differ both in approach and the quality and nature of the segmentation outcome depending on the various types of needs of the applications that use segmentation. Partitioning the images roughly into similar regions is sufficient for some of the applications like multimedia indexing and retrieval [86] whereas others need the objects in the images characterized according to concrete semantic significance. Accuracy and precision are vital for some of the applications but some others consider speed and automation as the most important criteria [86].

2.2 Types of segmentation algorithms

For many years, manual tracing was the only way of segmenting images for practical purposes, which is time-consuming, labor-intensive, inaccurate and not reproducible in most cases. There has been intense

research for the last few decades on making the segmentation process automatic so as not to require human intervention. But to design a fully automatic segmentation algorithm that can segment images in all types of applications with the desired level of accuracy, efficiency and precision is not yet successful due to the differences in the image acquisition protocols and modalities, ambiguity in the image, biological variability (for medical image) and is thus an unsolved problem, in general. One way to overcome this difficulty is to incorporate high-level expert human knowledge into the segmentation algorithm. For this purpose, assistance from a human operator is essential. Depending on the presence of the human interaction in the process, image segmentation algorithms can be of two types: automatic, or semiautomatic (interactive). Semiautomatic and interactive segmentation algorithms are similar in a sense that both require human intervention but are not defined identically; the difference will be discussed in Section 2.2.2.

2.2.1 Automatic segmentation

Automatic segmentation does not require any human intervention for segmenting objects of interest in the images because expert knowledge for detecting the desired objects is incorporated into the algorithm. In the case of automatic segmentation, the user has no option to refine the results; so algorithms must be robust to handle the differences in quality of the images due to phenomena such as noise, blur and sampling artifacts caused by limitations of the acquisition modalities. For many applications, automatic segmentation is challenging and not sufficiently reliable. A large volume of applications have already been developed based on automatic segmentation, many of which are in the area of medical image analysis because successful segmentation in medical images has potential in clinical applications like diagnosis, surgical planning and radiation treatment planning. Up to now, most of the automatic medical image segmentation methods are not robust enough to be practically used for clinical applications; that is why improvement in the accuracy and robustness of automatic segmentation is a focus of current research efforts. Automatic segmentation algorithms have been used to segment different types of objects and regions mainly in ultrasound, MRI and CT images. Among the objects of interests are the brain [4, 109, 87, 37, 51, 105], brain tumours [70, 10], mouse heart [60], prostate [72], articular cartilages of knee in MRI images [47], head structures in fetal MRI images [66], esophagus [45], four-chamber heart [151], liver in CT images [111], coronary vessels in X-ray angiographic images [120], lesion segmentation in breast ultrasound images [117], mammograms [73, 131, 146, 149], closed-contour anatomical and pathological structures in confocal microscopy fluorescence images [24], RNA [147] from Fluorescent Cellular images and unstained living cells [133] in bright-field microscope images.

2.2.2 Semiautomatic and Interactive segmentation

Segmentation algorithms that include human interaction are called interactive or semiautomatic. In such algorithms there is an interactive phase where an operator provides guidance, followed by an automatic phase that accomplishes the actual task of segmentation using the information provided. Human interaction is mainly feasible for the applications where there is a need for accurate delineation of the objects and the

number of images is not overwhelming such that human intervention for further correction or improvement of the result in an iterative fashion is practical [100]. A good interactive segmentation algorithm should satisfy the following criteria: user friendly interface, minimal amount of interaction, accurate and reproducible segmentation results, fast enough to accommodate real-time visual feedback and interactive refinement and smart and effective guidance for the user [140, 126].

Both semiautomatic and interactive segmentation require human interaction to assist the segmentation process by providing expert knowledge into the segmentation algorithm, which may be information regarding the different regions of the image corresponding to the objects and background, shape of the objects, intensity of a specific region of the image, etc. Semiautomatic and interactive segmentation differs depending on the role of the user input [58]. In the case of semiautomatic segmentation, user input is mainly used as an initialization for the automatic phase of the segmentation algorithm. In interactive segmentation, user input is used in the segmentation algorithm for producing a temporary result on which the user provides feedback. So, a user can evaluate the result and if the result is not satisfactory, they can refine the result through an iterative interaction procedure. Hence, interactive segmentation is an iterative process where the user input is used repeatedly to improve the result. Interactive segmentation algorithms need to be fast enough to accept user input iteratively and provide real time feedback for each instance of user input. But for semiautomatic segmentation, moderate speed of the segmentation algorithm is good enough as the user input is accepted only once at the initial phase. Based on these criteria, all the segmentation applications developed so far that involve human interaction, may be regarded as either semiautomatic or interactive segmentation.

2.2.3 Interaction in image segmentation

A semiautomatic or interactive segmentation method consists of four main components: computational part, interactive part, the user and the user interface [100]. The computational part is the set of programs written for implementing the underlying segmentation algorithm. This is the main component capable of partitioning the objects of interest from the background of the image given the required parameters, specific to the algorithm. The interactive part acts as an intermediary between the user and the computational part. It converts the outcome produced by the computational part into visual feedback to the user and data input by the user into parameters of the algorithm. Actual communication between the user and the computer takes place via the user interface which includes physical input and output devices.

2.3 Types of input provided by user

Olabarriaga and Smeulders [100], in 2001, identified three main kinds of inputs for user-assisted segmentation: setting parameter values, direct graphic input on the image and selecting from pre-set options in a menu. During the last few years, after the publication of that survey paper, a large number of works based on

interactive segmentation have been published but the modes of input through the interaction procedure still fall into these three main categories.

2.3.1 Setting parameter values

Many of the interactive segmentation methods require a user to provide parameter values for the computational part of the algorithm. Usually these parameter values are supplied through a slider, dial, text box or similar interaction method. Upon receiving these new parameter values from the user, the result is updated and displayed on the screen so that he/she can evaluate it. Some of the examples of parameter values entered by a user include the radius of the initial sphere; width of the narrow band and epsilon [71]; weighting parameter [71, 102]; radius of the spherical seed bubble for contour initialization and values and relative weights of the parameters acting on the contour evolution [150]; and maximum size of the segmented region [121].

Setting parameter values is relatively easy to implement but the problem is that the user needs to have sufficient knowledge about the parameters, their roles in the algorithm and impact on the resulting segmentation. In this case, a user either has to be familiar with the segmentation application or needs to be properly trained. Direct parameter input also has the limitation that not all types of information can be expressed using this method. In cases where the user needs to indicate a point inside the target object, parameter setting is not realistic, rather graphic input directly on the image is more reasonable.

2.3.2 Direct graphic input on the image

Direct graphic input is a common form of user interaction for entering parameters that are points or regions of an image. A great number of segmentation applications are found in the literature that require the user to directly draw scribbles or geometric shapes on the image, indicating a seed point or seed region to mark the regions of the image for initializing the process of segmentation. Parameter values entered in this case are the spatial positions in the image in the form of points, lines, rectangles or the intensity values of the image at particular spatial positions. Most of the interactive segmentations are binary where foreground objects are isolated from background and that is why interactions are mainly designed to supply information that can be used for characterizing the foreground and background regions in the image. Several types of interactions in the form of direct graphic input on the image have been employed so far depending on the types of images and the respective applications. Interactive segmentation algorithms of this category can be further classified into two classes: boundary based and region based.

2.3.2.1 Boundary-based interaction

For this class of segmentation algorithms, users need to interact directly on the boundary of the foreground object with the help of a dynamically generated curve to obtain the desired segmentation boundary. Algorithms of this class require the user to specify the boundary approximately. There are two main ways of accomplishing this. One type, known as active contour models, requires the specification of an approximate

boundary, that is usually drawn manually by the user or in some cases, is generated from a priori knowledge. This approximate boundary evolves towards the true boundary by minimizing a cost functional based on contour deformation and external constraint forces [69, 21]. *Snake* was the first active contour model, a controlled continuity spline guided by the image forces and external constraint forces, proposed in 1987 by Kass et al. [68, 69]. Image forces push the user specified curve toward the image features like dark lines, white lines, edges, termination of line segments, etc. External constraint forces push the snake toward a desired local energy minimum. Active contours largely rely on the values of the numerous number of parameters and the quality of the initial contour [116]. *Snake* has further been improved by [129, 27, 28].

The other approach requires the user to specify sequential control points on or near the boundary, and then the complete boundary is drawn by filling in the gaps between these points using a minimal path approach [29, 9, 89, 90, 116, 88, 58, 83, 80, 3, 148, 143]. *Live wire*, also known as *intelligent scissors* was the first algorithm of this type and was introduced in 1992 by Mortensen et al. [91] and Udupa et al. [136]. In this case, users have to interactively choose an optimal boundary segment by roughly tracing along the boundary with the mouse. A minimum cost contour from the current cursor position back to the last “seed” point along the object’s boundary is determined by computing the shortest path using Dijkstra’s algorithm. This minimum cost contour is continuously updated and displayed in real-time for the user so that more seed points can be added if the computed path does not fit to the true boundary of the object. Some other applications that use this method of interaction include the work in [138] where a point for initialization near the expected object contour and several other control points are provided by the user to begin the search for the optimal path among the edges in the image represented by a graph. *Lazysnapping* also uses this type of boundary based interaction but not for generating the contour from scratch, rather to refine the boundary of an already segmented object for a more accurate matching with the true boundary [79].

In addition to these methods of generating the complete contour by filling in the gaps between the user-provided control points, there is another type of method where the user needs to draw the entire contour roughly inside the object as an initialization of the segmentation process. This initial contour then evolves to the true boundary of the object as a result of a segmentation algorithm. Some of the examples of this type of method include the initial contour inside the object of interest in [145], initial approximate boundary of the object of interest in [65], initial contour drawn by the user which is actually a seed contour used as a training example and can be used to segment other image sequences of the same type [18].

2.3.2.2 Region-based interaction

Methods of this type require users to supply hints roughly to indicate which areas of the image belong to foreground and background regions. Here users need not enclose the entire foreground object or edit the entire boundary at pixel level. Users can provide these hints by clicking or drawing a few lines or curves by dragging the mouse cursor while holding a button i.e. by scribbling on the foreground and background regions of the image. In the case of multi-label segmentation, users need to do it for each region of the image

users want to segment. The computational part of the underlying segmentation algorithm then employs these user inputs for extracting foreground object from the background.

This method provides the user with an easier and quicker way of interaction and also users need not be acutely accurate and attentive while marking the foreground and background regions in the image. Users are free to work at any scale of the image. Segmentation results are generated, at least partially, even if the user provides incomplete hints. As more hints are supplied, the foreground/background models become more accurate. Several types of interaction modes are used for region-based interactive segmentation algorithms and depending on the types of these interaction modes, these algorithms can further be classified into several groups: Point-based interaction, Scribbles-based interaction, bounding-region based interaction.

2.3.2.2.1 Point-based interaction For this type of interaction, the user needs to indicate one or more points, known as “seed points”, either inside or outside the target object by clicking the mouse on the image. For most of the cases, a seed point placed inside the target object indicates the foreground and one placed outside the target object indicates the background region in the image. Each single click adds the associated pixel into the list of foreground or background pixels used as hard constraints for the segmentation process. Examples of the methods that have used this kind of interaction are the interactive segmentation techniques proposed by Boykov and Jolly [15] where user has to mark the foreground object and background by putting seed points inside the target object and in the background area respectively, and magic wand [1], where the process of segmentation starts with a user-marked point or region to compute a region of connected pixels provided that all the selected pixels fall within some adjustable tolerance of the colour statistics derived from the specified region. The user needs to click inside the target object on the image to set the gray value interval for region growing algorithm and also needs to click on the image for specifying the leakage area generated from the region growing algorithm [84]. A number of applications for liver tumor-segmentation require the user to place one point roughly in the middle of the tumour and another point outside the tumour i.e. in the surrounding liver tissue for specifying the maximal radius [75, 122]. Several other methods that have adopted this type of interaction method which include the works in [36, 126, 71, 78, 41, 65, 152, 12, 36, 123, 25], where a point inside the target object, selected by the user, has served as the seed point and in [12] where the seed point was placed outside the object to indicate the background. Zadok et al. [11] has used this type of interaction for user feedback to indicate the regions of disagreement i.e. to mark the wrongly labelled regions in the initial segmentation through few mouse clicks.

Sadeghi et al. [112] proposed an interactive segmentation application where point-based interaction has been used for providing seed points to indicate the foreground and background regions but still is different from all other methods of the same type because a mouse has not been used as the input device, rather a novel form of user input, eye gaze tracking, has been used for the first time in segmentation. Eye gaze tracking previously had been developed primarily for disabled users who are physically or neurologically impaired and unable to use keyboards and other pointing devices. Recent improvement in accuracy and decrease in the

cost of eye gaze tracking and estimation systems have inspired the authors to introduce eye gaze as the form of input in addition to keyboard and mouse which eventually has been proved to be an alternative medium of interaction.

2.3.2.2.2 Scribbles-based interaction In this case, user has to draw scribbles on the image, mainly, for marking the foreground and the background regions in the image. Based on the reviewed methods, this is the most widely used and popular method of interaction. The user draws a few lines or curves on the image by dragging the mouse cursor while holding a button. Two buttons of the mouse can be used for marking foreground and background regions separately in two different colours. This type of high level pen-type drawing does not require very accurate inputs, rather, the algorithm is expected to work so long as the provided markups are correct.

Most of the algorithms of this type require the user to mark both the foreground and background by drawing scribbles on the image which include the works [6, 8, 140, 2, 57, 7, 52, 112, 42, 77, 125, 99, 97, 40, 106, 3, 102, 79, 96, 113, 31, 46].

Some of the algorithms require the user to draw scribbles on the image but not exactly for marking the foreground and background regions e.g. for segmenting different structural components in the image. In [30], the user needs to mark the structurally important regions of an image. In the work of Zadok et. al. [11], the user has to draw a ‘cross’ to isolate two foreground regions which were wrongly merged into a single foreground region by the algorithm. In [35], the user needs to mark the regions needed for correcting the segmentation by adding or removing those regions from the segmented region. In [123] the user is required to draw a closed contour to specify a region inside the object and in [67] the user needs to draw a skeleton of the expected shape of the target object called a ‘rack’. The user has to scribble on the image for extracting the density value of the region of interest using the intensities of the spatial positions collected from those scribbles [71]. The authors of [62] designed an algorithm where the user is required to draw the long axis of the structure to be segmented on the image to mark the contour’s location and local coordinate system. In [19] the user draws one or more spheres within the object of interest for initializing the model into the 3D view. The authors of [19] designed an algorithm where the user is required to click and drag the mouse in the regions of interest for sampling image values in order to set the free parameters of the speed function.

Yushkevich et. al. [150] have provided an user interface which has relieved the user from directly scribbling on the image, rather the user can use that interface for placing one or more spherical seeds, of user chosen radius, directly on the image.

2.3.2.2.3 Bounding region-based interaction For this interaction method, a user has to define a region of interest containing the object to be segmented by dragging roughly a circle in the form of a closed contour or a bounding box loosely around the object. The user need not specify the background as the region outside the selected area is automatically considered background. This kind of interaction is simple and user friendly but not always sufficient for complete segmentation. That is why this primary interaction

can be considered as an initialization for the segmentation process because some additional interactions for further editing of the resulting segmentation are usually needed for the complete segmentation of the foreground objects. SIOX (Simple Interactive Object Extraction) [46] is an interactive segmentation tool recently integrated into the popular imaging software GIMP as the ‘Foreground Select Tool’ where users need to draw a rough circle surrounding the foreground object to be extracted. After extracting the region of interest which contains the foreground objects, the user can optionally specify one or more foreground objects by dragging a line or curve on these objects. These selections of foreground objects are optional in a sense that sometimes the initial interaction appears to be sufficient for producing complete segmentation. After each selection of the foreground objects, the resulting segmentation is updated and user can iterate the process until the result is satisfactory. Hence, the first interaction for indicating the region containing the foreground object is mandatory but the following interactions for selecting the foreground objects are for improving the segmentation, when necessary. Another example of this type of interaction, used for initializing the segmentation process, is *Grabcut*, [110] where the user has to drag a rectangle around the desired object which indicates the outer region of the rectangle as the background and the inner region contains the foreground objects. Here also, the initial interaction is sometimes sufficient for complete segmentation but not always. In that case, further editing of the resulting segmentation is essential which can be done by dragging the foreground brush and background brush on the wrongly labelled areas of the image. Blake et al. [13] require the user to enclose the object boundary by tracing with a fat pen which defines the “trimap” with foreground, background and unclassified labels.

Some other segmentation applications have also used the interaction method where the user-drawn marker has to enclose the object of interests entirely such as the rectangle in [59, 81, 145], the marker surrounding the object of interest in [125], and freehand closed contour for adding the surrounded region to the already segmented region or for considering the surrounded region as the background in [84].

Some algorithms have used more than one approach in their applications among the above mentioned. One example of this kind is *Lazysnapping* [79] which has combined both a region-based approach and boundary-based editing at pixel level for taking the advantage of region based approach from its quick way of providing hints and the ease and efficiency of boundary based interaction for making the boundary as accurate as possible. Here users need to mark the foreground and background by drawing one or more lines with the mouse on the original colour image with separate colours for indicating foreground and background. This first step is for providing quick hints to the segmentation algorithm which works at a coarse scale. The second step is for boundary editing which works at a finer scale or on the zoomed-in image where users can edit the object boundary by using two user interface tools. One of these two user interface tools for polygon editing provides users the ability to drag the vertex for modifying the shape of the polygon. Vertices can also be added, deleted or grouped to be processed together using this tool. Using another tool, user can replace a segment of a polygon by drawing a stroke.

2.3.3 Selecting from pre-set options in a menu

For this kind of interaction strategy, the user selects an option from a predefined menu for providing information to the algorithm where choices of the menu may be the values of parameters, object properties etc. User may need to choose an option from a pull down menu, forms, group of radio buttons for selecting the option from the list of entries [100]. Some of the examples of the this kind of interaction include the buttons to choose the type of image information to be used for snake evolution [150], the command to accept or reject the resulting segmentation [85, 134], to select among the templates to use for the object of interest [62], and to choose the object property for assisting the segmentation process [61].

2.4 Evaluation of interactive segmentation methods

This thesis proposes methodology for extensive evaluation of SIS algorithms using simulated observer models. Thus, we describe the current approaches for SIS algorithm evaluation here. This section focuses on different criteria for evaluating segmentation results. Different metrics used for measuring these criteria also have been explained in detail.

Segmentation algorithms are evaluated by assessing the quality of the experimental segmentations using both qualitative and quantitative approaches. Visual inspection is the way of measuring the quality of the experimental segmentations qualitatively. Experimental segmentations are compared with the ground truth segmentations through visual inspection for assessing the degree of similarity between these segmentations. But this qualitative assessment through visual inspection has not been proved to be sufficient for conclusive evaluation of a segmentation algorithm for all the cases nor does it provide a convenient way to compare the relative performance of different algorithms. Due to the lack of a standard framework for evaluating segmentation algorithms, several factors have been adopted for assessing the effectiveness of segmentation results. Validity, reproducibility and efficiency of these results have been evaluated quantitatively by measuring the accuracy, precision and computational time of the experimental segmentations respectively [135, 19].

2.4.1 Accuracy

Ground truth of an image are the foreground pixels representing true object regions. Accuracy of segmentation is the measure of the degree to which a delineation of the object agrees with the ground truth [86]. Accuracy of an experimental segmentation is usually evaluated by computing a measure of it's similarity to the ground truth and this similarity is usually measured using two types of criteria: the amount of volume or region overlap between the segmentations, or *overlap-based similarity* and distance between the boundary points of the segmentations, or *distance-based similarity*.

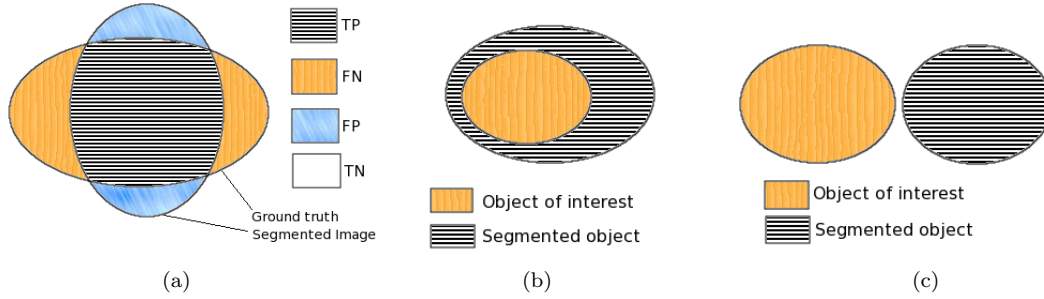


Figure 2.1: (a) Demonstration of TP, FP, TN, FN where the segmentations overlap (b) Object of interest is totally surrounded by the segmented object. (c) Object of interest and segmented object are totally disjoint

2.4.1.1 Overlap-based measures of accuracy

Several metrics based on region overlap have been used for measuring segmentation accuracy which are mainly characterized by a similarity measure between experimental and ground truth regions or volumes. While computing this similarity measure, region overlap is obtained in terms of the number of pixels or voxels contained in that region. Following are some of the commonly used region overlap-based metrics found in the literature:

Classification accuracy: This is the ratio of the number of correctly labelled pixels to the total number of pixels in an image [92, 93]. If the pixels or voxels are classified into *true positives (TP)*, *true negatives (TN)*, *false positives (FP)* and *false negatives (FN)* (Figure 2.1(a) demonstrates these quantities), then classification accuracy can be represented as:

$$CA = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \times 100\% \quad (2.1)$$

Examples of the segmentation applications that have used this metric for measuring accuracy include the methods described in [92, 93, 94]

The opposite concept of this metric also has been used in the literature [57, 81] where the ratio of the number of misclassified pixels to the total number of pixels in the original image has been used as a metric referred to as the *Misclassification rate (MR)* and can be expressed as:

$$MR = 1 - CA = \frac{|FP| + |FN|}{|TP| + |TN| + |FP| + |FN|} \quad (2.2)$$

Some authors have used the idea of *Misclassification rate* in a slightly different way, using the terms error rate [40] and segmentation error rate [13], where instead of considering the total number of pixels of the original image, only unclassified pixels to be classified have been considered. In this case, pixels, that are assigned labels from the user provided hints of foreground and background, are excluded from the total number of

pixels in the original image. Exclusion of these user-labelled pixels is more reasonable because inclusion of these pixels leads to lower misclassification rate than the actual rate.

Tanimoto coefficient: This is defined as the ratio of the area of region overlap between a segmented region and the ground truth region to the area of the union of those regions. If S and G are the sets of pixels in the segmented and ground truth regions respectively, Tanimoto coefficient is expressed as [48]

$$TC = \frac{|S \cap G|}{|S \cup G|} = \frac{|TP|}{|TP| + |FP| + |FN|} \quad (2.3)$$

Tanimoto coefficient is also known as Jaccard index (JI) [48, 86].

Here the numerator $S \cap G$ is the region that is common to both S and G . The denominator $S \cup G$ denotes the disagreement between the segmentations in addition to their agreement. This metric has a range of $[0,1]$ where 1 indicates the exact match between the the ground truth and the segmented images and 0 corresponds to complete mismatch. This measure is not biased to the segmentation that generates overly large or small number of segments and it penalizes false positives. This measure is not sensitive to small deviations in the ground-truth creation and integrates the accuracy and recall measurement into one unified function by involving both false positives and false negatives [48]. The Jaccard index has been used for evaluating segmentation accuracy by the methods in [86, 99, 34, 93, 94]. This metric has been used in [52] in percentage form by multiplying the value of Jaccard index by 100. A reformulated version of this metric is also used for measuring boundary accuracy by incorporating a tolerance to error for the pixels near the border by defining the sets of border pixels in a different way using fuzzy set theory to capture the inherent uncertainty in the edge positions. The reformulated Jaccard index has been used for measuring accuracy by the methods in [86, 99]. This Tanimoto coefficient/Jaccard index can be generalized to measure the similarity among n segmentations S_1, S_2, \dots, S_n which then takes the following form called the *Joint Tanimoto coefficient*:

$$JTC = \frac{\cap_{i=1}^n |S_i|}{\cup_{i=1}^n |S_i|} \quad (2.4)$$

JTC has the property that it is not restricted only to pair-wise similarity, rather it can compute similarity among n segmentations directly without averaging the pair-wise results, unlike some other measures.

A metric known as *volume overlap error* has been used by several methods for assessing segmentation accuracy for 3D image which is a different form of Jaccard index defined as the following expression:

$$VOE : \left(1 - \frac{|S \cap G|}{|S \cup G|}\right) \times 100\% = \frac{|FP| + |FN|}{|TP| + |FP| + |FN|} = 1 - TC \quad (2.5)$$

where S and G are the volumes of resulting segmentation and ground truth reference respectively. Equation 2.5 is equivalent to the TC/Jaccard index; it has just been inverted to form a dissimilarity measure instead of a similarity measure. In place of volume, area is considered when evaluating the segmentation of 2D image. Several interactive segmentation methods have used this index for measuring accuracy of segmentation as

described in [75, 125, 143, 122, 58].

Generalized Tanimoto coefficient: Crum et. al. [32] have derived the following general form of Tanimoto coefficient for characterizing similarity measure of region overlap among several fuzzy segmentations at each voxel for multiple labels for all pairs of segmentations using the results of fuzzy set theory:

$$TC_{PMF} = \frac{\sum_{pairs,k} \beta_k \sum_{labels,l} \alpha_l \sum_{voxels,i} \min(A_{kli}, B_{kli})}{\sum_{pairs,k} \beta_k \sum_{labels,l} \alpha_l \sum_{voxels,i} \max(A_{kli}, B_{kli})} \quad (2.6)$$

where $\min(A_{kli}, B_{kli})$ denotes the amount of fuzzy intersection across all l labels for all k pairs of segmentations at each voxel and $\max(A_{kli}, B_{kli})$ denotes the amount of fuzzy union in a similar way. α_l is a label-specific weighting factor that affects the relative contribution of each label to the overlap accumulated over all labels and β_k is a pair-specific weighting factor that affects how much each image pair contributes to the overlap accumulated over all labels and pairs of images.

Different variants of TC_{PMF} have been mentioned as the special cases of this general form and all of these special cases have been referred to collectively as *Generalized Tanimoto coefficient (GTC)*. Among the different versions of GTC, the version most suited to measuring similarity between groups of binary segmentations can be expressed as:

$$GTC = \frac{\sum_{pairs,k} \sum_{voxels,i} \min(A_{ki}, B_{ki})}{\sum_{pairs,k} \sum_{voxels,i} \max(A_{ki}, B_{ki})} \quad (2.7)$$

where (A_{ki}, B_{ki}) denotes the k -th pair of segmentations from the segmentation group, $\min(A_{ki}, B_{ki})$ is the smaller between the values of i -th voxel of that pair and $\max(A_{ki}, B_{ki})$ is the larger similarly. Previously described TC and Jaccard index are also the special cases of TC_{PMF} for a pair of binary segmentation. One thing that distinguishes the GTC from the other methods here is that it can intrinsically consider the mutual similarity of more than two segmentations. With the other methods, similarity is computed pair-wise and then averaged over the pairwise results to aggregate the overall similarity.

Dice coefficient: This metric was originally developed by Lee Raymond Dice in 1945 [39] for measuring similarity between two sets. It is strongly and explicitly related to the Jaccard index but differs in the normalizing factor in the denominator and is defined, in the context of similarity between two segmentations, as the ratio of the region overlap to the region average of two segmentations, expressed as:

$$DC = \frac{2|S \cap G|}{|S| + |G|} = \frac{2|TP|}{2|TP| + |FP| + |FN|} = \frac{2 * JI}{1 + JI} \quad (2.8)$$

where JI is the Jaccard index. While assessing the accuracy of segmentation using this metric, two segmentations will be the experimental and the ground truth segmentation and the amount of region area is measured in terms of the number of pixels or voxels contained in that region. Similar to Jaccard index, this metric also measures the actual agreement or coincidence of the ground truth foreground with the segmented image. This same metric has been mentioned as the k index [102], Dice volume overlap [35] and Dice similarity

coefficient [148, 35] by the authors in different works. This Dice coefficient can be generalized to measure the similarity among n segmentations S_1, S_2, \dots, S_n which then takes the following form called the *Joint Dice Coefficient*:

$$JDC = \frac{n(\cap_{i=1}^n |S_i|)}{\sum_{i=1}^n |S_i|} \quad (2.9)$$

JDC has the property that it is not restricted only to pair-wise similarity, rather it can compute similarity among n segmentations directly without averaging the pair-wise results, unlike some other measures.

Dice coefficient has been used by the some of the methods of interactive segmentation for evaluating segmentation accuracy e.g. in [102, 148, 35, 2, 65, 113].

True positive rate: This metric is defined as a ratio of the number of correctly labelled foreground pixels to the number of total foreground pixels in the ground truth as follows [97]:

$$TPR = \frac{|N_g \cap N_f|}{|N_g|} = \frac{|TP|}{|TP| + |FN|} \quad (2.10)$$

where N_f is the set of pixels classified as foreground pixels and N_g is the set of pixels in the foreground of the ground truth. It is the fraction of foreground pixels reported as being foreground pixels which has also been mentioned as *true positive*, *true positive fraction* or *sensitivity* in the literature [19, 78, 11]. Some applications have used this metric for measuring accuracy of segmentation such as [78, 11, 97, 19, 35].

This same metric has been mentioned as percent matching in [145] where it has been expressed in percentage form.

False positive fraction: This is the ratio of the number of pixels that are in the background region but classified as foreground pixels to the total number of foreground pixels in the ground truth [78]:

$$FP = \frac{|N_f - N_g| \cap |N_f|}{|N_g|} = \frac{|FP|}{|TP| + |FN|} \quad (2.11)$$

where N_f is the set of pixels classified as foreground pixels, N_g is the set of pixels in the foreground of the ground truth and N_b is the set of pixels in the background of the ground truth. This metric has been used for evaluating segmentation accuracy by the methods described in [78]

False negative fraction: This is defined [97] as the ratio of the number of pixels that are in the foreground region but classified as background pixels to the total number of foreground pixels in the ground truth [78] :

$$FN = \frac{|N_g - N_f| \cap |N_f|}{|N_g|} = \frac{|FN|}{|TP| + |FN|} \quad (2.12)$$

where N_f is the set of pixels classified as foreground pixels, N_g is the set of pixels in the foreground of the ground truth and N_b is the set of pixels in the background of the ground truth. This metric has been used for evaluating segmentation accuracy by the methods described in [78]

False positive rate: This metric is defined [97] as a ratio of the number of pixels that are in the background region but classified as foreground pixels to the total number of background pixels in the ground truth :

$$FPR = \frac{|N_f - N_g| \cap |N_f|}{|N_b|} = \frac{|FP|}{|FP| + |TN|} \quad (2.13)$$

where N_f is the set of pixels classified as foreground pixels, N_g is the set of pixels in the foreground of the ground truth and N_b is the set of pixels in the background of the ground truth. This metric has been used for evaluating segmentation accuracy by the methods described in [11, 97].

There is a metric called *specificity* which is actually an inverted form of FPR and is defined as a ratio of the number of correctly classified background pixels to the total number of background pixels in the ground truth i.e. the fraction of background pixels classified as being background pixels [19].

$$Specificity = \frac{|TN|}{|FP| + |TN|} = 1 - FPR \quad (2.14)$$

Several segmentation methods [19, 35] have measured segmentation accuracy using this metric. Specificity is used together with sensitivity so that both false positives and false negatives are considered. As for example, if an object of interest is totally surrounded by the segmented object (Figure 2.1(b)) sensitivity will be perfect 1.0 but actually the segmentation is not good at all which can be realized only when the specificity is also considered.

Correspondence ratio: This metric has been defined as the following expression [145]:

$$CR = \frac{|TP| - 0.5|FP|}{|TP| + |FN|} \quad (2.15)$$

CR compares the segmented foreground with ground truth foreground in terms of correspondence in size and location and balances the importance of FPs and FNs [145]. Example of the method that has used this metric for assessing segmentation accuracy include the applications described in [145].

Overlap Metric: It is defined as the following expression [78]:

$$OM = \frac{|TP|}{1 + |FP|} \quad (2.16)$$

where $|TP|$ and $|FP|$ denote the number of true positive and true negative pixels. Values of this metric are also in the range [0,1] where value 1 indicates perfect match between segmented image and ground truth image and value 0 means complete mismatch.

Some interactive segmentation methods have used this metric for measuring accuracy of segmentation such as [78]

Relative volume difference: This metric is expressed as [75]:

$$RVD = \frac{|S - G|}{|G|} \times 100\% = \frac{|FP| - |FN|}{|TP| + |FN|} \quad (2.17)$$

Here, S and G denote the same meaning as described for VOE. Smaller value of this metric indicates better

accuracy of segmentation. A negative value implies that the volume of experimental segmentation is smaller than that of the ground truth segmentation. This metric is not a good measure of segmentation accuracy because it measures the volume difference between the segmentations with respect to that of ground truth which may be very small even zero for significantly different segmentations where segmentation accuracy is actually very low. Even for a total mismatch between the completely disjoint segmentations (like Figure 2.1(c)), value of this metric may be zero which denotes the exact match theoretically, if the volumes of the segmentations are exactly same. This metric has been used for assessing the accuracy of segmentation by the methods proposed in [75, 143, 122].

A number of distance-based metrics have been used in the literature for evaluating accuracy where distance can be computed between the boundary pixels of the segmentations or from the difference between the pixel intensities of the segmentations. Distance-based metrics commonly used in the literature include *Root mean squared distance*, *Maximum symmetric surface distance*, *Hausdorff distance*, *Average symmetric surface distance*, *L2 Distance* as described below:

2.4.1.2 Distance-based measures of accuracy

Root Mean Squared Distance: It is the average minimum distance between two finite point sets. Let $G = \{g_1, g_2, \dots, g_n\}$ and $S = \{s_1, s_2, \dots, s_n\}$ be two finite point sets, then root mean squared distance is defined as [20]:

$$RMSD(S, G) = \sqrt{\frac{\sum_{s \in S} (\min_{g \in G} d^2(s, g))}{N}} \quad (2.18)$$

where $d(s, g)$ is the distance between s and g which may be Euclidean, Manhattan or any other metric of spatial distance. In case of measuring accuracy of segmentation, S and G are the segmented and the ground truth images respectively. This metric has been used for measuring segmentation accuracy in [20, 49, 122].

Mishra et. al. [88] have used *Mean squared error (MSE)* between the obtained contour and ground truth contour for measuring accuracy which is the square of the RMSD in Equation 1. This metric measures the minimum average deviation of the segmented boundary from the ground truth.

Maximum symmetric surface distance: This metric computes the maximum distance between the experimental segmentation and the ground truth segmentation for 3D images. This distance measures the maximum minimum point-wise distance between two point sets. It is defined as [75]:

$$MSD = \max\{\max\{dist(a, b)_{a \in S}\}, \max\{dist(b, a)_{b \in G}\}\} \quad (2.19)$$

where S and G are the surfaces of the resulting segmentation and ground truth reference respectively. If S and G are the set of points on the boundary of the segmented and ground truth images respectively then, this distance is a measure of the maximum mismatch between the ground truth and segmented image which is one aspect of accuracy. This metric has been used for evaluating segmentation accuracy by the proposed methods described in [75, 125, 143, 122, 58]. This metric also can be used for 2D images where, in place

of surface, contour should be considered. This metric also has been referred to as *Hausdorff Distance* [64], *maximum point to surface distance* [93, 94] and *maximum surface distance* [49].

Average symmetric surface distance: This is the general metric for computing the mean distance between the experimental segmentation and the ground truth segmentation for 3D image. It is represented as the following [75]:

$$ASD = \frac{\sum_{a \in S} \min dist(a, b) + \sum_{b \in G} \min dist(b, a)}{\sum S + \sum G} \quad (2.20)$$

where $\sum S$ and $\sum G$ are the number of pixels on the surfaces of the resulting segmentation and ground truth reference respectively. It measures the minimum average symmetric deviation between the segmented boundary and the ground truth boundary. This metric has been used for assessing segmentation accuracy by the methods in [75, 125, 143, 122]. This metric also can be used for 2D image by replacing surface with contour. Special cases of this metric are frequently used for 2D images and are referred to by different names. Passat et. al. [102] have used the metric *Mean point-to-set distance* and Yao and Chen [148] have used the metric *Mean contour distance* for measuring segmentation accuracy which are actually this same metric but with slightly different mathematical representations. This metric has also been called *Mean absolute distance*.

This metric is very similar to *RMSD* and *MSE* because basically all these metrics compute the average distance between the boundaries of the experimental segmentation and the ground truth except that, by squaring the distances in equation 2.19, larger disagreements between the surfaces/contours are penalized more.

L2 Distance: It is the distance between the experimental segmentation and the ground truth segmentation, expressed by the following equation:

$$L2 = \| I - I_G \|^2 \quad (2.21)$$

where I and I_G are the intensities of pixels of the segmented and the ground truth images respectively. Distance of each pixel in the ground truth foreground to all the pixels in the segmented foreground are summed together to get the value of this metric which is computationally very expensive. This metric is not very good for measuring segmentation accuracy because two significantly different segmentations may have smaller value for this metric than two nearly similar segmentations as the spatial distances between the pixels of the foreground objects are not considered.

For example, image A in Figure 2.2 is very different from image B and very similar to image C if considered pixel by pixel, but the value of this metric between image A and B is 215200, smaller than 217225, between image A and C. Even two totally disjoint regions may have very small value for this metric which indicates that the segmentations are nearly identical but practically that is a total mismatch. This metric has not been used that much and only one segmentation method [57] has been found that has used this metric.

100	70	0
70	100	30
30	0	30

(a) Image A

0	30	30
100	0	70
70	30	100

(b) Image B

105	75	5
75	105	35
35	5	35

(c) Image C

Figure 2.2: Image A is very different from image B and very similar to image C.

2.4.2 Reproducibility

Reproducibility, in general, refers to the consistency of the results from repeated measurements under the same conditions. In case of image segmentation, reproducibility is the consistency of segmentation results obtained from the segmentation application through interaction inputs provided by different users. It is also known as repeatability or reproducibility of segmentation. Reproducibility is evaluated without the use of the ground truth by computing the similarity of a group of experimental segmentations. For semiautomatic and interactive segmentation, reproducibility is crucial because segmentation depends on parameters *and* the information provided by the user through interactions. For almost all segmentation problem instances, the number of possible correct interactions is enormous and segmentation results are highly dependent on this diverse set of interactions. That is why producing consistent segmentation is a challenge for any interactive segmentation algorithm due to the high variability in the patterns of interactions provided by different users.

Any of the metrics which are used for evaluating accuracy can also be used for assessing reproducibility if that metric does not use ground truth explicitly and can be generalized to operate on groups of segmentations (and not just a pair). Some of the metrics for evaluating segmentation accuracy based on region overlap including Dice coefficient, Joint Dice coefficient, Tanimoto coefficient and Generalized Tanimoto coefficient, are also used for assessing precision e.g. in [19, 92, 148]. Mean contour distance has been used for measuring intra-user and inter-user reproducibility in [148].

When the similarity measure of an experimental segmentation is computed with respect to a ground truth segmentation, it represents the accuracy of that experimental segmentation but when this similarity measure is computed between two experimental segmentations, then it represents their mutual reproducibility. For N experimental segmentations, it can be generalized to compute their mutual precision by averaging the results obtained from all possible pairs of segmentations formed from N segmentations by using the metrics like Dice coefficient, Tanimoto coefficient, etc. As for example, Cates et al. [19] have measured precision by averaging the values of dice coefficients across all pairs of segmentations formed from ten (10) segmentations and Moschidis et al. [92] have similarly used Tanimoto coefficient for measuring precision from nine (09)

segmentations. But the metrics like Joint Dice coefficient (JDC), Joint Tanimoto coefficient (JTC) and Generalized Tanimoto coefficient (GTC) can be used directly to compute the precision of N segmentations without computing the precision from pairwise segmentations.

In addition with these metrics, some other metrics are used by the researchers for measuring the reproducibility of segmentation and **Coefficient of variation** is one of them. It has been used in my study and accordingly deserves the following description:

Coefficient of variation: It is a statistical metric expressed as the ratio of the standard deviation, σ , of a random variable, to its mean μ :

$$CV = \frac{\sigma}{\mu} \quad (2.22)$$

There is no strict rule to interpret the value of CV but some authors, such as Tew et al. [130], have used to an interpretation of CV suggested by Lellamo et al. [74], as $< 10\%$ (good), $10 - 25\%$ (moderate), $> 25\%$ (poor).

This metric is used for evaluating reproducibility and is computed from the values of other metrics of measuring segmentation performance. For example, Byrum et al. [17] have used volume CV for measuring reproducibility of brain and tissue segmentation volumes. There exist some other examples where volume CV has been used for measuring reproducibility of segmentation such as [124, 26, 70, 122].

2.4.2.1 Evaluation of reproducibility

In order to measure the reproducibility of a segmentation, the impact of the user interaction on the resulting segmentation needs to be investigated very rigorously. For this, different combinations of the correct interactions should be considered such that the corresponding changes in the results can be inspected. By analyzing the changes in the segmentation results generated using a large variety of possible interactions, impact of the interactions can be quantitatively measured which clearly demonstrates the actual effectiveness of the interactions. The best evaluation of reproducibility, in theory, would be to consider every possible correct interaction, but to ensure the usage of all possible correct interactions is simply not practical for most of the cases due to the extremely large number of combinations which leads to the question, how much is enough? For this reason, while evaluating an interactive segmentation algorithm, large number of human operators are required who can use the segmentation application for segmenting the images repeatedly which can ensure that most of the possible interaction patterns will be accommodated. As a result, assessment of these large number of segmentations will be statistically stable and significant which can make the assessment more reliable.

But to engage large number of human operator for an experiment is a real problem due to the limitations imposed by the logistic support, funding issue and and in some cases, the unavailability of enough individuals

with sufficient specialized domain knowledge. That is why evaluation of accuracy and/or reproducibility of interactive segmentation algorithms is typically performed by having a small number of experts in the problem domain who are well-trained in the use of the interactive segmentation system segment a number of cases. Indeed, this has been the case with numerous recent studies that analyze intra- and/or inter-observer variability [17, 19, 26, 34, 38, 44, 70, 83, 86, 103, 116, 124, 126, 148]. Of these, only the studies of Stammberger et al. [124] and Dach et al. [34] used more than 5 observers. The former used 7 observers, while the latter used 20 observers, but subdivided their data so that each case was only segmented by 5 different observers. In all other reviewed studies, no more than 5 observers segmented each case. Steger and Sakas’ 2012 study used only one observer and did not consider reproducibility for their proposed interactive segmentation tool [126]. Even as many as 12 or 20 examples of interactive segmentations of an object may not adequately sample the range of different possible observer interactions that would be expected to produce a correct segmentation – we call this the set of *correct interactions*. Even in the simplest of situations, where the interaction is selecting a seed point somewhere within an object, it is not possible to robustly characterize the inherent variability in segmentation accuracy due to variations in seed point placement (the underlying cause of inter- and intra-observer variability) using only a small number of example interactions. Recently, some authors have turned to constructing simulated *observer models* to take into account more interactions per case, and to avoid human involvement in the process of evaluation. Moschidis et. al. [92] has simulated two different patterns of user interactions in order to avoid human involvement where number of foreground and seeds is varied from 1 to 30. For each initialization per number of seeds, 9 different perturbations of seeds produced from variable seed displacement operations were used to generate 9 segmentation outcomes and then reproducibility of segmentation was evaluated by computing pairwise Tanimoto coefficient and measuring the effect of this perturbation of the input seeds on the resulting segmentations. Only 9 sets of seeds, in general, does not represent a very diverse set of examples of possible correct interactions though it depends a bit on the size of the objects involved and the image resolution. Nickisch et al [96] investigated “robot users” in the context of learning optimal parameters for interactive segmentation systems. The robot user emulates the process of a user iteratively correcting incorrectly segmented areas (false negatives and false positives) during which good parameters are learned. Robot users can be adjusted to exhibit different behaviours and the authors use a small number of different robot observers to segment each case. However, their robot users must be initialized with a fixed initial set of manually determined brush strokes which is not conducive to considering a very diverse sampling of correct interactions.

The main deficiency, therefore, of existing evaluation methods is that an insufficiently diverse sampling of the set of correct interactions for each case are used to draw conclusions about overall segmentation accuracy and reproducibility. Some subsets of correct interactions are more likely to occur than others due to the inter-user variability and very small number of samples, and, depending on the automatic phase of the segmentation algorithm, are more likely to result in a good segmentation. Therefore, one must consider a diverse set of correct interactions in order to compare algorithms fairly and take into account the consequences of poor

choices that might arise from fatigue or lapses in judgement on the part of the operator.

To overcome this deficiency, interaction models can be used where large number of user interactions are generated programmatically in such a way that can represent uniform and densely sampled set of correct interactions to capture all kinds of variability in the interaction pattern. Such an interaction model can overcome the effect of randomly placed user interactions, if the interactions are generated systematically and sampling uniformly from the set of all possible correct interactions. Statistical analysis of the metrics for evaluating reproducibility computed from the segmentations obtained from these large number of programmatically correct interactions can be a superior method of measuring reproducibility of interactive segmentation. In a recent study by Haque et al. [54], such an interaction model has been simulated by generating large number of densely sampled set of seed points programmatically for segmenting ovarian follicles in ultrasound images. These seed points have been categorized into three types of patterns based on the locations of the seed points with respect to the follicle centroid. Then the resulting segmentations have been analyzed statistically to evaluate the impact of those interaction patterns on the resulting segmentations which reveals the extent of variability in the segmentations due to the placement of the seed points inside the follicle.

2.5 Review of related works

According to my survey, no work has been found that has used the approach proposed in this thesis. That is why, small number of research papers have been discussed here which, in a broad sense, can be connected to the proposed idea. Recently, some authors have turned to constructing simulated *observer models* to take into account more interactions per case, and to avoid human involvement in the process of evaluation. Moschidis et. al. [92] has simulated two different patterns of user interactions in order to avoid human involvement where number of foreground and seeds is varied from 1 to 30. For each initialization per number of seeds, 9 different perturbations of seeds produced from variable seed displacement operations were used to generate 9 segmentation outcomes and then reproducibility of segmentation was evaluated by computing pairwise Tanimoto coefficient and measuring the effect of this perturbation of the input seeds on the resulting segmentations. Only 9 sets of seeds does not represent a very diverse set of examples of possible correct interactions.

Nickisch et al [96] investigated “robot users” in the context of learning optimal parameters for interactive segmentation systems. The robot user emulates the process of a user iteratively correcting incorrectly segmented areas (false negatives and false positives) during which good parameters are learned. Robot users can be adjusted to exhibit different behaviours and the authors use a small number of different robot observers to segment each case. However, their robot users must be initialized with a fixed initial set of manually determined brush strokes which is not conducive to considering a very diverse sampling of correct interactions.

In [127], Suinesiaputra et. al. have established a collaborative framework for building a community

resource of consensus ground truth segmentations of left ventricle (LV) in cardiac MRI images. Lack of publicly available image datasets and a common performance evaluation protocol has been considered as an open problem in LV segmentation and to solve that problem, this framework has been initiated. Ground truth for myocardium segmentations are built by applying consensus method using widely available images as input taking from both manual and fully automatic segmentations. Three manual and two fully automatic segmentation results served as the raters in the initial phase, but the ground truth can be refined iteratively by the inclusion of automated results into the consensus under certain conditions. These consensus images have been reported as more consistent than any other rater, even that with manual input. Similar resources of consensus ground truth have also been established in other domains, for example in the segmentation of the carotid arteries for stenosis evaluation [115, 53]; the detection of pulmonary nodules from lung CT images [137, 95], airway tree segmentation [82], and brain image segmentation [118].

As this last work has pointed to the lack of a common performance evaluation protocol, my proposed work can be related from that angle of view as my proposed evaluation technique using simulated observer models can ultimately lead to the development of a new segmentation evaluation protocol which also may be available online for public use.

2.6 Review of variabilities in user interactions

There are numerous studies which have investigated the variabilities in the segmentation results across different users for SIS algorithms. These studies [107, 141, 132, 108, 43] have found enough evidence to show that there are significant inter-user and intra-user variabilities in the segmentation results across different users. These studies have evaluated these variabilities and analyzed different aspects of these variabilities. Some of these studies have explained that, these variabilities in the segmentation results are the consequences of the variabilities in the interaction patterns provided by the users as the segmentation results are not supposed to be different if the interactions were same. Thus, these studies have indirectly deduced that the variabilities also exist in the interactions provided by the users.

Very recently, an user study has been conducted in my lab by a student Yunxia Li, under the supervision of Dr. Mark Eramian. This study has been just completed in the second week of August, 2016. Then Dr. Mark Eramian has given me the data and allowed me to analyze the data in order to investigate the potential variabilities in the interactions (seed point), provided by the users. In this study, 19 users segmented 75 images. I have analyzed the spatial positions of the seed points inside the foreground objects, for all 19 users, for each image separately and found that the seed points were placed across the entire region of the objects. Large variabilities were also observed in the spatial positions of the seed points, provided by each individual user. Figure 2.3 shows three images containing all the seed points, provided by 19 users, where seed points from each individual user are shown in different colours.

So, from these variabilities in the spatial positions of seed points, provided by the users, we can conclude

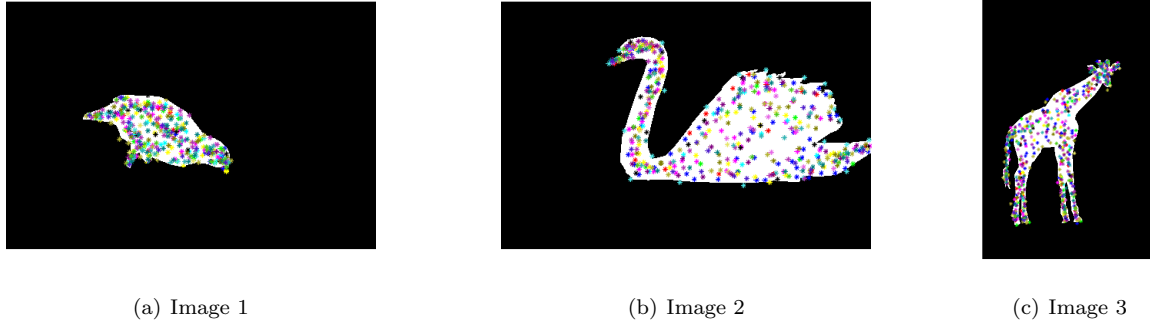


Figure 2.3: Seed points provided by 19 individual users represented in different colours

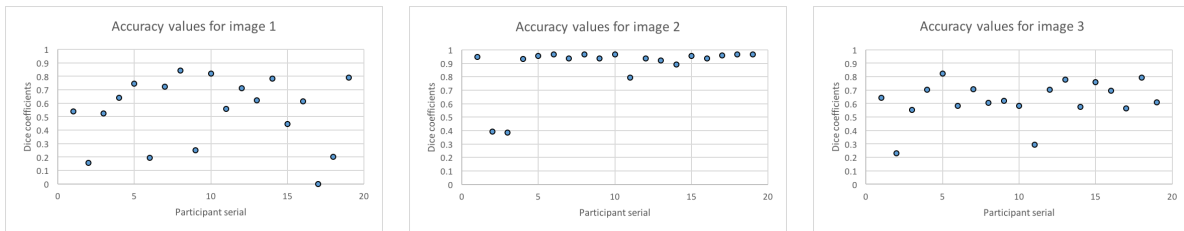


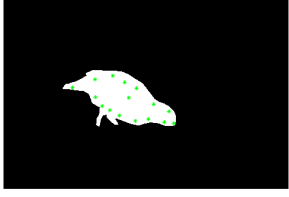

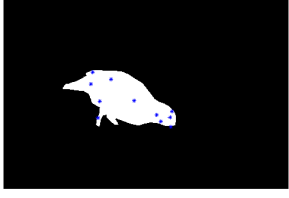








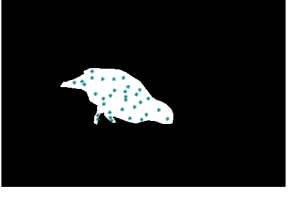


Figure 2.4: Accuracy values for three different images for the seed points provided by 19 individual users

that, as a group, the populations of seed points produced are spread uniformly around the entire area of the object. In order to investigate the variations in the resulting segmentations, accuracy values, in term of Dice coefficients, were also computed for the seed points provided by each individual user for each particular image and significant variations were observed. Figure 2.4 contains three charts showing the accuracy values for the same three images where we can observe significant variations among the accuracy values for all three images, though extent of these variations are smaller for the image 2, than that for two other images.

Variations in the segmentation results, in term of Dice coefficients, are also shown against the set of seed points in the foreground object, provided by each each individual user, with the corresponding accuracy values in the table 2.1. From this table, it can be observed that the accuracy values are different, not only for different seed point patterns, but also for similar seed point patterns, for some cases. Hence, it is apparent that the seed points that seem similarly placed can result in quite different accuracies, as well as points that are quite differently placed.

User	Interactions	Accuracy	User	Interactions	Accuracy
------	--------------	----------	------	--------------	----------

01		0.541	11		0.559
02		0.158	12		0.713
03		0.522	13		0.621
04		0.639	14		0.781
05		0.747	15		0.447
06		0.192	16		0.613
07		0.722	17		0

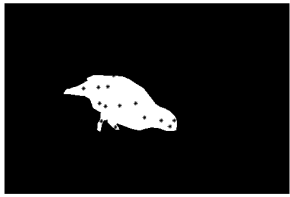
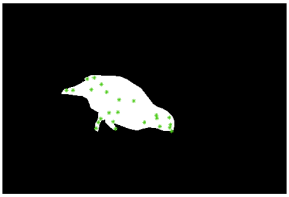

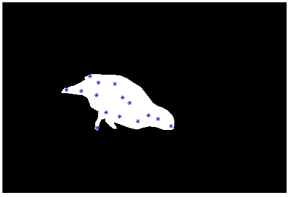

08		0.843	18		0.200
09		0.249	19		0.790
10		0.821			

Table 2.1: Interactions inside the foreground object provided by each individual user and corresponding accuracy

So, from this data, we can see that segmentation accuracy can be highly sensitive to the exact seed point placement and that similar sets of seed points can result in both very similar and very different accuracies. This evidence strongly supports the need for evaluating algorithms with input from more than five users in order to properly characterize the potential distribution (variability) of segmentation accuracy.

CHAPTER 3

SIMULATED INTERACTION MODELS

3.1 Overview of Methodology

Our proposed methodology is designed for extensive evaluation of any SIS algorithm when the objects of interest fall within a particular generic shape category denoted as *star* shape [139] defined with respect to the centre of the object based on simple geometric properties. A shape is a star shape if there exists an interior point p for which there is no straight line between p and a point on the boundary that is not completely contained within the shape. This criteria is not as restrictive as it sounds which is apparent by the objects having star shapes shown in Figure 3.1 obtained from [139]. Our methodology consists of several steps which extensively evaluate the performance of a SIS algorithm.

1. **Obtain the set of images with ground truth and choose the SIS algorithm and interaction mode:** This proposed methodology can be applied to any type of images like natural images, medical images etc., for extensive evaluation of segmentation performance for any SIS algorithm. The type of the images may also differ based on the acquisition technology, such as CT, ultrasound, MRI, etc. There should be a sufficient number of images in the dataset to justify the statistical analysis based on several statistical tests and, at the same time, not so many as to compromise practicality. The segmentation algorithm can be any algorithm which can fit into the semiautomatic way of segmentation where the algorithm starts working after being initialized with the required parameters supplied through user interactions.
2. **Generate the interactions programmatically:** We define an interaction to be the data provided by the operator to the automatic phase of a semiautomated segmentation algorithm. For example, in the case of an algorithm that accepts seed points or brush-strokes supplied by the operator specifying foreground and background areas, an interaction is a single seed point or brush stroke. A *correct*

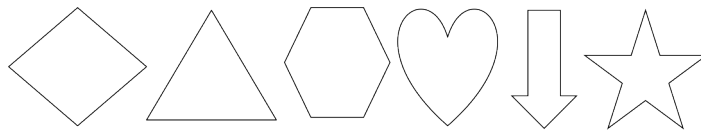


Figure 3.1: Some objects with star shapes obtained from [139]

interaction is one which provides contextual information that would be expected to produce a correct segmentation, e.g., a seed point that is inside the object to be segmented, or a brush stroke that correctly indicates areas of foreground and background. Any other mode of interaction used in the semiautomatic or interactive segmentation applications, e.g. closed contour, can also be used in this methodology.

Seed points, brush stroke and closed contour have been used as the means of supplying input parameters to the algorithm by the users, for our experiments. For semiautomatic and interactive segmentation applications, users specify the foreground and background regions using the interactions directly on the image. Users' action of supplying interactions have been simulated by generating those interactions programmatically. Details of these three kinds of interactions and the procedure for generating those interactions are described in Section 3.1.1.

3. **Categorize the interactions:** Programmatically generated interactions should be categorized based on a criterion of interest to investigate whether there is any significant difference in segmentation performance between the groups of interactions. One such criterion is position of the interaction with respect to the object of interest, determined by the distance of the interaction from a landmark point inside the object. As the interaction, particularly which one is used to mark the foreground, is placed inside the foreground object; distance of an interaction can be measured from a landmark point inside the object such as the object's centroid. Another criteria can be the characteristics of the image in the neighbourhood of the interaction. For example, interaction could be categorized by properties like texture, intensity values, noise, etc.

4. **Analyze the results using statistical methods:** Several metrics are computed to evaluate the quality of the resulting segmentation. Then these segmentation results should be analyzed using statistical methods in order to make the analysis robust, reliable and scientific. The main objective of the analysis is to look for differences in performance among the interaction groups defined by the criteria chosen in step 3. The presence of these differences determines the impact of the variation in the interaction patterns on the resulting segmentations, providing an in-depth analysis of algorithm performance.

Dice coefficient, root mean squared distance (RMSD) and Hausdorff Distance (HD) are used for measuring segmentation accuracy. Then values of these metrics are analyzed by applying several statistical methods, which include mean, standard deviation, coefficient of variation (CV), regression, Wilcoxon Rank-Sum test, Kruskal-Wallis hypothesis test, etc. Results of this analysis are represented by charts and tables. Statistical analysis of the performance metrics between categories of interactions is discussed in Section 4.3.

So, from this outline of the proposed methodology, it can be noted that the inputs of the methodology are the image dataset with ground truth and the choice of the SIS algorithm. Then, the images of the dataset are segmented by the SIS algorithm and these segmentations are evaluated in terms of

accuracy and analyzed statistically. So, it is clear that the methodology doesn't depend on the type of the images, as long as the images are segmented by the SIS algorithm. Hence, this methodology should work for any kind of image. Output of this methodology is the extensive evaluation of the performance of the chosen SIS algorithm(s) in the forms of tables and charts.

3.1.1 Programmatic Generation of Interactions

The second step of the proposed methodology is to generate the user interactions, but not by employing a human user, but rather by generating them programmatically. For each object to be analyzed, a set of correct interactions needs to be generated. For this study, Seed point, brush stroke and closed contour are used as the means of supplying input parameters to the algorithm by the users. Details of these three kinds of interactions and the procedure for generating these interactions are described in the next few subsections.

3.1.1.1 Seed point Interaction

For seed point interaction, the user has to click once inside each object of interest to indicate the foreground object, and the corresponding clicked point is known as the 'seed point'. For each object to be analyzed, a set of seed points is generated. As the seed points are the pixels inside the objects, theoretically all the pixels within an object can be treated as seed points. Since using every pixel within an object as a seed point would result in an excessive amount of data, seed point locations are sampled at regular intervals on a grid with a variable spacing depending on object size. Seed point locations are sampled more sparsely for larger objects to maintain computational feasibility. Grid spacing is determined using the following procedure:

1. Determine the approximate number of seed points N , for each object, to be used for segmenting the object. The size of the object, in terms of total number of pixels, is used to determine this number. The relationship between the object size and number of seed points need not be linear in order to limit the number within a feasible range. For this purpose, a function can be developed which computes the number of seed points for each object from the number of total pixels in that object. The function can be determined empirically by testing different values for the feasible range. A feasible range of seed points is important because it ensures the generation of a minimum number of seed points that is enough to get sufficient number of segmentations needed for the statistical analysis and on the other hand, it also ensures that number of seed points is not so large as to be computationally infeasible.
2. Determine the grid spacing as $\lceil \sqrt{A/N} \rceil$ where A is the area (in pixels) of the object (determined from the ground truth). Total number of seed points obtained, using this grid spacing as the regular interval, is not likely to be exactly N , but should be very close to N .

Figure 3.2 shows an object with the grids inside the object region. The distance between these gridlines has been computed using this procedure. Each intersection point of the gridlines is the position of a seed point.

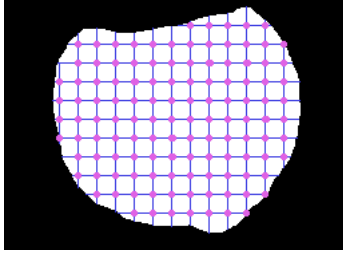


Figure 3.2: An object with grids inside the region where each intersection point of the grids represents the position of a seed point.

3.1.1.2 Brush Stroke Interaction

For brush stroke interactions, our interaction model simulates the user’s interaction to draw one or more scribbles for indicating the foreground and background regions. Accordingly, brush strokes have been generated for both foreground and background regions. For both cases, two types of strokes are generated: straight and curved. For each object to be analyzed, a set of brush strokes is generated. Straight brush strokes are the segments of straight lines but thickness of the strokes can vary between 3 and 5 pixels, which is the typical range of brush stroke width used in almost all published works.

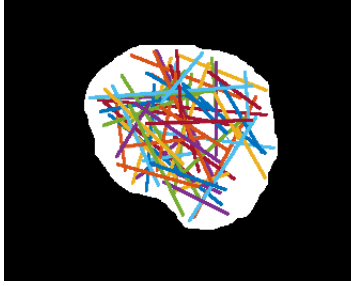
Straight brush strokes are generated randomly inside the object region. The size of these strokes are varied within a particular range depending on the area of the object. The length of this stroke L satisfies Equation 3.1:

$$strokeUnit \times lim1 \leq L \leq strokeUnit \times lim2. \quad (3.1)$$

Here $lim1$ and $lim2$ are the lower and upper limits, respectively, and $strokeUnit$ is the unit of length of a stroke in pixels. The following heuristic formula using the *diameter* of the object determines the value of $strokeUnit$ where the relation between $strokeUnit$ and *diameter* of the object is not linear:

$$strokeUnit = \begin{cases} l_1, & diameter \leq d_1 \\ l_2, & d_1 < diameter \leq d_2 \\ \vdots & \\ l_n, & diameter > d_n \end{cases} \quad (3.2)$$

Here d ’s are pre-selected thresholds on the diameter, and the l ’s are the corresponding stroke lengths. The number of these strokes also depend on the area of the object. Another heuristic formula using the total number of pixels $numPixels$ within the object region determines the number of these strokes $numStrokes$:



(a)

Figure 3.3: (a) Straight brush strokes within object region

$$\mathbf{numStrokes} = \begin{cases} \text{round}(\text{numPixels}/f_1), & \text{numPixels} < t_1 \\ \text{round}(\text{numPixels}/f_2), & t_1 \leq \text{numPixels} < t_2 \\ \vdots \\ \text{round}(\text{numPixels}/f_n), & \text{numPixels} \geq t_n \end{cases} \quad (3.3)$$

Here t 's are pre-selected thresholds on the number of pixels within the inner most region of the object and f 's are the corresponding parameters to compute the number of strokes. Figure 3.3(d) shows straight brush strokes generated inside the object. Here only some of the sampled straight brush strokes from the set of all straight brush strokes are shown for better visibility.

Curved brush strokes are generated as segments of curves whose widths are also varied between 3 to 5 pixels, as these are the typical widths observed in existing implementations of SIS algorithms. These strokes are segments of iso-contours of the distance transform of the object region and roughly parallel to the object edge. The number of iso-contours in each object is determined by the average diameter of the corresponding object and computed by Equation 3.4

$$\mathit{numContour} = \text{round}(\mathit{maxDist}/\mathit{div}) \quad (3.4)$$

where $\mathit{maxDist}$ is the maximum distance in the distance transform of the complemented ground truth of each object and div is a number which depends on the feasible number of contour and is determined from the width of the contour. For example, for $\mathit{width} = 3$, div can be, $\mathit{width} + 3 = 6$ to allow some gap between the contours. The number of strokes $\mathit{nStrokes}$ obtained from each iso-contour is determined using Equation 3.5

$$\mathit{nStrokes} = \text{round}(\mathit{numPixIsoCntr}/\mathit{strokeUnit}) \quad (3.5)$$

where $\mathit{numPixIsoCntr}$ is the number of pixels on an iso-contour and $\mathit{strokeUnit}$ is the unit of stroke length in pixels, determined from the diameter of the object using Equation 3.2.

Length of a stroke is equal to the *strokeUnit* for very small iso-contours, which produces only one stroke; for all other cases, length of a stroke= $strokeUnit - strokeGap$, where *strokeGap* is the gap between two successive strokes which is set at 8 pixels. The gap between iso-contours is 3 pixels.

Brush strokes for the background region are also generated as straight and curved strokes. For each image, a fixed set of background strokes are used. As one of the objectives of this study is to investigate the impact of the variations in the interaction patterns on the segmentation results, only foreground strokes are variable; so that the corresponding difference in the result can be observed as the effect of the change in the interaction. Each image is segmented once for each of the foreground strokes and thus the corresponding changes in the segmentation results are observable because the changes in segmentation results are only due to the changes in the foreground strokes. If background strokes were also variable, individual effect of the changes in foreground and background strokes could not be distinguished. Moreover, for a particular foreground stroke, a huge number of background strokes would be needed to generate the segmentations. Thus, the number of segmentations generated for all foreground strokes combined with all sets of background strokes opens up a gigantic combinatorial explosion. For these reasons, using a fixed set of background strokes for each image is justified for this study, but studying the effect of varying background strokes at the same time as varying foreground strokes is a huge undertaking that must be left to future work.

3.1.1.3 Closed Contour

This type of interaction mode is used to mark the foreground region. A user draws a closed contour inside the foreground object. Two types of closed contours are generated. The first is the closed contours whose shape are elliptical in general and do not depend on the shape of the object. The second is the iso-contours whose shape are roughly parallel to the edge of the object. Live users are free to draw a closed contour anywhere inside the foreground object; the simulated interaction model has a similar ability. In order to simulate the actions of real users, closed contours are generated at different sizes where the range of size varies and is directly proportional to the distance between the centroid and the nearest point on the boundary of the foreground object. For generating large numbers of regularly sampled contours, the following procedure is carried out:

- First, several curves inside the foreground region are determined. These curves follow the edge of the foreground object and number of these curves is determined from the distance between the centroid and the nearest point on the boundary of the foreground object. These curves serve as the base lines for generating the closed contours.
- A set of points on each of these base lines are sampled at a regular interval and these points are used as the centroid of the ellipses to be generated. Figure 3.4 shows the image of a foreground object containing few base lines with one base line having two ellipses generated considering the centroids on that base line.

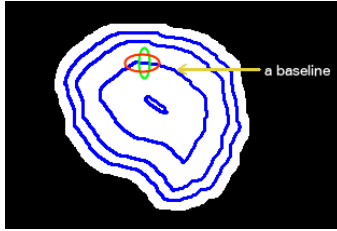


Figure 3.4: Example of a baseline with two ellipses having the centroids on it

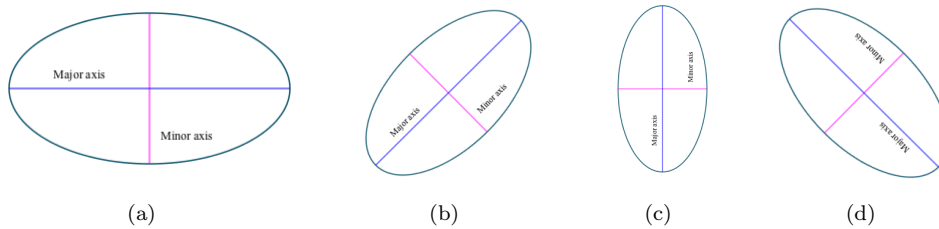


Figure 3.5: Ellipses of different orientations obtained by rotating the axes

- Lengths for the major and minor axes of the ellipses to be generated are computed which starts from the base line closest to the object boundary (let us denote this base line as the first base line). The distance between this first base line and closest point on the boundary of the object is used to compute the length of the major axis. Thus, orientation of the major axis is tangent to the base line. The length of the minor axis is then computed as $minorLength = majorLength * Ecc$ where Ecc is the eccentricity of the foreground object. Lengths of major axis for all other base lines are computed by successively increasing the length at a fixed rate and lengths of the minor axis are computed using the same equation. Varying lengths of these axes are used to generate the closed contours of different sizes.
- Axes of the ellipse are also rotated with an equal increment for the angle to generate ellipses of varying angles to enhance the degree of resemblance of the contours to the real ones drawn by humans. The increment for the angle of rotation is also determined empirically considering the total number of contours to be generated within feasible range, because rotating the axes produces a number of contours of the same size but with different orientations; for example, if the increment for the angle of rotation is 45 degree, the number of contours will be $= 180/45 = 4$. Figure 3.5 shows four ellipses of different orientations obtained from rotating the axes.
- Another type of closed contour which are actually iso-contours are also generated to be used as the initial contours to mark the foreground region. These iso-contours are roughly parallel to the edge or boundary of the foreground object. The number of these contours depends on the size of the foreground object and is determined from the distance between the centroid and the nearest point on the boundary

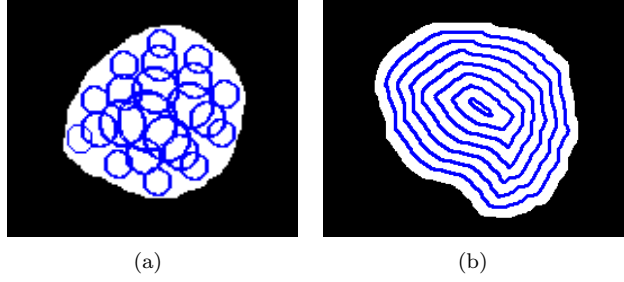


Figure 3.6: (a) An object with few generated closed contours inside the object region (b) An object with some of the generated iso contours inside the object region .

of the foreground object using Equation 3.6.

$$numIsoContour = round(maxDist/div) \quad (3.6)$$

where $maxDist$ is the maximum distance in the distance transform of the complemented ground truth of each object and div is a number which depends on the feasible number of iso-contours that fit into the object area. The value of div is selected such that it ensures a minimum gap between two successive iso-contours. This value can be varied empirically to control the number of contours to be generated. Sizes of these contours are varied between the closest to the farthest from the boundary of the foreground object.

Figure 3.6(a) shows an object with some of the closed contours of various sizes from the large set of generated closed contours inside the object region and Figure 3.6(b) shows some of the generated iso-contours inside the region of an object, both generated by the simulated interaction model. For better visibility, only a few of the generated closed contours and iso-contours are shown; otherwise, each individual contour could not be identified due to the high density of the contours.

3.1.2 Categorization of interactions

The proposed methodology requires that interactions should be categorized according to the position of the interaction with respect to the object, where this position is numerically represented by the distance of the interaction from the centroid of the object. For the seed point mode of interaction, seed points are categorized into several groups considering the distance of the seed point from the centre of the foreground object. To determine the category of a seed point, out of n categories in total, a binary ground truth image (foreground pixels representing true object regions) is negated and then the distance transform of that image is computed. From this transform, distance a of the seed point from the nearest boundary point is determined and distance from the seed point to the centroid b is calculated using the Euclidean distance metric. The seed point category \mathbf{c} is then determined by a double threshold of the quantity $\frac{a}{a+b}$:

$$\mathbf{c} = \begin{cases} \text{category 1,} & \frac{a}{a+b} \leq 1/n \\ \text{category 2,} & 1/n < \frac{a}{a+b} \leq 2/n \\ \vdots & \\ \text{category n,} & n - 1/n < \frac{a}{a+b} \end{cases} \quad (3.7)$$

For the brush stroke mode of interaction, two types of brush strokes, straight and curved, are generated, where straight brush strokes are randomly generated inside the object region. Categorization of straight brush strokes is different from that of seed points because the category of a seed point is determined by the position of the interaction inside the object region with respect to the centroid of the object where the interaction is fully contained within the same categorical area. But this technique of categorization fails due to the randomly generated position of the straight brush stroke inside the object region where all the pixels of a straight brush stroke may not belong to the same categorical area. In that case, the category of each pixel of a straight brush stroke is first determined applying the same technique which is used for seed points and the total pixel counts for each category is computed. Then the category of a straight brush stroke is determined by majority vote of the categories of the individual pixels that comprise the straight brush stroke.

Curved brush strokes are categorized into several groups depending on the size of the foreground object. The number of groups for categorization may vary between 1 and 3 and depends on the size of the object. As curved brush strokes are the segments of iso-contours, the category of a brush stroke is determined from that of the corresponding iso-contour i.e., categories of all the brush strokes originating from a single iso-contour are the same. The category of an iso-contour is first determined and then that category is assigned to all the brush strokes originating from that iso-contour. The category of an iso-contour is not directly computed, rather the number of iso-contours for each category is computed, where *category1* is for the innermost region and other categories are assigned in ascending order for the regions located gradually toward the object boundary. The total number of iso-contours is used to determine the number of iso-contours for each

category using Algorithm 1:

```

input : Number of iso-contour  $nContour$ 
output: Categories for each iso-contour
initialization;
if  $nContour = 1$  then
     $category1 \leftarrow 1$ ;
     $category2 \leftarrow 0$  ;
     $category3 \leftarrow 0$  ;
else if  $nContour = 2$  then
     $category1 \leftarrow 1$ ;
     $category2 \leftarrow 0$  ;
     $category3 \leftarrow 1$  ;
else
     $numForAllCategories \leftarrow \text{floor}(nContour/3)$ ;
     $rem \leftarrow \text{mod}(nLevel, 3)$ ;
     $category1 \leftarrow numForAllCategories$ ;
     $category2 \leftarrow numForAllCategories$ ;
     $category3 \leftarrow numForAllCategories$ ;
    if  $rem = 1$  then
         $category2 \leftarrow category2 + 1$  ;
    else if  $rem = 2$  then
         $category1 \leftarrow category1 + 1$  ;
         $category2 \leftarrow category2 + 1$  ;
    end
end

```

Algorithm 1: Determining the number of iso-contour for all categories

Closed contour interactions are also categorized into groups depending on the size of the foreground object. Categorization of closed contour is exactly same as that of straight brush strokes, where the category of each pixel of a closed contour is first determined by applying a similar technique to that used for seed point to compute the total pixel count for each category. Then the category of a closed contour is determined by majority vote of the categories of the individual pixels that comprise the contour.

Another type of closed contour, which are actually iso-contours, are equidistant from the edge of the object and accordingly, almost all the pixels of an iso-contour belong to the same categorical area. For determining the category of an iso-contour, distances of all the pixels of an iso-contour from the boundary of the foreground object are computed. Although these distances are supposed to be almost the same for all the pixels, still the average of these distances are computed to avoid any unforeseen instances such as if the contour is unusually wide, where some pixels of the contour may fall within another category. Then, this

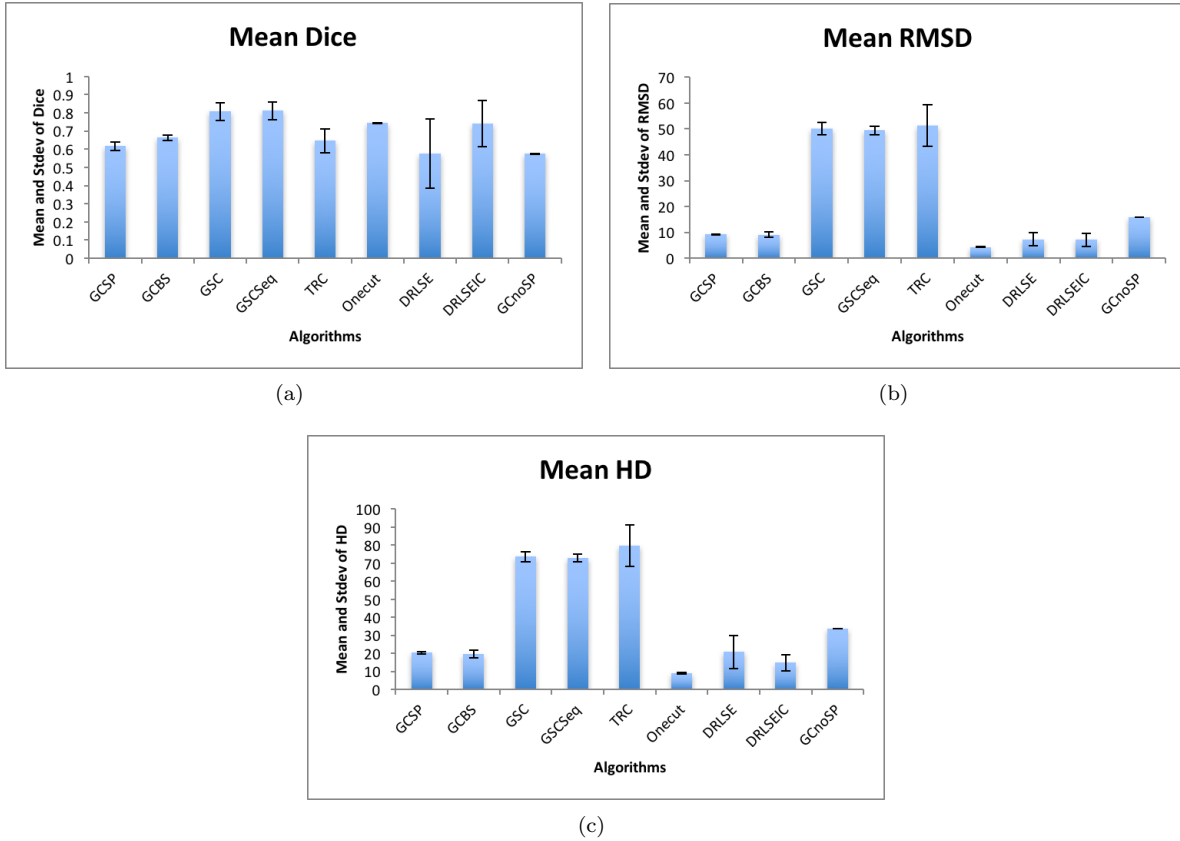


Figure 3.7: Mean and standard deviation (sample) of Dice, RMSD and HD for all nine algorithms

mean distance is used to determine the category of the iso-contour in the same way as it is determined for seed point.

3.2 Evaluation methodologies

3.2.1 Traditional Methods

Existing approaches for evaluation of segmentation use small number of interactions provided by 3 – 5 human operators for each object to be segmented. This small number of interactions cannot be categorized into different groups and consequently, analysis of the segmentation results based on the categorization of interactions is not possible and hence can only produce the overall measurement of segmentation accuracy. For example, existing approaches for evaluation of segmentation would have produced the segmentation accuracy in terms of Dice, RMSD and HD as presented in the charts in Figure 3.7 for nine algorithms. These nine algorithms have been implemented to demonstrate this proposed methodology through two case studies described in Chapter 4. More specific information about these algorithms are mentioned in Table 4.1

3.2.2 Mean segmentation accuracy within categories

Accuracy is computed for all segmentations using three different metrics: Dice coefficient, RMSD and HD. Mean and standard deviation of the accuracies for the segmentations resulting from using the interactions located within each object are computed. Then these values are categorized according to the groups of interactions based on the positions of the interactions within the object region or texture properties of the interactions. These mean values of accuracy in term of Dice, RMSD and HD are sorted according to the size of the objects and represented in charts for all the interactions and for all groups of interactions separately. Trends of these accuracy measures are determined and presented by linear regression lines known as trend lines. These trend lines indicate the relation between the independent variable object size and the segmentation results in term of accuracy. In order to determine the significance of these regressions, p values of the regressions are computed.

Segmentation results are categorized according to the groups of interactions in order to examine whether these results are same for all groups of interactions. This is accomplished by testing whether the accuracy values for all three groups of interactions come from the same distribution. This is a hypothesis test where the null hypothesis is that the metric values for three groups of interactions come from the same distribution. Before testing this hypothesis, normality of the accuracy values should be tested. If the values are normally distributed, this hypothesis can be tested by applying the Student's T-test or classic one way ANOVA. But if the values are not normally distributed, a non-parametric alternative, *Kruskal-Wallis Test* should be applied which does not require the data to be normally distributed.

3.2.3 Coefficient of variation for lightweight evaluation of reproducibility potential

Measuring the reproducibility of SIS algorithms is essential but expensive, whether done with humans (money and time) or simulated Interaction models (computationally). We can quickly assess whether an algorithm is likely to be reproducible before incurring such expense by computing Coefficient of variation (CV) of accuracy measures. CV is the ratio of the standard deviation of a population to the populations sample mean. A high CV implies high variability in the accuracy, and that the reproducibility is likely to be poor. However, the converse that a low CV implies high reproducibility is not always true, since consistency of an accuracy measure does not imply that any errors in accuracy are the same. Thus, if we observe a high CV of accuracy, we may conclude that reproducibility is likely to be low, and not worth investigating in detail. CV of Dice coefficient, RMSD and HD are computed from the segmentation results for each group of interactions. The histograms of these CV values show their distribution and enables comparison of the algorithms with respect to their potential reproducibility.

Computing CV can determine whether a SIS algorithm is likely to have poor or high reproducibility. Reproducibility measure is likely to be different for different SIS algorithms and different interaction modes.

Even for a particular algorithm, reproducibility may not be similar for different groups on interactions. This possibility encourages us to inspect whether the CV values for three different groups of interactions are different or not, from the statistical point of view. This investigation should include all the tests between all possible pairs of interaction groups for each algorithm. For this, CV values of Dice, RMSD and HD for each group of interaction are compared with that of two other groups of interactions to statistically test whether the values for each pair of the group of interactions are same in term of population mean. In order to assess whether the CV values come from the same population or not, a non-parametric statistical hypothesis test named *Wilcoxon rank-sum test (WRS)* also known as *Mann-Whitney U test* is applied if the CV values are not normally distributed.

CHAPTER 4

SEGMENTATION OF FOLLICLES IN ULTRASOUND IMAGES: A CASE STUDY OF THE PROPOSED METHODOLOGY USING POSI- TION OF INTERACTION FOR CATEGORIZATION

The proposed methodology outlines the methods to evaluate the performance of any SIS algorithm extensively. In order to demonstrate the proposed methodology, a total of seven SIS algorithms combined with three types of interaction modes, making a total of nine segmentation algorithms, were implemented and the segmentation results were analyzed using the proposed methodology.

Table 4.1 gives a list of the algorithms, their abbreviated names used in the remainder of the text, the interaction mode used for each algorithm, and a reference to the corresponding papers for further details.

For these experiments, a dataset of 32 ultrasound *in vivo* human ovarian images obtained from a previous study [5] were used which were acquired using high-resolution Ultramark 9 and ATL HDI 5000 ultrasound machines with 5–9 MHz multifrequency convex array transducers (Advanced Technologies Laboratories, Bothell, WA). The size of each image is 640×480 pixels and the maximum number of follicles in an image was fourteen. This dataset consists of 81 follicles with a diameter larger than 2.5mm. This is a significant size threshold because even human observers have difficulty correctly identifying follicles of this size or smaller. Manually delineated ground truth segmentations of these follicles were provided by a single, highly experienced human operator.

Table 4.1: List of the algorithms and interaction modes used in our experiments

Name of the algorithm	Acronym	Interaction mode
Graphcut with Star shape Prior [139]	<i>GCSP</i>	Seed point
Graphcut with Star shape Prior [139]	<i>GCBS</i>	Brush stroke
Geodesic Star Convexity [52]	<i>GSC</i>	Brush stroke
Sequential Geodesic Star Convexity [52]	<i>GSCSeq</i>	Brush stroke
Trust region convexity [50]	<i>TRC</i>	Brush stroke
Onecut [128]	<i>Onecut</i>	Brush stroke
Distance Regularized Level Set Evolution [76]	<i>DRLSE</i>	Closed contour
Distance Regularized Level Set Evolution [76]	<i>DRLSEIC</i>	Closed iso-contour
Graphcut without Star shape Prior [14]	<i>GCnoSP</i>	Seed point

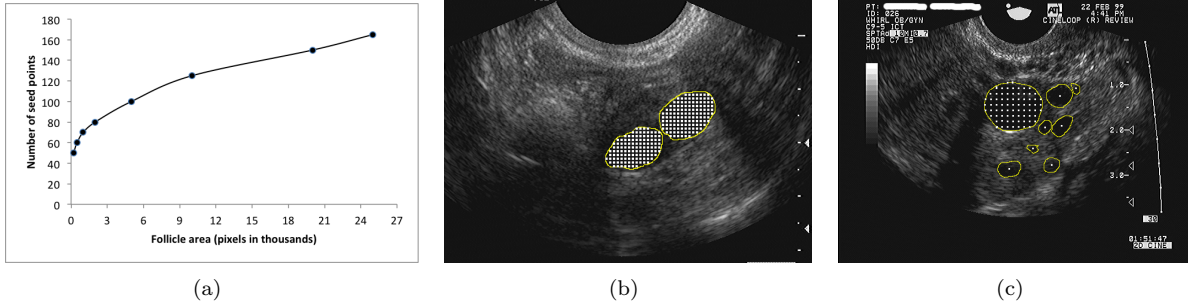


Figure 4.1: (a) Piecewise cubic polynomial function used for determining the number of seed points for each follicle. (b) Programmatically generated seed points inside each follicle in an ultrasound image. (c) The large follicle is segmented once with each of the seed points shown while the remaining follicles' seed points (at the centroids of their regions) are held constant. This process is repeated for each other follicle in the image.

4.1 Interaction models for generating interactions within follicles

For these segmentation applications, sets of correct interactions for three types of interaction modes were generated. For each follicle to be analyzed, a set of seed points was generated for several segmentation algorithms. Seed points were sampled on a grid with a spacing of between 2 and 14 pixels depending on follicle size. The number of seed points to be used for the follicle was determined as a function of its area using the cubic spline interpolation function shown in Figure 4.1(a). Interpolation points were selected empirically based on the data set. This resulted in sets of seed points consisting of between 50 and 375 seed points for each follicle.

Each follicle was segmented using each seed point from its set of correct interactions exactly once while the seed points for any other objects in the image were held constant (Figure 4.1(c)). These constant seed points were the centroids of the follicle region, determined from the ground truth. This method of generating and using seed points for segmenting objects in images was published in 2013 [54].

For five algorithms, a set of brush stroke interactions were generated for each follicle to be analyzed. Two types of brush strokes, straight and curved, were generated for both foreground (follicle regions) and background regions. Straight brush strokes were generated randomly inside the follicle region. Sizes of the straight brush strokes were determined using Equation 3.1 for $lim1 = 0.6$ and $lim2 = 2.0$. The value of $strokeUnit$ was computed using Equation 3.2 for $n = 5$ and the specific thresholds shown in Equation 4.1:

$$\mathbf{strokeUnit} = \begin{cases} 20, & diameter \leq 32 \\ 35, & diameter \leq 56 \\ 50, & diameter \leq 96 \\ 65, & diameter \leq 136 \\ 85, & diameter > 136 \end{cases} \quad (4.1)$$

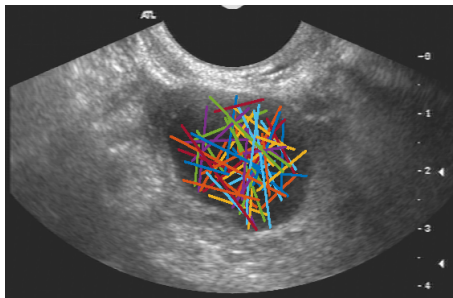


Figure 4.2: Straight brush strokes inside the follicle region

These thresholds have been determined empirically based on the diameters of the follicles considering the fact that the maximum length of a stroke within a follicle should be roughly proportionate to the follicle size, which of course, does not prevent generation of shorter strokes. The number of strokes for a follicle were determined using Equation 3.3 for $n=3$ and replacing the parameters with the values in Equation 4.2:

$$\mathbf{numStrokes} = \begin{cases} \text{round}(\text{numPixels}/140), & \text{numPixels} < 1000 \\ \text{round}(\text{numPixels}/150), & 1000 \leq \text{numPixels} < 2000 \\ \text{round}(\text{numPixels}/200), & \text{numPixels} \geq 2000 \end{cases} \quad (4.2)$$

Expected feasible numbers of strokes within follicle regions were considered for determining these values. Figure 4.2 shows a few straight brush strokes randomly generated within a follicle region. Only a subset of the generated strokes are shown to allow for better visibility; otherwise, strokes could not be individually identified due to the high density of strokes.

Curved strokes were generated as the segments of iso-contours of the distance transform of the follicle region. The number of iso-contours in each follicle, which varied between 1 to 14, was determined using Equation 3.4 for $div=6$. The number of strokes in each iso-contour varied between 1 and 6 and was computed using Equation 3.5. The length of each stroke varied between 27 to 77 pixels and were determined using Equation 3.3. The gap between two successive iso-contours is 3 pixels and between strokes on the same iso-contour is 8 pixels.

Brush strokes for the background region were also generated as straight and curved strokes but the number of these strokes was not dependent on the background area, rather, it was determined empirically. Various numbers of background strokes, starting from 1 as the number of strokes, were tested to determine the right number of strokes that are good enough for producing the best possible segmentation with all other settings being unchanged and seven strokes were found sufficient for most of the images. An exhaustive testing approach was used for this empirical study. A fixed set of manually generated background strokes were used for each image. Figure 4.3 shows an image with the generated curved brush strokes inside the ovarian follicle in the ultrasound image.

The third type of interaction generated for two algorithms was closed contour. Two types of closed

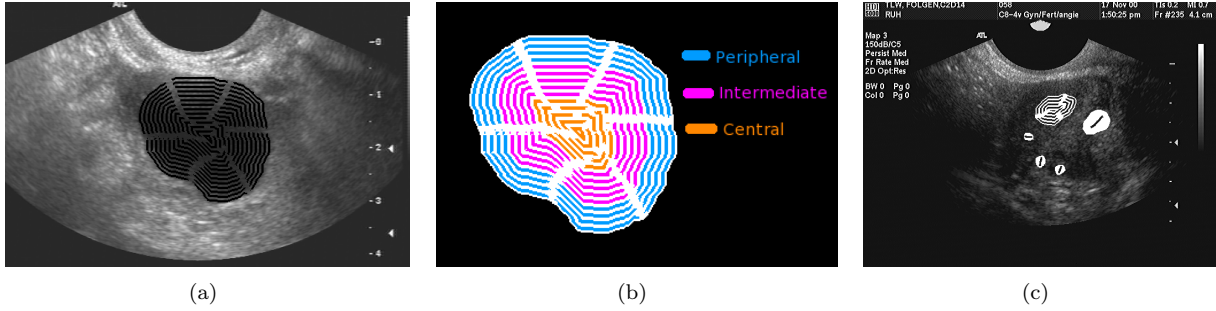


Figure 4.3: (a) Curved brush strokes inside the follicle (b) Curved brush strokes of three categories shown in different colours (c) Each follicle is segmented using each brush stroke from its set of all brush strokes exactly once while the brush strokes for any other follicle in the image are held constant

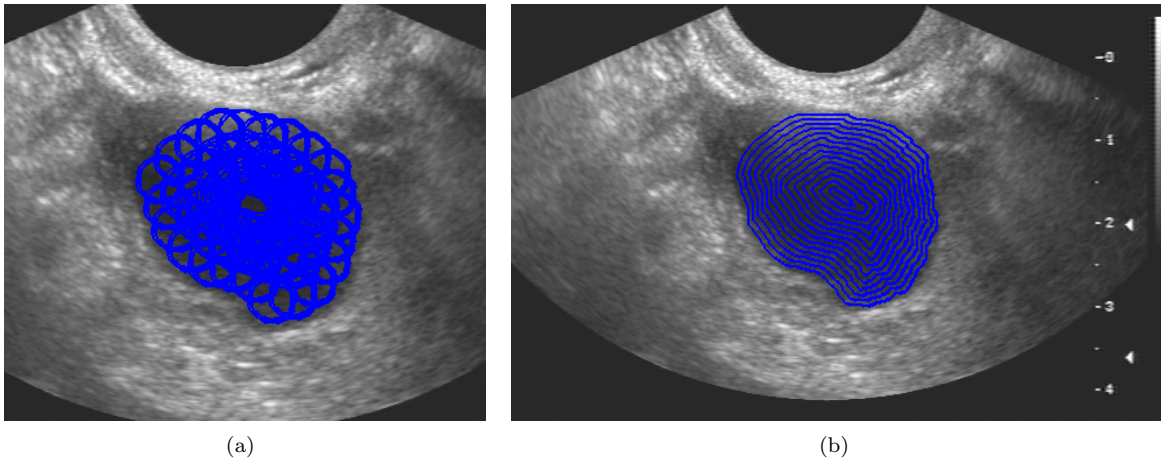


Figure 4.4: (a) Closed contour and (b) Iso-contour inside the follicle region

contours were generated. One type of closed contours, elliptical in shape, was generated in different sizes almost everywhere in the follicle area (Figure 4.4(a)).

Another type of closed contour were generated which was quite different, in shape and number, from the first type. These were iso-contours and the number of these contours was much less compared to that of the first type of contour due to the difference in shapes between these two types (Figure 4.4(b)). The number of these contours was computed using Equation 3.6 using $div = 1.5$.

4.2 Categorization of interactions

In order to analyze the segmentation results, interactions were categorized according to the position of the interaction inside the follicle region, where this position is numerically represented according to the distance of the interaction from the centroid of the follicle. For seed point mode of interaction, seed points are

categorized into three groups, denoted as peripheral, intermediate and central, considering the distance of the seed point from the centre of the follicle using Equation 4.3 for $n = 3$ and where a is the distance of the seed point from the nearest boundary point of the follicle and b is the distance of the seed point from the centroid of the follicle.

$$\mathbf{c} = \begin{cases} \text{peripheral,} & \frac{a}{a+b} \leq 1/3 \\ \text{intermediate,} & 1/3 < \frac{a}{a+b} \leq 2/3 \\ \text{central,} & 2/3 < \frac{a}{a+b} \end{cases} \quad (4.3)$$

Curved brush strokes were categorized into two or three groups depending on the size of the follicle using Algorithm 1 described in Section 3.1.2. For follicles having a diameter larger than 3.75 mm, sampled brush strokes were categorized into three groups: central, intermediate and peripheral depending on their positions within the follicle region from the centroid. For each follicle having a diameter between 2.5 mm and 3.75 mm, sampled brush strokes were similarly categorized into two groups: central and peripheral. Figure 4.3(b) shows the central, intermediate and peripheral strokes in a follicle region.

Straight brush strokes were categorized into two or three groups depending on the size of the follicle. The category of each pixel of a straight brush stroke is first determined applying the same technique which is used for seed points and the total pixel counts for each category is computed. Then the category of a straight brush stroke is determined by majority vote of the categories of the individual pixels that comprise the straight brush stroke.

Closed contour type of interactions were categorized into two or three groups in the same way as brush strokes. For determining the category of an iso-contour, distances of all the pixels of an iso-contour from the boundary of the follicle were computed. Then, the mean of these distances were computed and was used to determine the category of the iso-contour in the same way as for seed points.

4.3 Analysis of the segmentation results

After generating and categorizing the interactions, all the images of the follicle dataset having a total of 81 follicles with diameter > 2.5 mm were segmented by nine segmentation applications using seven different algorithms and three different interaction modes. Then the results were analyzed using several statistical methods which will be explained in the next sub sections.

4.3.1 Mean Segmentation Accuracy Within Interaction Categories

Evaluation of segmentation results start with measuring the accuracy of segmentation using three metrics: Dice, RMSD and HD. For each algorithm and each follicle to be analyzed, the accuracy of the segmentation resulting from each generated foreground stroke was computed using each of the three accuracy measures.

Mean and standard deviation of these metrics for each follicle were computed. According to the proposed methodology, interactions were categorized into different groups and segmentation results were analyzed separately for each group of interactions. This analysis revealed a detailed picture of mean segmentation accuracy within interaction categories presented in figures 4.5 to 4.7. Large volume of findings and observations from these figures can be generalized as a summary that, there can be significant variabilities in the resulting segmentations depending on the position of interactions for individual follicles.

Figure 4.5 presents the mean and standard deviations (as error bars) of Dice for each follicle, grouped by the overall, central, intermediate and peripheral group of interactions (the columns) for all nine algorithms *GCSP*, *GCBS*, *GSC*, *GSCSeq*, *TRC*, *Onecut*, *DRLSE*, *DRLSEIC* and *GCnoSP* (the rows). Large variations in the values of Dice coefficients can be observed from the error bars for two algorithms *DRLSE* and *DRLSEIC*. These large variations indicate that these two algorithms perform poorly for this particular segmentation problem i.e. more affected by the position of interactions. The follicle indices are positioned on the horizontal axis in decreasing order of their cross-sectional diameter. A linear regression line, obtained by using the function option ‘Linear Trendline’ (Excel 2011 for Mac, Microsoft Corporation), fit to this data shows that Dice coefficients generally decrease for smaller follicles. For algorithms *GCSP* (top row) and *GCnoSP* (bottom row), slopes of the linear regression lines not only show clear downward trend but also the slopes are very similar for all groups of interactions, which indicate that for these two algorithms, Dice coefficients decrease with decreasing follicle size no matter where the interactions are provided inside the foreground. For all other algorithms, trends of the linear regression lines are slightly downward except for the *TRC* algorithm (5th row from top to bottom). Slopes of the trend lines for the algorithms *GCBS* (2nd row from top to bottom) and *Onecut* (6th row from top to bottom) are slightly downward and are similar for all groups of interactions, which indicate that for these algorithms, Dice coefficients decrease very slightly with decreasing follicle size for all types of interactions provided by users inside the foreground object i.e., the trend remains same with varying follicle size. For five other algorithms, slopes of the trend lines are not similar for all kinds of interaction groups, which means that, for these algorithms, the trend varies with varying follicle size depending on the type of the interaction group. These five algorithms, except for *TRC*, have, in general, a very slight downward trend of the regression lines, but the slopes are not same for four types of interaction groups. The slopes of the trend lines for the central interactions is largest compared to that for intermediate and peripheral interactions and slopes of the trend lines gradually decrease from left to right order i.e., from central to peripheral order. This indicates that the impact of the follicle size on the Dice coefficients gets weaker as the interactions move from the central region to the peripheral regions inside the foreground object. For the algorithm *TRC*, the trend line is almost horizontal for overall, intermediate and peripheral categories, whereas the trend line is very slightly downward for central interactions, which is an indication that for this algorithm, the Dice coefficient doesn’t change much with varying follicle size except when the interactions are provided in the central follicle region.

Figure 4.6 presents the mean and standard deviations (as error bars) of RMSD for the overall, central,

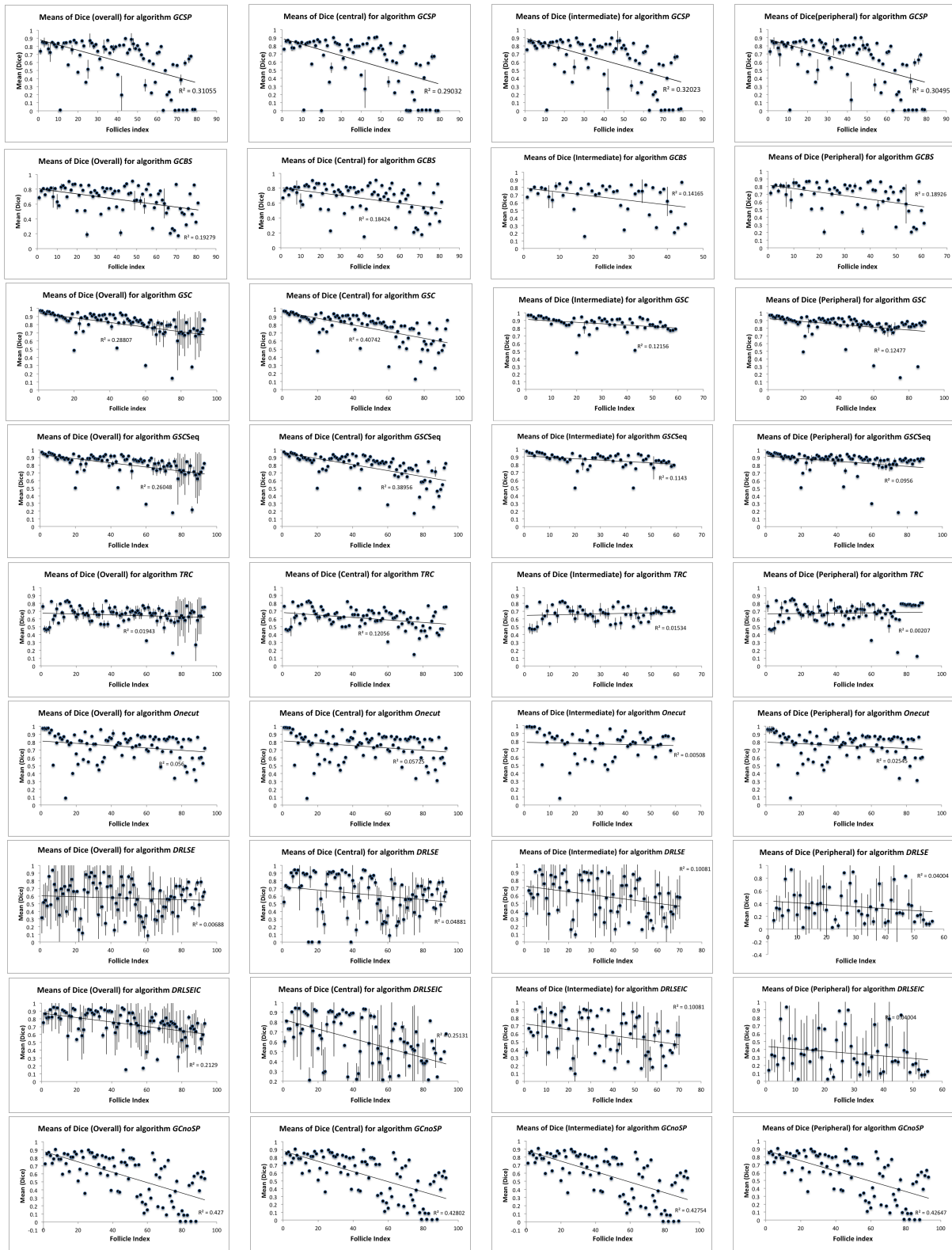


Figure 4.5: Mean and Standard Deviation of Dice with error bar for the overall, central, intermediate and peripheral group of interactions from the left to right order respectively in each row for nine algorithms *GCSP*, *GCBS*, *GSC*, *GSCSeq*, *TRC*, *Onecut*, *DRLSE*, *DRLSEIC* and *GCnoSP* from the top to bottom row order respectively.

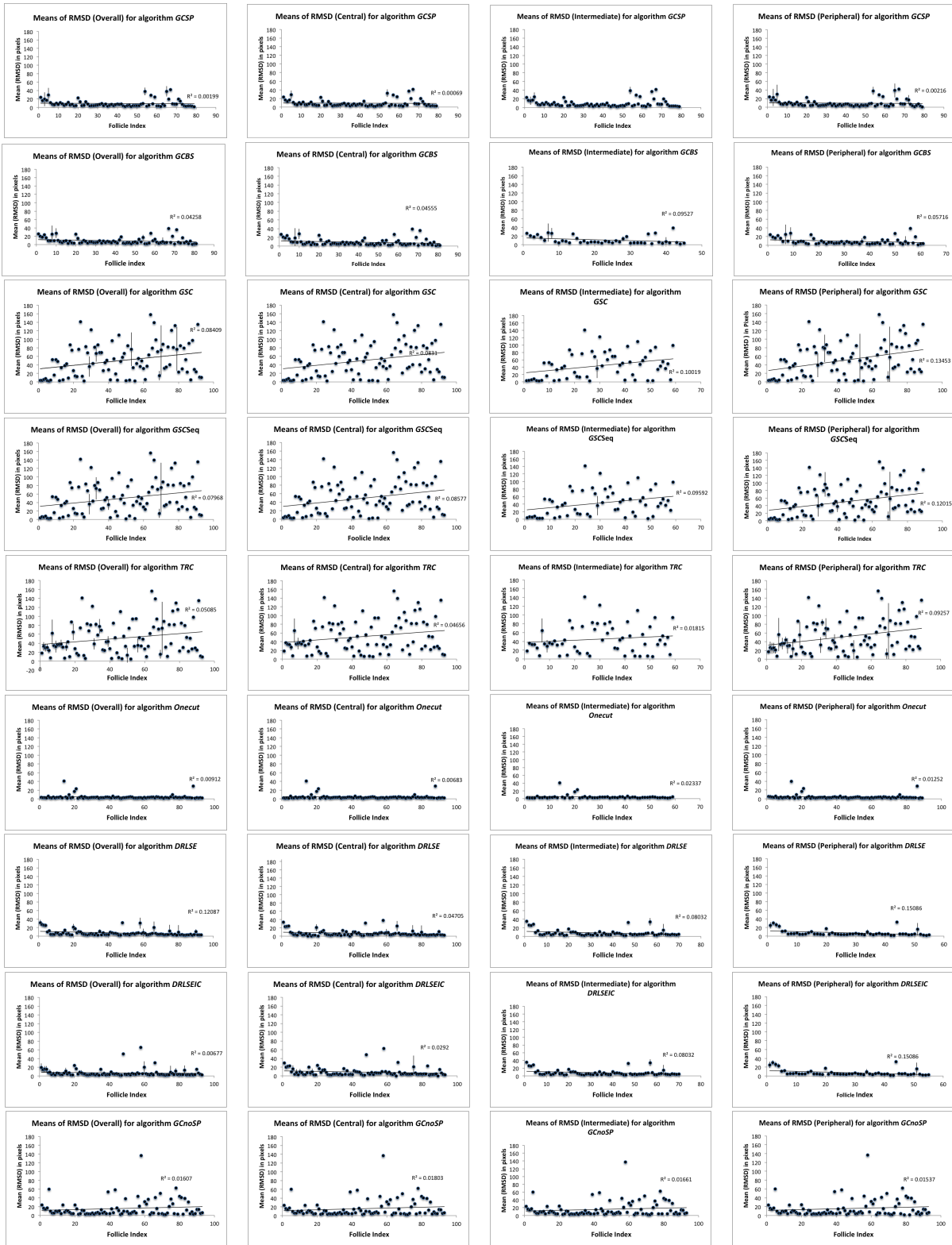


Figure 4.6: Mean and Standard Deviation of RMSD with error bar for the overall, central, intermediate and peripheral group of interactions from left to the right order respectively in each row for nine algorithms *GCSP*, *GCBS*, *GSC*, *GSCSeq*, *TRC*, *Onecut*, *DRLSE*, *DRLSEIC* and *GCnoSP* from top to bottom row order respectively.

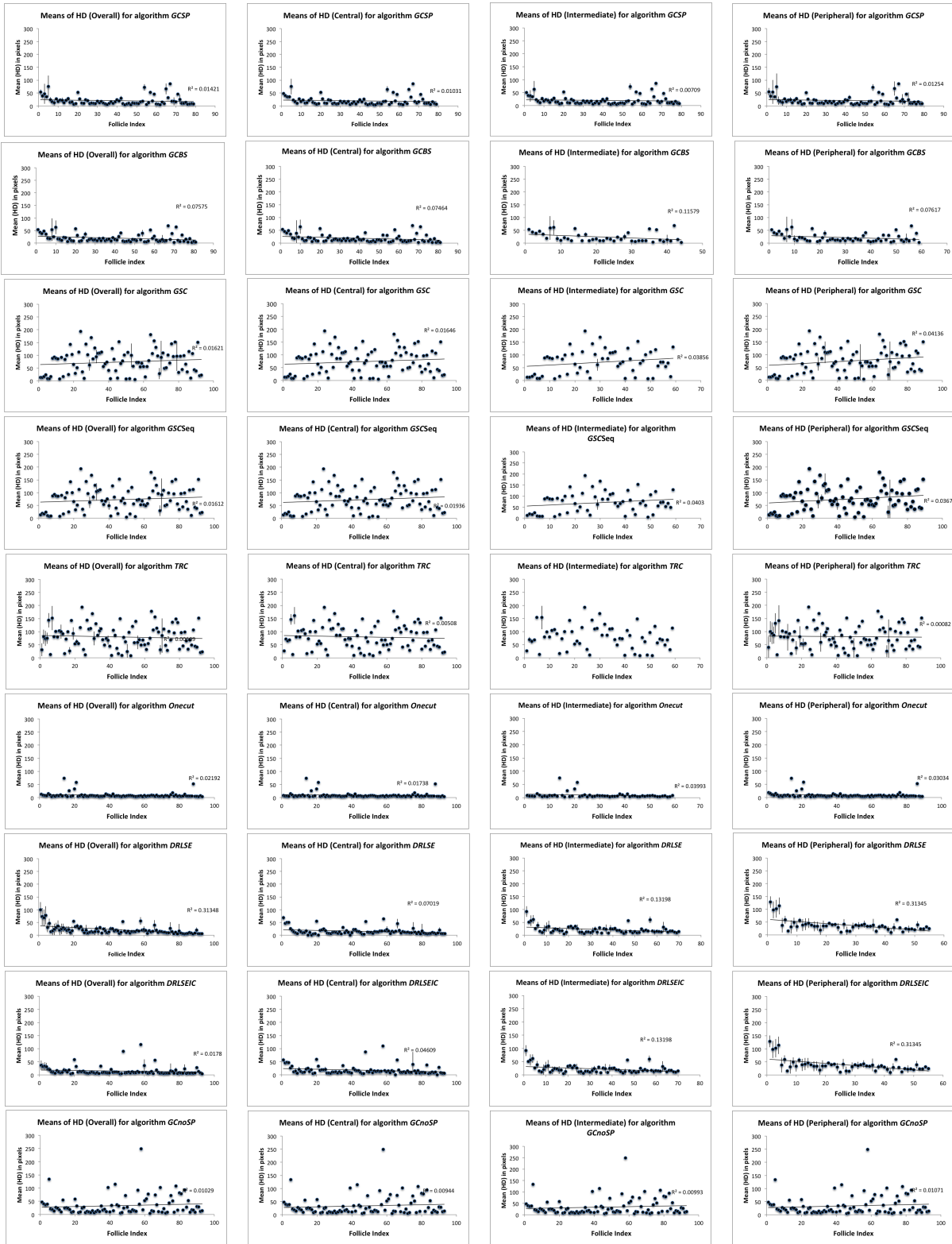


Figure 4.7: Mean and Standard Deviation of HD with error bar for the overall, central, intermediate and peripheral group of interactions from left to right order respectively in each row, for nine algorithms *GCSP*, *GCBS*, *GSC*, *GSCSeq*, *TRC*, *Onecut*, *DRLSE*, *DRLSEIC* and *GCnoSP* from top to bottom row order respectively.

intermediate and peripheral group of interactions in the same fashion and arrangement of algorithms as in Figure 4.5. From the charts in this figure, it is difficult to find out any general trend for the regression lines because the trends are not similar for different algorithms. For algorithms *GCSP* (top row), *Onecut* (6th row from top) and *GCnoSP* (bottom row) trend lines are almost parallel to the horizontal line which shows that the segmentation results, in terms of RMSD, do not vary with varying object size for all types of interaction patterns. For algorithms *GSC* (3rd row from), *GSCSeq* (4th row from top) and *TRC* (5th row from top), trend lines are upward for all kinds of interaction groups, which indicate that RMSD values increase with decreasing follicle size i.e., the segmentation results, in terms of RMSD, get worse as the object size gets smaller. For algorithms *GCBS*, *DRLSE* and *DRLSEIC*, trend lines are slightly downward for all groups of interactions which generally indicate that RMSD values decrease with decreasing size of the objects i.e., the segmentation results, in terms of RMSD, get slightly better as the object size gets smaller. As the y-axes of these charts represent the values of RMSD, the magnitude of these values give an idea about the performance of the segmentation for a particular algorithm for a specific type of interaction pattern. From this view point, the RMSD is relatively small for the algorithms *Onecut*, *GCSP*, *GCnoSP*, *DRLSE* and *DRLSEIC*, which indicate that the segmentation results, for these algorithms are better compared to that of other algorithms. On the other hand, large variations in the values of RMSD are observed for the algorithms *GSC*, *GSCSeq* and *TRC*, which are the indications that, accuracy can vary a lot based on the choice of interaction location. Another interesting findings from these charts is that, for all the algorithms, trends of the regression lines are similar for all types of interaction groups i.e., the interaction pattern has little impact on the trend of the RMSD values.

Figure 4.7 presents the mean and standard deviations (as error bars) of HD for the overall, central, intermediate and peripheral group of interactions again in the same fashion and arrangement as in Figure 4.5. The charts of Figure 4.7 do not show that the different algorithms have different trends with decreasing object size. For algorithms *GCSP* (top row), *TRC* (5th row from top), *Onecut* (6th row from top) and *GCnoSP* (bottom row), trend lines are almost parallel to the horizontal line, which indicates that the segmentation results, do not vary with varying object size for all categories of interactions. For algorithms *GSC* (3rd row from top) and *GSCSeq* (4th row from), trend lines are upward for all categories of interactions which indicate that HD increases (worsens) with decreasing size of the objects. For algorithms *GCBS*, *DRLSE* and *DRLSEIC*, trend lines are slightly downward for all types of interaction patterns which generally indicate that HD decreases (improves) with decreasing size of the objects. From the charts of this figure, it can be observed that the magnitude of HD are relatively small for the algorithms *Onecut*, *GCSP*, *GCnoSP*, *DRLSE* and *DRLSEIC*, which is an indication that the segmentation performance of these algorithms is better than that of other algorithms. By contrast, large variations in the HD values are evident for the algorithms *GSC*, *GSCSeq* and *TRC*, which imply that, accuracy can vary a lot based on the choice of interaction positions. Like RMSD, linear regression lines show the same trends for all groups of interactions for each particular algorithm which is an evidence that segmentation results, in terms of HD, do not depend

much on the position of the interactions inside the follicle region.

In the last few paragraphs, mean segmentation results in terms of Dice, RMSD and HD are discussed from the viewpoint of the impact of follicle size on these results. It was observed that the impact of object size on the segmentation results were not consistent. For each metric, some algorithms are not sensitive to the follicle size irrespective of the position of the interactions inside the follicle region, whereas there is evidence of slight impact of the follicle size on the segmentation results for some of the algorithms. Again, where impact of follicle size is evident, it is not same for all groups of interactions for some of the algorithms. There are some algorithms too, for which impact is visible but the trends of the impact are similar for all groups of interactions. Due to this diverse relation of follicle size with the segmentation result metrics, it has become very important to assess the significance of these regressions in term of the predictor variable i.e., independent variable, *object size*, on the response or dependent variable Dice, RMSD and HD. p values for the independent variable tells us whether the independent variable has the predictive capability which is statistically significant. An independent variable that has a low p value suggests that changes in the values of independent variable are related to changes in the dependent variable. Conversely, a larger p value suggests that changes in the independent variable are not associated with changes in the dependent variable.

Table 4.2 presents the p values of the regressions for Dice, RMSD and HD for overall, central, intermediate and peripheral group of interactions for all nine algorithms. Significant p values in the table are shaded grey. Significant p values suggest that for these regressions, changes in the follicle size are associated significantly with the changes in the corresponding segmentation metric. We can observe that all the entries, except one, for the algorithm *GCSP* have significant p values, which is almost same for algorithms *GCBS*, *GSC*, *GSCSeq* and *DRLSE*, where all the entries, except two, have significant p values. This tells us that, predictive capability of the independent variable *follicle size* on the dependent variable *segmentation metric* are statistically significant for these cases. For algorithm *DRLSEIC*, all the entries for Dice, three entries for HD and one entry for RMSD are significant, which suggests that this algorithm has significant relationship for Dice and HD with follicle size. Algorithm *TRC* has all entries for RMSD and one entry for both Dice and HD are significant, which means this algorithm has significant relationship between RMSD and follicle size. All entries for Dice, for algorithms *Onecut* and *GCnoSP* and one entry for HD, for algorithm *Onecut*, are significant, which suggest that predictive capabilities of *follicle size* on Dice are significant, but not significant for RMSD and HD.

4.3.2 Analysis of Variance of Segmentation Accuracy Between Interaction Categories

In this section, Dice Coefficient, RMSD and HD values for each follicle are used to inspect whether these values for three groups of interactions central, intermediate and peripheral come from the same distribution. If the values for three groups of interactions come from the same distribution, that means there is no impact of the position of the interaction on the resulting segmentation in term of that particular metric for that particular

Table 4.2: P values of Dice, RMSD and HD regressions of three groups of interactions for all nine algorithms

		GCSP	GCBS	GSC	GSCSeq	TRC	Onecut	DRLSE	DRLSEIC	GCnoSP
Dice	Overall	0.00025	0.00475	0.00001	0.00004	0.40526	0.00179	0.86924	0.0019	0
	Central	0.0002	0.00802	0	0	0.43104	0.00144	0.07762	0.00079	0
	Intermediate	0.00018	0.15184	0	0	0.00003	0	0.00008	0.0343	0
	Peripheral	0.0003	0.05392	0.00127	0.00262	0.92533	0.00209	0.00019	0.00441	0
RMSD	Overall	0.0003	0.00071	0.00043	0.00059	0.02651	0.84417	0	0.10105	0.99482
	Central	0.04001	0.00031	0.0005	0.00053	0.04036	0.93618	0	0.00385	0.96316
	Intermediate	0.0553	0.00079	0.78768	0.7547	0.04488	0.08301	3.2907E-13	0.06297	0.98582
	Peripheral	0.0179	0.00021	0.0013	0.00194	0.03532	0.36417	0	0.91599	0.99354
HD	Overall	0.00172	0.00016	0.00408	0.00459	0.80257	0.40155	0	0.03054	0.72278
	Central	0.00398	0.00006	0.0043	0.00401	0.9689	0.63492	0	0.00031	0.7062
	Intermediate	0.00398	0.0008	0.34164	0.31872	0.00023	0.02089	1.11022E-15	0.01756	0.71587
	Peripheral	0.00149	0.00012	0.01204	0.01515	0.33709	0.08533	0	0.83582	0.73073

algorithm. This is a hypothesis test where the null hypothesis is that the metric values for three groups of interaction come from the same distribution. This hypothesis could be tested by applying the classic one way ANOVA but as the values are not normally distributed, a non parametric alternative for classic one way ANOVA, *Kruskal-Wallis Test* has been chosen which does not require the data to be normally distributed. For each algorithm and each follicle to be analyzed, a *Kruskal-Wallis test* is performed for the three populations consisting of the results from each category of interactions and a p value is obtained and then the histogram of the p values have been computed for three different metrics.

Figures 4.8 to 4.10 show the histogram of the p values obtained for each follicle for the metric Dice, RMSD and HD respectively, for each algorithm separately, where the range of the p values is divided into six intervals: [0-0.05), [0.05-0.2), [0.2-0.4), [0.4-0.6), [0.6-0.8) and [0.8-1.0), represented by the x-axis, and the y-axis represents the percentage of the p values that fall into each bin.

Figure 4.11 shows the same histograms for the metric Dice, RMSD and HD, but for all nine algorithms combined together. Thus each bin contains nine values for nine algorithms represented by nine bars of different colours. These aggregated histograms are useful to get the comparative view for all the algorithms.

The histogram for Dice coefficients shows that the percentages of p values in the first bin ([0-0.05)) are quite large for all algorithms except the two algorithms *GCSP* and *GCnoSP*, where 2.5% and 0% of the p values are in the first bin for these algorithms, respectively. The percentage of p values for other algorithms ranges from 35% to 85%, where the algorithm *DRLSEIC* holds the top position. These percentage of p values tell us that, for example, for 85% of the total follicles, null hypothesis is rejected by the *Kruskal-Wallis Test* i.e., there is insufficient evidence to infer that Dice coefficients of three groups of interactions come from the same distribution for 85% of the total objects of interest (follicles). For algorithm *GCnoSP*, none of the p values are significant, which means the null hypothesis cannot be rejected, i.e., there is insufficient evidence to say that Dice coefficients, for all the objects, do not come from the same distribution.

Histograms of p values for RMSD show trends similar to those for Dice coefficients except for the algorithm *DRLSE*, for which only 8% of p values are significant, and the algorithms *GCSP* and *GCnoSP*, where 10%

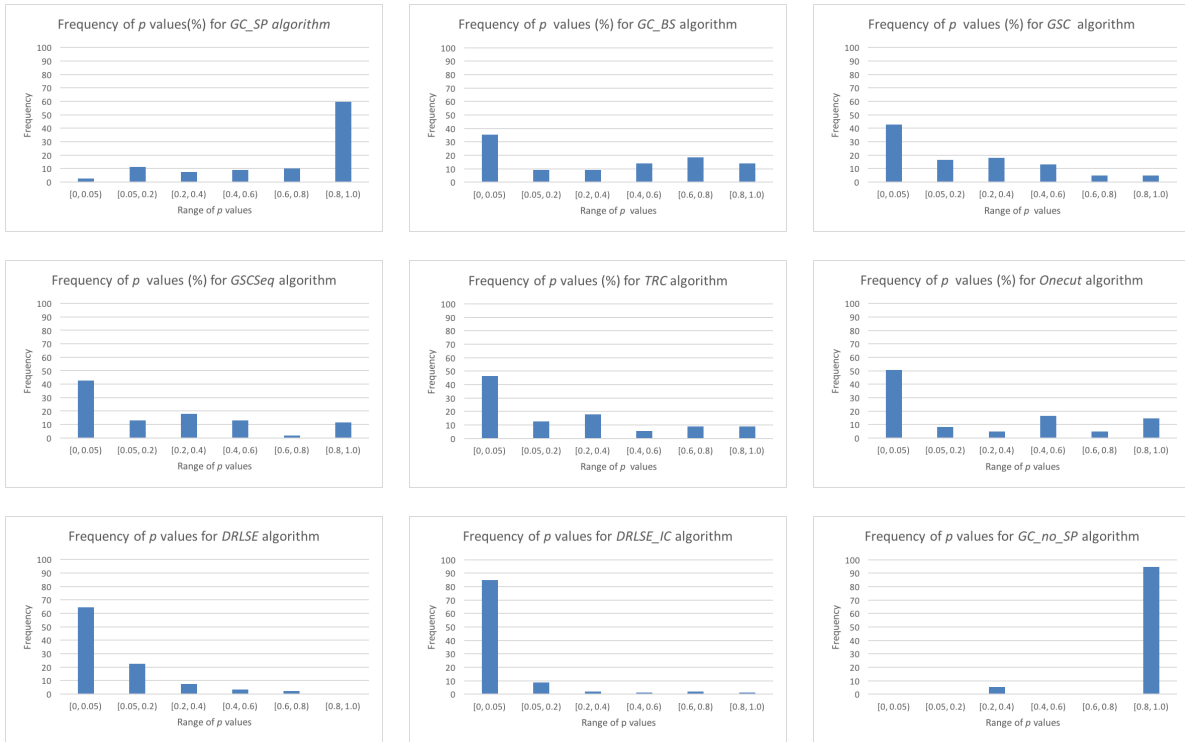


Figure 4.8: Histogram of p values obtained from Kruskal-Wallis Tests for Dice Coefficient for nine algorithms

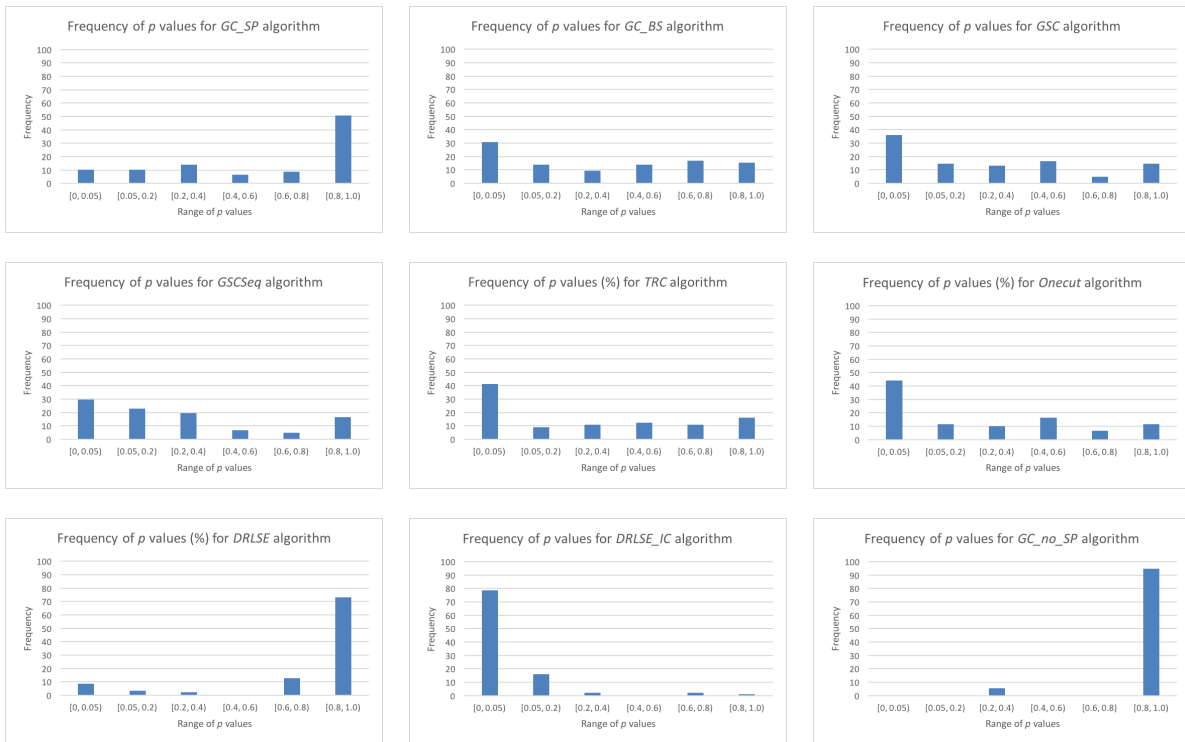


Figure 4.9: Histogram of p values obtained from Kruskal-Wallis Tests for RMSD for nine algorithms

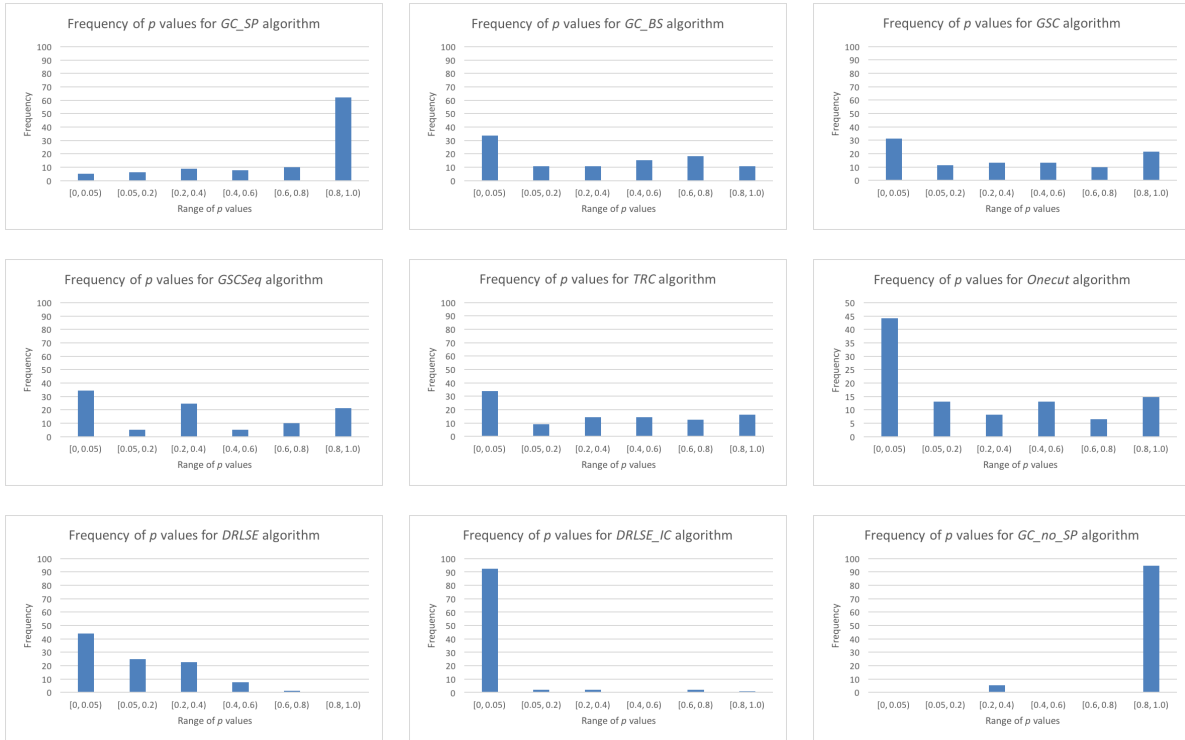


Figure 4.10: Histogram of p values obtained from Kruskal-Wallis Tests for HD for nine algorithms

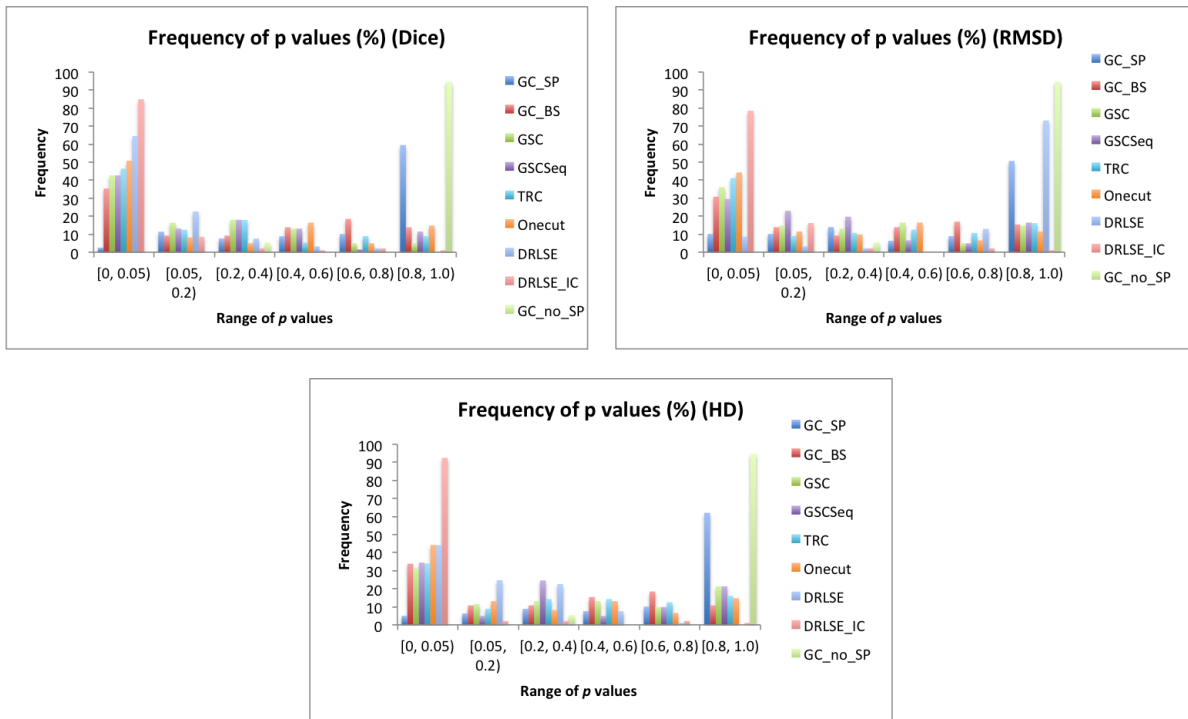


Figure 4.11: Histogram of p values (aggregation of the same data presented in Figures 4.8 to 4.10) obtained from Kruskal-Wallis Tests for Dice Coefficient, RMSD and HD for all nine algorithms

and 0% of the p values are significant for these algorithms, respectively. For all other algorithms, percentage of p values ranges from 30% to 80%, where again DRLSEIC is on the top position. Similar to Dice, the largest percentage of p values in the last bin [0.8-1.0) is for the algorithm *GCnoSP*. Two other large percentage of p values in this bin are for the algorithms *GCSP* and *DRLSE*. These values suggest that p values for all the objects are insignificant for the algorithm *GCnoSP* and are insignificant for almost all the follicles for the algorithms *GCSP* and *DRLSE* i.e., for these algorithms, there is insufficient evidence to conclude that either all or almost all the RMSD values for three groups of interactions do not come from the same distribution.

The histograms for HD values demonstrate trends similar to the Dice results, where the lowest percentage of p values in the first bin are for the algorithms *GCSP* and *GCnoSP* which are 5% and 0%, respectively. Besides these algorithms, the percentage of p values for other algorithms are in the range from 30% to 92%, where the algorithm *DRLSEIC* is again on top with the largest percentage of p values. Algorithms *GCnoSP* and *GCSP* have the largest percentage of p values in the last bin. These values indicate that the HD values for all three groups of interactions come from the same distribution for all or almost all the objects for algorithm *GCnoSP* and *GCSP*, respectively i.e., for these two algorithms, position of the interaction inside the foreground objects has no significant impact on the resulting segmentation in term of HD.

4.3.3 Coefficient of Variation within Interaction Categories

For any kind of segmentation algorithm, accuracy is very important, but for a SIS algorithm, reproducibility is also very crucial because a SIS algorithm may not be useful if it's accuracy is good but reproducibility is not. Coefficient of variation (CV) is used as a quick test to compute the potential reproducibility of all the nine SIS algorithms. CV of the Dice coefficient, RMSD, and HD were computed from the segmentation results generated by the nine combinations of seven algorithms and three interaction modes for each follicle. This quick test is particularly useful to identify whether a segmentation algorithm has poor reproducibility, which suggests that further exploration of that particular algorithm doesn't look promising.

Histograms of the CV values of Dice, RMSD and HD for all nine algorithms for the 81 follicles are shown in Figure 4.12. The range of CV values are divided into ten unequal intervals along the horizontal axis; since most of the values are in the range of 0 to 0.2, this interval is divided into 9 bins which illustrate the distribution of CVs within this range. CV values greater than 0.2 are included in a single bin. The vertical axis represents the percentage of follicles for which the CV fell into the specified range. Values of CV of Dice, RMSD and HD are computed for all of the nine algorithms. Then the CV values are categorized into four groups: overall (includes all interactions generated), central, intermediate and peripheral. For each group of interactions, CV of all nine combinations are presented together.

4.3.3.1 Dice CV

The first column (from left to right) in Figure 4.12 shows histograms of the Dice coefficient CV values for all of the nine algorithms for 81 follicles calculated for the overall, central, intermediate and peripheral groups of interactions.

From the figure for Dice for the group overall, it can be observed that for three algorithms *GCBS*, *Onecut* and *GCnoSP*, more than 50% and for algorithm *GCSP* more than 30% of CV values are in the first bin (0 – 0.0025). So reproducibility, in general, for these four algorithms might be potentially high. Again, among these four algorithms, 90% (highest percentage) CV values for *Onecut* algorithm are in the first bin which clearly keeps it in the leading position in term of potentially high reproducibility. For algorithms *DRLSE* and *DRLSEIC*, 71% and 45% CV values are in the last bin (0.2 – 1.0) and these are the only two significant frequency values in this nine which indicate that, in general, these two algorithms do not have high reproducibility.

For the interactions located in the central region of the foreground object, for seven algorithms out of nine, except *DRLSE* and *DRLSEIC*, more than 50% CV values are in the first bin(0 – 0.0025) and among these seven, more than 70% CV values for five of them are in the first bin (0 – 0.0025), except *GCSP* and *TRC*, for which 54% and 65% CV values are in that range. These values are a clear indication that when the interactions are provided in the central region of the foreground object, segmentation reproducibility might be potentially high for most of the algorithms and interaction modes. For two algorithms, *DRLSE* and *DRLSEIC*, only 3% and 17% CV values are in this range, which is an evidence that segmentation reproducibility of these two algorithms are poor even when the interactions are provided in the central region. The highest percentage of CV values for these two algorithms are in the range 0.2 to 1.0, which also proves the poor reproducibility of segmentation of these algorithms. For this category of interactions, *Onecut* algorithm again takes the leading position, which is a clear indication that this algorithm has the greatest potential for good reproducibility.

For the intermediate category of interactions, four algorithms *GSC*, *GSCSeq*, *Onecut* and *GCnoSP* have more than 70% CV values in the first bin(0 – 0.0025) and three other algorithms *GCSP*, *GCBS* and *TRC* have more than 30% values in this range, whereas two other algorithms *DRLSE* and *DRLSEIC* have very small percentage (less than 10%) of CV values in this range. These numbers show that for this group of interactions, the aforementioned four algorithms might have high segmentation reproducibility whereas three other aforementioned algorithms are average and last two algorithms are poor in terms of potential reproducibility of segmentation. The last range of CV values has got only a single significant frequency value, which is for the algorithm *DRLSE*, which is different from the previous two groups of interactions because the algorithm *DRLSEIC* has a very small percentage of CV values in this range. This phenomenon can be explained in a way that for the algorithm *DRLSEIC*, intermediate group of interactions are actually closer to the boundary of the foreground object compared to the central group of interactions, which forces the segmented boundary close to the actual boundary of the object.

Three algorithms *GCBS*, *Onecut* and *GCnoSP* have more than 50% and three other algorithms *GCSP*, *GSC* and *DRLSEIC* have more than 30% CV values in the first bin (0 – 0.0025) for the peripheral group of interactions. The *Onecut* algorithm is again on top in terms of CV, as it was for all other group of interactions. A moderate CV for the algorithm *DRLSEIC* is not surprising because, for all other algorithms except *DRLSEIC*, peripheral interactions are located in a corner area compared to the object boundary whereas for the algorithm *DRLSEIC*, peripheral interactions are closest to the true boundary, as the interactions are closed contours and almost all the pixels of that contour are in the peripheral region.

4.3.3.2 RMSD CV

The second column (from left to right) in Figure 4.12 shows the distribution of CV values for overall, central, intermediate and peripheral interactions for RMSD for all nine combinations of algorithms and interactions. Six algorithms *GCBS*, *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCnoSP* have more than 50% CV values in the first bin(0 – 0.0025) for overall group of interactions which is an indication that these algorithms are reasonably good candidates for high segmentation reproducibility for all kinds of interaction patterns. One important observation here is that the two algorithms *DRLSE* and *DRLSEIC* have no CV values in this bin at all; rather, 75% CV values of these algorithms are in the last bin(0.2 – 1.0) which suggests that these two algorithms have low reproducibility.

For the central group of interactions, the six algorithms *GCBS*, *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCnoSP* have more than 70% and *GCSP* algorithm has almost 50% CV values in the first bin (0 – 0.0025) which indicates that all these algorithms are likely to have good segmentation reproducibility when interactions are provided in the central region. Algorithms *DRLSE* and *DRLSEIC* have most of their CV values in the last bin(0.2 – 1.0), with no other algorithm having a significant percentage of CV values in this bin, which again shows the trend that these algorithms have poor reproducibility even for the interactions placed in the central region.

For the intermediate group of interactions, more than 75% of CV values are in the first bin(0 – 0.0025), for algorithms *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCnoSP* whereas 34% and 47% CV values are in this bin for the algorithms *GCSP* and *GCBS*, respectively. Very small percentage of CV values in the first bin and largest percentage of CV values in the last bin(0.2 – 1.0) for the algorithms *DRLSE* and *DRLSEIC* again indicate segmentation reproducibility of these algorithms are not good, rather might be worse than the overall and central group of interactions. The frequency of CV values for this group of interactions are roughly smaller in the first bin and larger in the last few bins compared to the central group of interactions, which represents the fact that overall potential segmentation reproducibility gets a little worse when interactions are shifted from the central to intermediate region.

Potential segmentation reproducibility of the algorithms in term of RMSD CV gets even worse when the interactions are placed in the peripheral region, which is evident in the bottom chart of the second column (from left to right) in Figure 4.12. Five algorithms *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCnoSP*

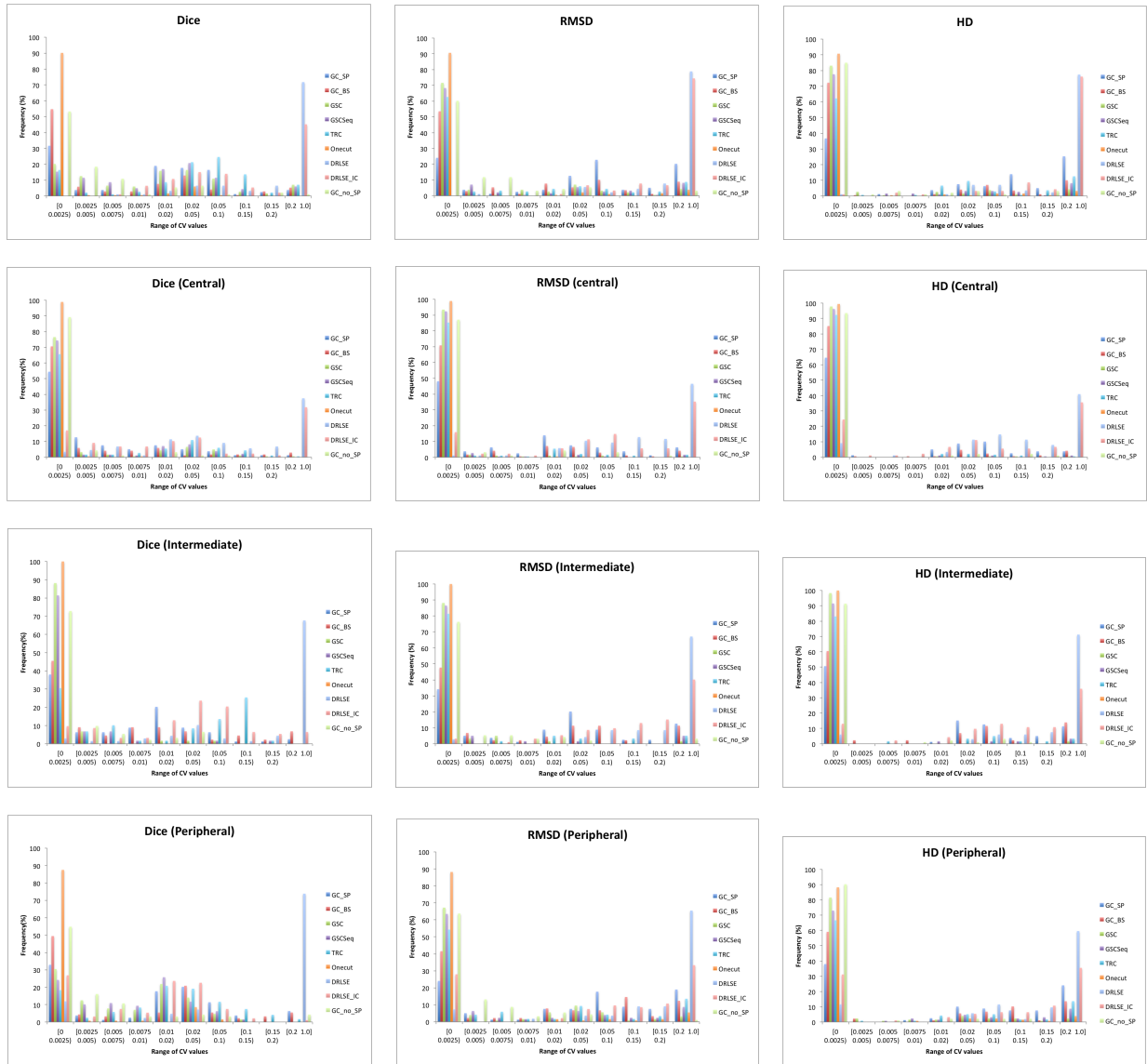


Figure 4.12: Coefficient of variation of Dice, RMSD and HD for four groups of seed points

have more than 50% CV values in the first bin(0 – 0.0025) and the smallest percentage of CV values in this bin is for the algorithm *DRLSE* as was true for all other groups of interactions. The largest percentage of CV values in the last bin (0.2 – 1.0) is again for the algorithms *DRLSE* and *DRLSEIC*, which show the same trend that segmentation reproducibility of these two algorithms is poor for peripheral interactions. Algorithm *DRLSEIC* has close to 30% CV values in the first bin, which indicates that potential segmentation reproducibility of this particular algorithm gets gradually better when interactions move from central to peripheral region, which is an opposite trend compared to all other algorithms. This trend can be explained by the fact that the iso-contour actually gets closer to the actual object boundary in the order from central to peripheral regions, which is completely opposite to other interaction modes. *DRLSEIC* is the only algorithm whose potential segmentation reproducibility gets better for the interaction order from central to peripheral; though, in general, its potential reproducibility is not good, which can be recognized by the large frequency of CV values in the last bin. The potential reproducibility of algorithm *DRLSE* is the poorest among all the algorithms, which is evident by the largest frequency of CV value in the last bin, but the trend is not similar to algorithm *DRLSEIC*.

The potential segmentation reproducibility of the *Onecut* algorithm is the best among all the algorithms, which can be observed by the largest percentage of CV values in the first bin for all groups of interactions. It was also true when segmentation reproducibility was measured using Dice CV.

4.3.3.3 HD CV

The third column (from left to right) in Figure 4.12 shows the distribution of HD CV values for overall, central, intermediate and peripheral interactions for all nine combinations of algorithms and interactions. For all the interactions together, the six algorithms *GCBS*, *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCnoSP* have more than 60% and algorithm *GCSP* has 36% CV values in the first bin(0 – 0.0025). Algorithms *DRLSE* and *DRLSEIC* have almost no CV values in the first bin, whereas these algorithms have almost 80% CV values in the last bin. These numbers represent the fact that these six algorithms have high chance of being reproducible, whereas algorithms *DRLSE* and *DRLSEIC* appear to have poor reproducibility. Another algorithm which has a significant percentage of CV values in the last bin is *GCSP* which has some small percentage of CV values spread across several bins.

All algorithms, except *DRLSE* and *DRLSEIC*, have more than 60% CV values in the first bin (0 – 0.0025) for the central group of interactions whereas algorithms *DRLSE* and *DRLSEIC* have only a small percentage of CV values in this bin. These two algorithms not only have the highest percentage but also have the only significant percentage of CV values in the last bin. These numbers indicate that, except for *DRLSE* and *DRLSEIC*, all other algorithms are likely to have higher reproducibility.

Similar to overall group, the same seven algorithms have more than 50% CV values in the first bin(0 – 0.0025) for the intermediate group of interactions, whereas algorithms *DRLSE* and *DRLSEIC* have a small percentage of CV values in this bin. Similar to the overall and central group of interactions, the largest

frequency values in the last bin are for the algorithms *DRLSE* and *DRLSEIC*. In this case, percentages of CV values are larger than that for central group of interactions.

For the peripheral group of interactions, the six algorithms *GCBS*, *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCnoSP* have more than 50% and the two other algorithms *GCSP* and *DRLSEIC* have more than 30% CV values in the first bin. Only algorithm *DRLSE* has small percentage of CV values in this range. A significant percentage of CV values for algorithm *DRLSEIC* is notable here because most of the algorithms have smaller frequency of CV values in this bin compared to central and intermediate group of interactions. Algorithms *DRLSE* and *DRLSEIC* still have the highest frequencies of CV values in the last bin, though for both of them, frequency of CV values are smaller than for the intermediate group of interactions.

Like Dice coefficient and RMSD, algorithm *Onecut* has again the largest percentage of HD CV values in the first bin ($0 - 0.0025$) for three groups of interactions, which suggests that *Onecut* algorithm is the best candidate among all algorithms in term of potential segmentation reproducibility. In this case, *GSC* and *GCnoSP* algorithms are very close to *Onecut* algorithm in terms of distribution of CV values. Algorithms *DRLSE* and *DRLSEIC*, like Dice coefficient and RMSD, have the highest percentage of CV values in the last bin, which is further evidence that these two algorithms exhibit poorer segmentation reproducibility than all other algorithms. The frequency of CV values for algorithm *DRLSEIC* is almost similar for central, intermediate and peripheral group of interactions, though for algorithm *DRLSE*, these are dissimilar for different groups of interactions. One interesting point to observe here is that the three bins of CV values from second to fourth, contain very few CV values.

4.3.4 Significance test of CV for pairs of interaction groups

From the analysis of the last section, we found that some algorithms have poor reproducibility and some other algorithms have the possibility of having high reproducibility using CV of three accuracy measures. We noticed that reproducibility was not same for all algorithms and all interaction modes. Even for a particular algorithm, reproducibility was not similar for different groups of interactions. This encouraged us to inspect whether the CV values for three different groups of interactions are different or not, from the statistical point of view. We sought to test this between all possible pairs of interaction groups for each algorithm. The CV values of Dice, RMSD and HD for each group of interactions are compared to that of two other groups of interactions to statistically test whether the values for each pair of interaction groups come from the same population. In order to assess this, a non-parametric statistical hypothesis test named ***Wilcoxon rank-sum test (WRS)***, also known as the ***Mann-Whitney U test***, is applied as the CV values are not normally distributed. For each of the nine algorithms, CV values of Dice, RMSD and HD for each group of interactions are paired forming a total of six pairs: overall and central, overall and intermediate, overall and peripheral, central and intermediate, central and peripheral and intermediate and peripheral. Six pairs for each of the metrics Dice, RMSD and HD have made total $6 \times 3 = 18$ combinations for each of the nine algorithms. Each pair of the CV values are tested by the WRS test and a p value as the outcome of the test is reported. Table

4.3 shows the p values for all the tests using all pairs of CV values for all nine algorithms. According to the hypothesis of the WRS test, a p value <0.05 indicates that the null hypothesis that two samples come from the same population is rejected i.e., the pair of CV values are significantly different. On the other hand, p value >0.05 indicates there is insufficient evidence of difference between the two samples to conclude that they are drawn from different distributions. P values <0.05 in the table are shaded gray. 117 out of 162 entries in the table are less than 0.05, which indicates that 72% of the p values are significant i.e., for these cases, CV values of two groups are different. Some of the interesting facts can be observed from this table which need to be explained are given below:

- For all three metrics Dice, RMSD and HD, all the p values in the topmost row are significant i.e., the CV values for the overall and central group of interactions come from different populations for all nine algorithms. This is a clear indication that segmentation results for the interactions in the central region are different than the average segmentation results obtained from considering the whole set of interactions.
- For the metric Dice, all the p values in the fifth row (from the top) are significant i.e., the Dice CV values for the central and peripheral group of interactions do not come from the same population for all of the nine algorithms. This shows that segmentation results for the central interactions are different than that for peripheral interactions. For the metric HD also, all the p values, except for the algorithm *DRLSEIC*, are significant, which indicates that interactions in the central region generate different segmentation results than that for the interactions in the peripheral region.
- For the pair of interactions in the central and intermediate regions, p values are significant for all three metrics, for all algorithms except *DRLSEIC*. This points to the fact that the segmentation results generated by the central and intermediate interactions are different, except for the algorithm *DRLSEIC*, for which there is no significant evidence that the means are different. It also indicates that interactions in the central region are not only different from the peripheral region but also from intermediate regions on average.
- The largest and second largest number of non significant p values are found in the third and last rows for each metric Dice, RMSD and HD, which means that, in most cases, there is insufficient evidence to conclude that the mean performance metrics differ significantly between the intermediate and peripheral regions.
- From the viewpoint of each individual algorithm, the *GSC* algorithm has the highest number of significant p values (17 out of 18 entries) which makes the algorithm most sensitive to the position of the interaction inside the foreground object. By contrast, *GCBS* and *Onecut* algorithm have the lowest and second lowest number of significant p values, which means there is little evidence to conclude that these algorithms are greatly sensitive to the position of interactions.

Table 4.3: P values of Dice, RMSD and HD CV between different groups of interactions for all nine algorithms

Dice	Overall and Central	1.1672e-06	3.2493e-10	1.1793e-32	6.1627e-32	7.1029e-27	6.7891e-07	1.1223e-07	0.0068817	1.0741e-19
	Overall and Intermediate	0.0058	0.61457	4.2399e-20	1.156e-18	0.0006595	0.054298	0.33176	7.351e-08	8.311e-06
	Overall and Peripheral	0.7809	0.43159	0.0005772	0.00027051	1.4573e-10	0.080093	9.9147e-05	7.4021e-18	0.71308
	Central and Intermediate	0.0028	6.4306e-08	1.0865e-07	8.6638e-08	8.2464e-15	0.0044224	0.0002066	0.76806	7.4555e-11
	Central and Peripheral	9.1029e-06	3.1614e-07	4.4222e-26	4.341e-25	4.6272e-18	0.0051626	9.0098e-09	0.00083798	7.1451e-20
	Intermediate and Peripheral	0.0207	0.80435	9.5063e-14	1.4593e-12	0.96087	0.92613	5.0944e-05	1.6997e-06	4.9715e-06
RMSD	Overall and Central	2.3647e-06	5.9566e-11	3.2094e-14	3.0957e-14	9.6484e-12	4.9984e-07	0.00015673	5.3221e-09	6.7315e-19
	Overall and Intermediate	0.00421	0.74846	9.5058e-05	0.0010704	0.012497	0.020442	0.043044	3.3134e-08	3.8476e-08
	Overall and Peripheral	0.8374	0.66185	0.8175	0.48666	0.6114	0.0018664	0.057341	5.8836e-12	0.87447
	Central and Intermediate	0.0132	2.1039e-08	6.4962e-05	1.1215e-05	1.1732e-05	0.017211	0.046607	0.075002	2.7018e-10
	Central and Peripheral	1.7319e-06	8.0056e-08	2.7046e-12	2.6634e-10	9.2565e-09	0.087599	0.1399	0.43324	6.9009e-19
	Intermediate and Peripheral	0.0033	0.84167	0.00062545	0.021939	0.070158	0.58298	0.97818	0.014434	4.0925e-08
HD	Overall and Central	2.8023e-08	4.4888e-08	3.0705e-11	4.5404e-12	1.1054e-10	4.3254e-06	6.0133e-08	1.1881e-10	9.6985e-11
	Overall and Intermediate	0.0039	0.90651	4.9308e-05	0.00085136	0.016019	0.017114	0.20806	7.6983e-09	0.00090676
	Overall and Peripheral	0.8019	0.35433	0.0472	0.13328	0.18867	0.10355	0.10166	4.8013e-11	0.0095579
	Central and Intermediate	1	3.4354e-07	0.012909	0.0010164	3.2766e-05	0.042627	4.661e-05	0.089054	7.6751e-08
	Central and Peripheral	1.5782e-07	1.758e-05	5.3654e-06	4.7046e-07	2.2006e-06	0.0033033	0.0046939	0.99554	2.9474e-09
	Intermediate and Peripheral	0.0186	0.36138	0.031299	0.06481	0.35969	0.39515	0.44215	0.13684	0.26279

4.4 Results and Discussion

Based on the statistical analysis of the results obtained using groups of correct interactions for the nine segmentation applications of seven algorithms combined with three interaction modes, we have the following main results:

1. Reproducibility of segmentation measured by CV of Dice, RMSD and HD, in general, indicate that for all of the algorithms except *DRLSE*, reproducibility worsens in the order of central to the peripheral group of interactions i.e., for better reproducibility, user interactions should be positioned in the central region and peripheral region should be avoided. This result would not be realized by traditional approaches because they do not categorize interactions to determine the impact of the position of interactions on the reproducibility of segmentation.

The magnitude of the gradual change is higher for Dice CV compared to RMSD CV and HD CV i.e., impact of interaction position on the reproducibility of segmentation is more apparent when it is measured by Dice CV. This is evident by the concentration of high frequencies in the first bin which is smaller in the histogram of Dice CV compared to that of RMSD and HD (Figure 4.12). This result also could not be achieved by standard evaluation methods due to the absence of the categorization of interactions.

Onecut is the best among the algorithms in term of potential reproducibility for all groups of interactions measured by all three metrics. *GCBS*, *GCnoSP* are the next two best algorithms which also perform well in terms of reproducibility for all groups of interactions, but the two other algorithms *GSC* and *GSCSeq* perform well only when the interactions are positioned in the central and intermediate regions. In general, it is also true that when interactions are provided in the central region, almost all the algorithms perform well except the algorithm DRLSE which suggests that this algorithm may be not suitable for semiautomatic segmentation for this particular image dataset. These differences in the impact on the reproducibility of segmentation among the different groups of interactions could not be attained with standard methods because they would have computed CVs only for the whole set of interactions.

Segmentation performance, in term of potential reproducibility, of algorithm DRLSEIC is also poor, which is not unexpected. Interestingly, its performance gradually degrades for the groups of interactions in the order from peripheral to central, which is contrary to all other algorithms. This is because the interactions get closer to the true boundary of the object in the order of interaction groups from central to peripheral, which is explained in detail in Section 4.3.3.1. Frequencies of CV values are flat and small in the middle bins of the histogram and get slightly larger for the interaction groups in the order from central to peripheral which is similar to the general trend of the segmentation reproducibility getting worse in the same order. Without categorizing the interactions, we could only know the overall trend of reproducibility for this algorithm and this interesting difference among the effect of different interaction groups could not be discovered.

Here, it is to be noted that, reproducibility is only a measure of consistency of segmentation performance and an algorithm could be precisely reproducible but very poor from the standpoint of accuracy.

2. Potential reproducibility of segmentation is not same when the interactions are placed anywhere inside the object region compared to when they are placed in the central region, which is clearly evident by the first rows (from top to bottom order) for all three metrics Dice, RMSD and HD in table 4.3. Potential reproducibility of segmentation are not same when these are compared between the central and peripheral interactions, which is also true between the interactions provided in the central and intermediate regions. For some of the algorithms, mean Dice CV, RMSD CV and HD CV values are also not the same between the intermediate and peripheral interaction groups, but for some other algorithms, these values come from the same distribution. This suggests that the difference between the intermediate and peripheral interactions, in terms of corresponding impact on segmentation reproducibility, are not always significant. Number of pairs of interaction groups that do not exhibit significant differences, in term of CV values, is smallest for Dice and gradually increases for RMSD and HD, which tells us that the difference between the potential segmentation reproducibility, in terms of CV, are more apparent when these results are measured by RMSD and HD than Dice. These subtle differences among the effect of different interaction groups on segmentation reproducibility could not be obtained using standard

methods because they do not categorize interactions and does not employ statistical hypothesis testing to analyze segmentation performance.

3. The impact of object size on the resulting segmentations is not the same for all algorithms and not same for all groups of interactions too. Four trends are observed in the charts of the Figure 4.5 for Dice:

- (a) Mean Dice decreases with decreasing follicle size and the slope of the trend line is same for all groups of interactions. Algorithms *GCSP*, *GCBS* and *GCnoSP* fall into this category.
- (b) Mean Dice decreases with decreasing follicle size but slope of the trend line is not same for all groups of interactions. Algorithms *GSC*, *GSCSeq* and *DRLSEIC* are included in this category.
- (c) Mean Dice almost does not change with the follicle size and this trend is same for all groups of interactions. Algorithm *Onecut* belongs to this category.
- (d) Mixed trends for different groups of interactions for each particular single algorithm i.e., mean Dice may decrease, increase or remain almost constant with decreasing follicle size for different groups of interactions. This category includes algorithms *TRC* and *DRLSE*.

For RMSD, there were also four trends observed in Figure 4.6:

- (a) Mean RMSD decreases with decreasing follicle size and the slope of the trend line is same for all groups of interactions. Algorithms *GCBS* and *DRLSE* fall into this category.
- (b) Mean RMSD decreases with decreasing follicle size but slope of the trend line is not same for all groups of interactions. Algorithm *DRLSEIC* is included in this category.
- (c) Mean RMSD almost does not change with the follicle size and this trend is same for all groups of interactions. Algorithms *GCSP*, *Onecut* and *GCnoSP* belong to this category.
- (d) Mean RMSD increase with decreasing follicle size for different groups of interactions. This category includes algorithms *GSC*, *GSCSeq* and *TRC*.

Figure 4.7 for HD also presents four kinds of trends which are exactly same as that for RMSD but the list of the algorithms for each type of trend are not same.

- (a) Mean HD decreases with decreasing follicle size and the slope of the trend line is same for all groups of interactions. Algorithm *GCBS* belongs into this category.
- (b) Mean HD decreases with decreasing follicle size but the slope of the trend line is not same for all groups of interactions. Algorithms *DRLSE* and *DRLSEIC* are included in this category.
- (c) Mean HD almost does not change with the follicle size and this trend is same for all groups of interactions. Algorithms *GCSP*, *Onecut*, *GCnoSP* and *TRC* fall into this category.
- (d) Mean HD increase with decreasing follicle size for different groups of interactions. This category includes algorithms *GSC* and *GSCSeq*.

Existing evaluation approaches compute means for the whole set of interactions i.e., does not categorize the interactions into different groups and does not sort the mean according to the object size. All of these findings could not have been discovered without categorizing interactions into different groups and using statistical methods for analyzing the segmentation results.

4. The impact of object (follicle) size on resulting segmentations are diverse and do not follow any particular pattern. For some of the cases, there is no impact at all and for some of the cases, different patterns of impact were observed. Even whenever impacts are evident, not all of them are significant. Significance of these impacts, assessed by the p values of the regressions between the individual metric and object (follicle) size, partly represented by the trend lines in figs. 4.5 to 4.7, suggest that for 31.48% cases, p values are not significant. This tells us that for these cases, *object size* doesn't have predictive capability on the response or dependent variable Dice, RMSD or HD, which means that the changes in the values of these metrics are not associated with the changes in object (follicle) size. According to the proposed methodology, applying the method of statistical analysis, including the use of trend lines and testing significance of the regressions, helped to discover this fine detail about the impact of object size on segmentation results, which couldn't be possible using standard methods of evaluation.
5. In order to understand the impact of position of the interaction inside the object on the segmentation, testing of the hypothesis, whether the Dice, RMSD and HD values for three groups of interactions for each object come from the same distribution, has been conducted for all algorithms. Results of the hypothesis tests show that, for 41%, 31% and 35% of objects (follicles) respectively for Dice, RMSD and HD, metric values do not come from the same distribution considering all nine algorithms. This indicates that, in general, segmentation results differ, depending on the position of the interaction inside the object. This impact again varies depending on the algorithm itself and the particular metric being used for evaluating the segmentation results. These findings could have been revealed only by applying statistical methods for analyzing segmentation results based on categorization of interactions.

Results of this study have revealed many subtle details about the performance of nine segmentation algorithms, which allow us to know about the real characteristics and true capability of each algorithm. Knowing this detailed performance is significant because segmentation performance varies for various reasons which include the algorithm itself, interaction modes, interaction positions, object size, image quality, etc. For comparing the performance of several algorithms, existing methods use only a small number of segmentations for each case, employing on average 3 to 5 human operators. The drawback of these traditional methods is that, there are significant variabilities in the interaction patterns provided by the human operators and a small number of interactions are not enough to capture these variabilities. As the success of SIS applications, in the real world, depends on the end users; they should be informed the best practices for the SIS algorithms to obtain the best possible outcome and for this reason, exploration of the SIS algorithms, in terms of the fine detail and subtle differences regarding the segmentation performance, is essential. This kind of study, for scientifically evaluating the performance of SIS algorithms, can reveal some additional valuable information

about the performance of SIS algorithms, which could not be discovered by the standard method of evaluation due to the following reasons:

1. 3-5 interactions is an insufficiently diverse sampling of the interactions that would be expected to produce a correct segmentation.

Due to these reasons, our proposed methodology employs simulated Interaction models which overcomes the shortcomings of the existing methods in the following ways:

1. A far more diverse and comprehensive sampling of the set of interactions, expected to produce a correct segmentation, are considered.
2. Use of statistical methods for analyzing the results has made this proposed methodology reliable and sound. In order to investigate the impact of the position of interaction on the resulting segmentation, values of Dice, RMSD and HD for all groups of interactions have been tested to inspect whether there is any difference among the means of these values for different groups i.e., whether values of these metrics for three groups of interactions come from the same distribution or not. This hypothesis has been tested using statistical methods, which most of the existing methods do not follow.

Hence the proposed methodology overcomes the flaws of the currently existing methods and provides a scientific approach for in-depth performance analysis of the segmentation algorithms. Use of simulated interaction models and sound statistical methods have made our proposed methodology different from the existing methods by capturing the variabilities in the set of user interactions and revealing subtle differences in the performance of different algorithms that can not be elucidated by established methods.

4.5 Summary of results

The comprehensive description of the results in the last section is very useful to know about the in-depth details and analysis; but to get a brief picture about the relative performance of the nine algorithms is not easy due to the high volume of the results. This section presents the summary of the results, which helps to get an overview about the results at a glance. Figure 4.13 shows the mean and standard deviation of the accuracy measures of the segmentation results in terms of Dice, RMSD and HD for all nine algorithms categorized into overall and three groups of interactions. This figure provides an opportunity to see the comparative results for all nine algorithms as a whole and also categorically based on the groups of interactions. From this figure, it is evident that the algorithms *GSC*, *GSCSeq* and *Onecut* are the top three algorithms considering the segmentation results in term of the Dice coefficient. But the Dice coefficients for different groups of interactions are not consistent, rather variations among these values are visible. In term of consistency of the performance metric Dice coefficients among different groups of interactions, algorithms *GCnoSP*, *Onecut*, *GCBS* and *GCSP* are superior than other algorithms, which is clearly visible. So, in order to assess the performance of the algorithms, both mean accuracy and consistency of the performance metrics among the

categories of interactions should be considered. While selecting an algorithm for a particular application based on performance, a trade-off between mean accuracy and consistency of the performance metrics is essential. From this viewpoint, algorithm *Onecut* can be a good choice. But if mean accuracy is considered more important for a particular application and consistency of the performance metric is bit ignored provided that the placement of the interactions inside the object region can be controlled, then algorithms *GSC* and *GSCSeq* can be better choices. Hence, this comparative view helps users to decide which algorithm to use for a particular application depending on the requirement.

Figure 4.13 also shows the mean and standard deviation of the RMSD values where algorithms *Onecut*, *DRLSE*, *DRLSEIC*, *GCSP* and *GCBS* have the lowest values (low RMSD values indicate good accuracy and vice versa), but in terms of consistency among the values for different groups of interactions, algorithms *Onecut*, *GCnoSP* and *GCSP* are better than other algorithms. So considering both mean accuracy and consistency of the performance metrics among the groups of interactions, algorithm *Onecut* is on top.

When overall mean accuracy of segmentation results are considered in HD, algorithms *Onecut*, *DRLSEIC*, *GCSP*, *GCBS* and *DRLSE* have the best values but consistency of the performance metrics among the values for different groups of interactions for algorithms *DRLSE* and *DRLSEIC* are very poor. In this regard, algorithms *Onecut*, *GCnoSP* and *GCSP* have better consistency of the performance metrics than other algorithms. Hence, considering the balance between mean accuracy and consistency of the performance metrics among the groups of interactions, algorithm *Onecut* secures the top position.

Figure 4.14 shows the percentage of follicles having significant p values indicating whether Dice, RMSD and HD values for different groups of interactions come from the same distribution or not. For all three accuracy measures, algorithm *DRLSEIC* has the highest percentage of follicles having significant p values, which means that Dice, RMSD and HD values for the highest percentage of follicles do not come from the same distribution for this algorithm. Algorithms *Onecut* and *DRLSE* have one of the three highest percentages of follicles having significant p -values for all three accuracy measures. Algorithm *GCnoSP* has the lowest percentage of follicles having significant p values for all three accuracy measures i.e., percentage of follicles is lowest for this algorithm where Dice, RMSD and HD values do not come from the same distribution.

Figure 4.15 presents the percentage of CV values in the range [0-0.0025] for all three accuracy measures Dice coefficient, RMSD and HD for all nine algorithms. Algorithm *Onecut* has the highest percentage of CV values in that range, and algorithms *GCnoSP* and *GSC* are among the top three for two accuracy measures among the three for each case. Algorithms *DRLSE* and *DRLSEIC* have the lowest percentage of CV values for all three accuracy measures.

So, after scrutinizing all these results at a glance, performance of the algorithm *Onecut* sounds very promising and holds the top position considering the mean accuracy and consistency of performance metrics among the different groups of interactions, whereas algorithms *GCSP*, *GCnoSP*, *GCBS* are also fairly good in term of overall performance. Performance of the algorithms *TRC*, *DRLSE* and *DRLSEIC* do not look promising as a whole and should be less preferred for this type of segmentation application.

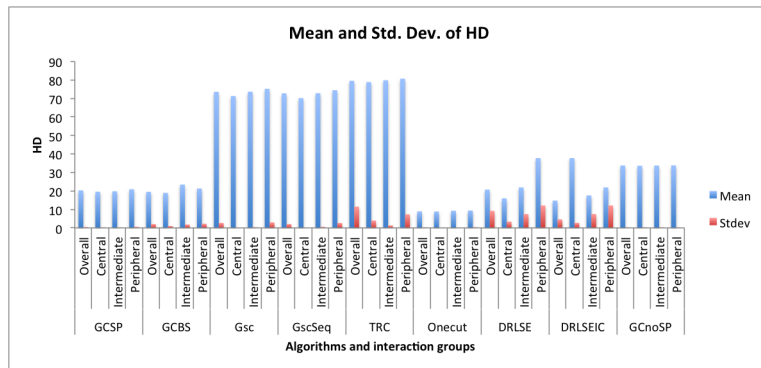
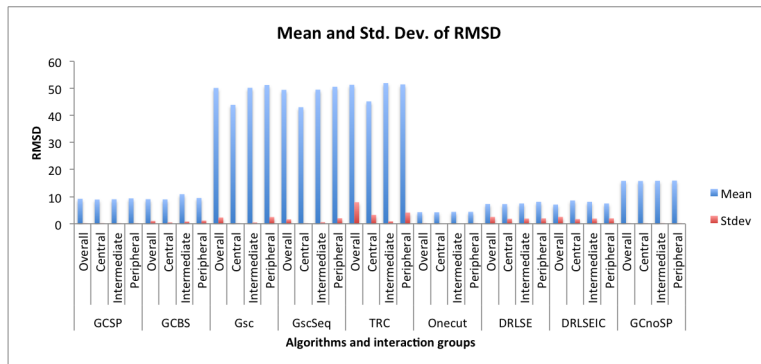
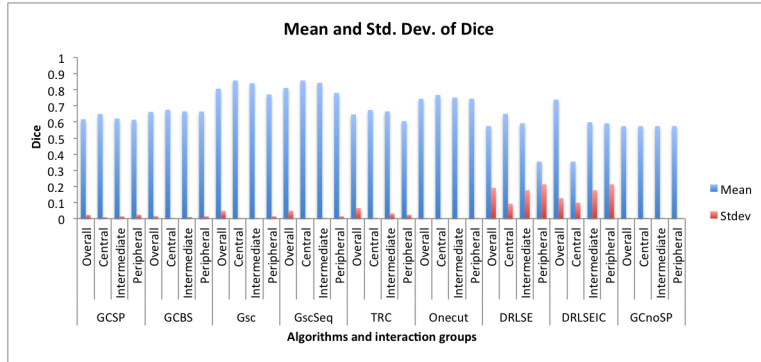


Figure 4.13: Mean and standard deviation of Dice, RMSD and HD for all groups of interactions for all algorithms

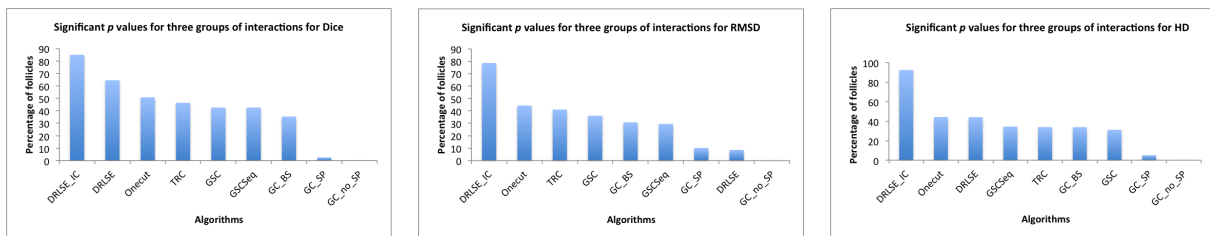


Figure 4.14: Percentage of follicles having significant p values for Dice, RMSD and HD for all algorithms

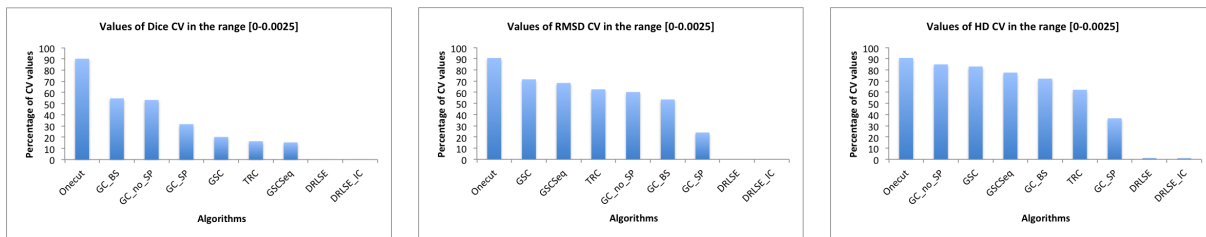


Figure 4.15: Percentage of CV values in the range [0-0.0025] for Dice, RMSD and HD for all algorithms

CHAPTER 5

SEGMENTATION OF FOLLICLES IN ULTRASOUND IMAGES: A CASE STUDY OF THE PROPOSED METHODOLOGY USING LOCAL IMAGE PROPERTIES FOR INTERACTION CATEGORIZATION

The methodology for evaluating the performance of semiautomatic and interactive segmentation (SIS) algorithms, described in Chapter 3, specified two potential criteria for categorizing the interactions. The first was categorizing by spatial position, which was demonstrated in Chapter 4.

The second option was the categorization of interactions by the properties of the image in the vicinity of the interactions. More specifically, these particular regions of the image can be characterized by the properties like texture, intensity values, noise, etc. The methodology for evaluating the performance needs to be adapted in order to reflect the required change in categorization and statistical analysis, which were steps 3 and 4 respectively in the proposed methodology. Before going into the details of the case study, some basic background information about the image texture properties related with this study is presented in the next section.

5.1 Image Texture Properties

Textures are complex visual patterns composed of entities or sub patterns, that have characteristic brightness, colour, slope, size, etc. Texture can be easily perceived by humans and is considered to be a rich source of visual information. There is no generally agreed upon formal or complete definition of texture. Texture can be characterized by the spatial arrangement of colour or intensities in an image. It is a repeated arrangement of primitive elements over the region of an image. It is not a characteristic of a pixel, rather property of a group of pixels. Many researchers have accepted the definition of texture as the variation in the grayscale intensity values in the spatial domain, as this definition has been proven appropriate in a wide range of applications and has been studied intensely during the last few years.

Haralick [55] proposed the use of gray level co-occurrence matrix (GLCM), one of the most widely used general-purpose methods of texture analysis, which defines texture as the spatial distribution of pairs of gray values from which various second-order features are derived. It considers the association between two pixels; the first pixel is known as a reference and the second is known as a neighbour pixel.

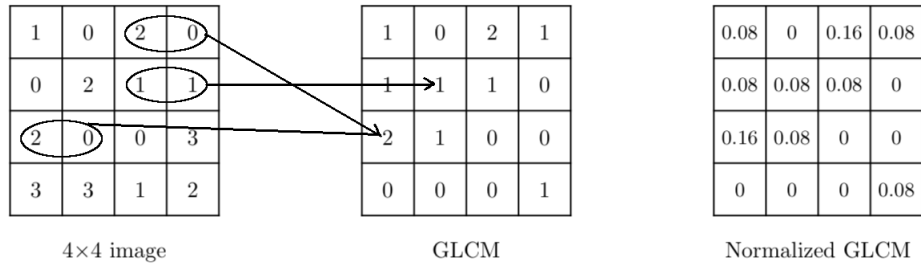


Figure 5.1: A 4×4 image and its GLCM for displacement $d = (1, 0)$

A GLCM is a matrix where the number of rows and columns is equal to the number of gray levels, G , in the image I defined as $I(x, y)$, $0 \leq x \leq N_x - 1, 0 \leq y \leq N_y - 1$ where N_x and N_y are the dimensions of the image along the X and Y axis. The gray level co-occurrence matrix P_d^θ of size $G \times G$ for a displacement vector $d = (\Delta x, \Delta y)$ and direction θ is defined as follows. The element (i, j) of P_d^θ is the relative frequency of the pair of pixels having the gray levels i and j , where the pixels are separated by the displacement d within a given neighbourhood.

$$P_d^\theta(i, j) = \#\{(p, q), (r, s) : I(p, q) = i, I(r, s) = j\} \quad (5.1)$$

where $(p, q), (r, s) \in N_x \times N_y; (r, s) = (p + \Delta x, q + \Delta y)$.

Figure 5.1 shows the co-occurrence matrix P_d^θ with displacement $d = 1$ and the direction is horizontal ($\theta = 0$). This spatial relationship considers the horizontally adjacent pixels. The co-occurrence matrix can be normalized by dividing by the sum of all of its entries. Normalized co-occurrence matrix is represented here as p_d^θ .

From the co-occurrence matrix, Haralick proposed thirteen texture features. Among these, four features: energy, contrast, correlation and homogeneity, were used for this study. These four features were selected because they are well known and proven general-purpose features that are readily present for characterizing the texture properties of ultrasound grayscale images which constitute the dataset used in this study. These four features are described as follows:

5.1.1 Energy

This texture feature is also denoted as angular second moment (ASM) feature.

$$Energy = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p_d^\theta(i, j)^2 \quad (5.2)$$

It is a measure of homogeneity of the image and indicates the uniformity of texture [55]. When pixels are very similar, only a few gray levels will be available, producing a GLCM containing only a few elements with relatively large values, which will result in high sum of squares i.e., energy value.

5.1.2 Contrast

Variations of the intensity or gray level value between the reference and neighbouring pixel is measured by contrast.

$$Contrast = \sum_{n=0}^{G-1} n^2 \left\{ \sum_{i=0}^{G-1} \sum_{j=0; |i-j|=n}^{G-1} p_d^\theta(i, j) \right\} \quad (5.3)$$

For equal values of i and j , the pixel is on the diagonal and $i - j = 0$ which represent pixels that are exactly similar to their neighbour, so they are assigned a weight of 0. If difference between i and j is 1, there is a small contrast, and the weight is 1. If i and j differ by 2, contrast increases and the weight is 4. The weights continue to increase quadratically as $(i - j)$ increases.

5.1.3 Correlation

Correlation measures the linear dependency of gray level values between the pixels at the specified positions relative to each other.

$$Correlation = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p_d^\theta(i, j) \frac{(i - \mu_x)(j - \mu_y)}{\sigma_x \sigma_y} \quad (5.4)$$

where μ_x , μ_y and σ_x , σ_y are the means and standard deviations of p_x and p_y .

$$\mu_x = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} i \cdot p_d^\theta(i, j) \quad \mu_y = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} j \cdot p_d^\theta(i, j) \quad (5.5)$$

$$\sigma_x = \sqrt{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu)^2 p_d^\theta(i, j)} \quad \sigma_y = \sqrt{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (j - \mu)^2 p_d^\theta(i, j)} \quad (5.6)$$

It shows the relation of a reference pixel to its neighbour where 0 is uncorrelated and 1 is perfectly correlated.

5.1.4 Homogeneity

This feature is also denoted as the inverse difference moment (IDM) feature.

$$Homogeneity = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{p_d^\theta(i, j)}{1 + |i - j|^2} \quad (5.7)$$

IDM weight value is the inverse of the Contrast weight. It measures the closeness of the distribution of the GLCM elements to the GLCM diagonal. High homogeneity of any texture means that there are a lot of

pixels with the same or very similar gray level value along the GLCM diagonal i.e., large number of nonzero entries are concentrated along the diagonal or close to the diagonal.

5.2 Evaluation of SIS algorithms following the methodology

5.2.1 Segmentation and Generation of Interactions

The first step of the methodology is to segment the set of images using an SIS algorithm for a particular type of interaction mode. In order to demonstrate the proposed methodology, a total of seven SIS algorithms were combined with three types of interaction modes, making a total of nine segmentation applications. These have been implemented and used to segment the set of images described in Chapter 4. The second step of the methodology is to generate interactions programmatically, which was described in Section 4.1 for three interaction modes. In the next section (5.2.2), the procedure used for categorizing interactions into groups by similarity of local image features is described. Section 5.2.3 describes the analysis of these groups of features using the methods established in Chapters 3 and 4.

5.2.2 Categorization of interactions

All the interactions of three types: seed point, brush stroke and closed contour, are categorized based on the values of several image features, including the aforementioned four GLCM texture features. Each interaction is categorized for each feature separately. For each feature, categorization is performed by grouping interactions into three groups which have ‘low’, ‘medium’ and ‘high’ values for each feature. ‘Low’, ‘medium’ and ‘high’ ranges for a feature value are found by grouping the feature values into three categories using the K-means clustering algorithm. As in Chapter 4, interactions are categorized in order to investigate variations in segmentation performance between categories. In contrast to Chapter 4, interaction categories in this experiment are based on local image intensity and texture features, rather than spatial location.

5.2.2.1 Image features for each interaction

A number of image features are computed for each interaction to be used for categorization of the interactions as described in the previous section.

For seed points, the following features are used where window size 1 was used for calculating the GLCM:

- GLCM Contrast
- GLCM Correlation
- GLCM Entropy
- GLCM Homogeneity
- Seed point intensity

Features	Acronyms	Range of Values		
		0-36	36-72	72-180
Seed point intensity	SPI	0-38.0617	38.0617-73.0247	73.0247-192.4198
Mean neighbour pixel intensity	NMI	0-8.3919	8.3919-17.1498	17.1498-70.5652
Std Dev of neighbour pixel intensity	NSI	0-113.1452	113.1452-427.1738	427.1738-1172.6262
Mean contrast	MC	0-0.47322	0.47322-0.86593	0.86593-1.0
Mean correlation	MCR	0-0.050624	0.050624-0.16424	0.16424-0.32988
Mean energy	ME	0-0.32513	0.32513-0.49806	0.49806-0.83024
Mean homogeneity	MH			

Table 5.1: Features and bins used for seed point interaction

- Mean intensity of pixels within a 10-pixel radius of the seed point
- Standard deviation of intensity of pixels within a 10-pixel radius of the seed point

For brush strokes and closed contours, the following features are used where all pixels belonging to the interaction were included in the GLCM:

- GLCM Contrast
- GLCM Correlation
- GLCM Entropy
- GLCM Homogeneity
- Mean intensity of all pixels in the interaction
- Standard deviation of all pixels in the interaction
- Mean over all pixels, p , in the interaction, of the mean intensity of pixels within a 10-pixel radius of p
- Mean over all pixels, p , in the interaction, of the standard deviation of intensity of pixels within a 10-pixel radius of p

5.2.2.2 Categorization using image features

For each feature, low, medium and high ranges for the feature are established using K-means clustering of the feature values recorded over all interactions. Table 5.1, 5.2, 5.3 and 5.4 present the names of the features with their acronyms used throughout this chapter and corresponding ranges obtained from K-means algorithm for each of the feature values for the seed point, brush stroke, closed contour and closed iso-contour interactions, respectively.

All the interactions for each of the interaction modes seed point, brush stroke, closed contour and closed iso-contour are grouped into the three ranges determined for each of the feature values. For each group of interactions defined in this way, the aggregate performance of the segmentations produced by the interactions in the group are determined in terms of Dice, RMSD and HD. This grouping of interactions allow us to understand the distribution of the feature values across the interactions and corresponding mean segmentation results tell us about the relation of these feature values to the segmentation results. However, histograms

Features	Acronyms	Range of Values		
Mean Intensity of the pixels	MI	0-36.6	36.6-69.0286	69.0286-142.9356
Std Dev of pixel Intensity	SI	0-7.4172	7.4172-16.0033	16.0033-48.995
Mean neighbour pixel intensity	NMI	0-37.0004	37.0004-67.9921	67.9921-144.6095
Std Dev of neighbour pixel intensity	NSI	0-6.954	6.954-13.412	13.412-41.7785
Mean contrast	MC	0-42.8025	42.8025-229.6551	229.6551-849.2246
Mean correlation	MCR	0-0.44411	0.44411-0.8422	0.8422-1.0
Mean energy	ME	0-0.052468	0.052468-0.13834	0.13834-0.27694
Mean homogeneity	MH	0-0.32218	0.32218-0.47625	0.80553-1.0

Table 5.2: Features and bins used for brush stroke interaction

Features	Acronyms	Range of Values		
Mean Intensity of the pixels	MI	0-35.3846	35.3846-64.8199	64.8199-115.3481
Std Dev of pixel Intensity	SI	0-6.1525	6.1525-12.094	12.094-35.1476
Mean neighbour pixel intensity	NMI	0-35.6296	35.6296-64.3135	64.3135-113.4973
Std Dev of neighbour pixel intensity	NSI	0-6.5721	6.5721-12.3621	12.3621-38.0648
Mean contrast	MC	0-41.9104	41.9104-158.9534	158.9534-886.1909
Mean correlation	MCR	0-0.43463	0.43463-0.83656	0.83656-1.0
Mean energy	ME	0-0.028037	0.028037-0.068989	0.068989-0.28928
Mean homogeneity	MH	0-0.31497	0.31497-0.46858	0.81314-1.0

Table 5.3: Features and bins used for closed contour interaction

Features	Acronyms	Range of Values		
Mean Intensity of the pixels	MI	0-26.7439	26.7439-39.4167	39.4167-75.4812
Std Dev of pixel Intensity	SI	0-6.743	6.743-13.6407	13.6407-32.1475
Mean neighbour pixel intensity	NMI	0-33.6823	33.6823-49.2474	49.2474-76.0547
Std Dev of neighbour pixel intensity	NSI	0-6.2592	6.2592-11.9467	11.9467-24.4362
Mean contrast	MC	0-47.254	47.254-159.425	159.425-663.472
Mean correlation	MCR	0-0.44631	0.44631-0.83175	0.83175-1.0
Mean energy	ME	0-0.027805	0.027805-0.071746	0.071746-0.25301
Mean homogeneity	MH	0-0.31159	0.31159-0.4656	0.4656-0.78875

Table 5.4: Features and bins used for iso-contour interaction

of the interactions associated with the corresponding segmentation results do not necessarily confirm any significant impact of the feature values on these segmentation results. Existence and extent of this impact is measured by the statistical analysis, which is explained in the next section.

5.2.3 Analysis of the results using statistical methods

In order to investigate whether local image properties have any impact on segmentation, the relation between image properties of the interactions and corresponding segmentation results need to be studied. For SIS algorithms, interactions are used to generate segmentations, and image properties of the interactions are computed. After computing the image properties, interactions are categorized into three groups according to the feature values selected as thresholds. Then means of the corresponding resulting segmentations according to the interaction categories based on feature values are computed.

5.2.3.1 Mean segmentation accuracy among the groups of interactions based on feature values

Here we present the mean Dice, RMSD and HD values of the segmentations generated by the interaction groups. Tables 5.5 to 5.10 present the segmentation performance metrics for each group of interactions.

5.2.3.1.1 Brush Stroke Interactions Table 5.5 presents the mean Dice coefficient values for the groups of brush stroke interactions generated by the algorithms *GSC*, *GSCSeq*, *TRC*, *Onecut*, and *GCBS*. Some of the key points from these results are summarized as follows:

- Among the eight features, exactly the same trend is observed for four features MI, NMI, MCR and ME where number of strokes is lowest for the highest range of the intensity, but mean Dice value is highest for these strokes for almost all algorithms except one algorithm *TRC*.
- The results for MH are similar to those for MI, except the behaviour of algorithm *TRC* has been replaced by the algorithm *GSC* in this case.
- For SI, the trend is same, i.e., mean Dice value is highest for the strokes in the third bin but the difference is that it is true for all algorithms, whereas for the features MI, NMI, MCR and ME, algorithm *TRC* is an exception. It suggests that segmentation results are relatively better when the strokes are placed in the region where variability of the pixel intensities are higher.
- NSI as a feature yields the opposite trend of the features MI, NMI, MCR and ME where mean Dice is smallest for the third bin for all the algorithms except *TRC*. This is an example which could never be discovered by the existing standard evaluation methodologies.
- For the feature MC, mean Dice is highest for the third bin for the algorithms *GSC*, *GSCSeq* and *Onecut*, whereas mean Dice is lowest for this bin for the algorithms *TRC* and *GCBS*.

Table 5.5: Dice values of the segmentations generated by algorithms *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCBS* for brush stroke interactions categorized according to the thresholds of the eight feature values.

Feature	Range of Values	Number of Stroke	GSC	GSCSeq	TRC	Onecut	GCBS
Mean Intensity(MI)	0-36.6	560	0.83072	0.83491	0.65264	0.72125	0.56664
	36.6-69.0286	322	0.93068	0.92248	0.58406	0.92497	0.73567
	69.0286-142.9356	69	0.95937	0.95207	0.5817	0.95667	0.76925
Stdev Intensity(SI)	0-7.4172	632	0.87253	0.87363	0.61776	0.81015	0.63282
	7.4172-16.0033	271	0.87157	0.86647	0.63618	0.79484	0.6471
	16.0033-48.995	48	0.90506	0.90266	0.64271	0.8403	0.66609
NbrMeanIntensity(NMI)	0-37.0004	538	0.83151	0.83621	0.65184	0.72449	0.56862
	37.0004-67.9921	339	0.92267	0.91493	0.59062	0.90588	0.72157
	67.9921-144.6095	74	0.95861	0.95163	0.57797	0.95786	0.76686
NbrStdevIntensity(NSI)	0-6.954	577	0.88176	0.88117	0.60768	0.82512	0.64654
	6.954-13.412	274	0.86993	0.86855	0.65007	0.79521	0.63371
	13.412-41.7785	100	0.83942	0.83863	0.64926	0.73769	0.6059
Mean Contrast(MC)	0-42.8025	711	0.89263	0.89107	0.6267	0.83169	0.64227
	42.8025-229.6551	225	0.81657	0.82289	0.61294	0.73535	0.61288
	229.6551-849.2246	15	0.84589	0.77157	0.67903	0.73103	0.84872
Mean Correlation(MCR)	0-0.44411	131	0.83656	0.84851	0.62127	0.70287	0.50964
	0.44411-0.8422	248	0.84407	0.83757	0.65824	0.7226	0.60246
	0.8422-1.0009	572	0.89538	0.89406	0.61023	0.86796	0.68376
Mean Energy(ME)	0-0.052468	862	0.87325	0.87238	0.62448	0.80305	0.64277
	0.052468-0.13834	69	0.86154	0.86437	0.6375	0.83209	0.54715
	0.13834-0.27694	20	0.94467	0.93241	0.56962	0.90533	0.77293
Mean Homogeneity(MH)	0-0.32218	430	0.85491	0.8545	0.61717	0.78	0.62397
	0.32218-0.47625	376	0.89132	0.88755	0.62363	0.81652	0.64977
	0.47625-0.80553	145	0.88818	0.89051	0.64697	0.86442	0.65284

Table 5.6 shows the analogous results for RMSD. From these results, some of the interesting observations are stated as follows:

- RMSD values are very small for the algorithm *Onecut* for all features (High values of RMSD indicate poor accuracy and vice versa).
- For MI and NMI, RMSD values are smallest in the high range compared to the strokes for two other ranges for the algorithms *GSC*, *GSCSeq* and *TRC*. For this range, RMSD values for the algorithms *GSC*, *GSCSeq* and *Onecut* are very small compared to two other algorithms. For particularly algorithm *Onecut*, RMSD values are small for all three ranges. These fine details about the performance of the algorithm with respect to the local image properties is an example which could not be achieved through the conventional evaluation methodologies due to the absence of large number of regularly sampled interactions and categorization of interactions based on the feature values.
- The trends for SI feature are similar to those for MI. RMSD values are lowest for the third range but the magnitude of the RMSD values are much larger than that for MI, except for the algorithm *Onecut*.
- For the feature SI, RMSD values are highest for the third range for the algorithms *GSC*, *GSCSeq* and *TRC* compared to that for two other ranges.
- RMSD values are lowest for the third range for all algorithms for the feature MC. RMSD value is very small for the third range for algorithm *GCBS*.

Table 5.6: Segmentation result RMSD for algorithms that used brush strokes interactions

Feature	Range of Values	Number of Stroke	GSC	GSCSeq	TRC	Onecut	GCBS
Mean Intensity(MI)	0-36.6	560	43.6608	42.8925	46.5645	5.2844	11.9902
	36.6-69.0286	322	16.9823	17.7819	33.6892	2.6231	16.7194
	69.0286-142.9356	69	4.7324	5.5656	21.5925	4.7968	20.0092
Stdev Intensity(SI)	0-7.4172	632	30.4615	30.4627	40.0219	4.2669	14.3347
	7.4172-16.0033	271	36.233	35.5424	42.2696	4.3067	12.912
	16.0033-48.9946	48	24.4599	25.9407	34.6886	5.6473	19.1688
NbrMeanIntensity(NMI)	0-37.0004	538	43.3572	42.8289	46.5375	5.3788	12.1896
	37.0004-67.9921	339	19.3588	19.4838	34.713	2.6427	15.9946
	67.9921-144.6095	74	4.8114	5.5824	21.7442	4.6646	20.2512
NbrStdevIntensity(NSI)	0-6.954	577	27.2219	27.0847	38.2467	4.3722	15.3026
	6.954-13.412	274	37.6297	37.522	42.6763	4.3575	12.8885
	13.412-41.7785	100	42.2731	42.2071	46.5234	4.1817	11.1774
Mean Contrast(MC)	0-42.8025	711	29.1781	29.0023	38.9841	4.086	15.4467
	42.8025-229.6551	225	40.5231	40.3476	45.731	5.2183	10.8659
	229.6551-849.2246	15	25.4361	28.7142	27.1178	3.7051	3.424
Mean Correlation(MCR)	0-0.44411	131	49.4491	46.1881	57.6939	3.2664	10.9512
	0.44411-0.8422	248	46.6897	47.0922	45.8228	6.331	12.72
	0.8422-1.0009	572	21.3077	21.6785	34.0769	3.7358	15.5413
Mean Energy(ME)	0-0.052468	862	30.9447	30.8328	40.0859	4.4296	13.8123
	0.052468-0.13834	69	46.2566	45.3257	49.1782	3.1583	18.7829
	0.13834-0.27694	20	18.9402	21.2098	23.3304	4.9328	13.8284
Mean Homogeneity(MH)	0-0.32218	430	32.8269	33.6516	41.9904	4.6604	13.9754
	0.32218-0.47625	376	29.104	28.1473	39.7899	4.3928	14.2666
	0.47625-0.80553	145	35.6593	35.0068	37.2213	3.3051	14.5182

- Trends for MCR are similar to those of MC except for the algorithm *Onecut*, for which all three RMSD values are very small but the lowest value is for the first range instead of the third range as in the case of all other algorithms.
- MCR also shows trends similar to those for ME, except for the algorithm *GCBS*.
- For the feature MH, RMSD values, for each particular algorithm, are close for all three ranges. Beside the smallest for the algorithm *Onecut*, RMSD values are also reasonably small for the algorithm *GCBS*.

Table 5.7 shows the results for HD (High HD values indicate poor accuracy and vice versa). From these results, some of the interesting observations are stated as follows:

- HD values are smallest for the algorithm *Onecut* for all features among all algorithms.
- For the features, MI and SI, HD values are lowest for the third range for algorithms *GSC*, *GSCSeq* and *TRC*, whereas HD values are highest for this range for two other algorithms.
- An almost similar trend, like that described in the previous two bullet points, is also observed for the feature NMI, except for the algorithm *TRC* for which lowest HD value is for the second range.
- Trend for the feature NSI is just opposite to that for the features MI and SI i.e., HD values are highest for the third range for algorithms *GSC*, *GSCSeq* and *TRC* and lowest for the third range for two other algorithms.

Table 5.7: Segmentation result HD for algorithms that used brush strokes interactions

Feature	Range of Values	Number of Stroke	GSC	GSCSeq	TRC	Onecut	GCBS
Mean Intensity(MI)	0-36.6	560	69.9194	68.7222	88.2278	10.9733	24.5167
	36.6-69.0286	322	31.5615	32.8513	63.9334	7.6085	34.9135
	69.0286-142.9356	69	15.2451	16.793	70.2599	14.1366	43.5166
Stdev Intensity(SI)	0-7.4172	632	51.4628	51.4452	77.9521	9.7696	29.7976
	7.4172-16.0033	271	58.4845	57.7642	82.08	9.9875	26.5642
	16.0033-48.9946	48	41.5783	42.7882	69.4302	14.3624	40.4816
NbrMeanIntensity(NMI)	0-37.0004	538	69.3663	68.506	88.9237	11.16	24.9411
	37.0004-67.9921	339	35.1243	35.4851	64.2623	7.4804	33.3611
	67.9921-144.6095	74	15.451	16.7848	70.4891	13.9253	43.8704
NbrStdevIntensity(NSI)	0-6.954	577	47.2737	47.175	77.2937	10.1633	31.6244
	6.954-13.412	274	60.0489	59.9639	81.4533	10.0807	27.0224
	13.412-41.7785	100	66.3926	65.7119	79.2538	9.4409	23.2267
Mean Contrast(MC)	0-42.8025	711	49.3989	49.0865	76.0379	9.712	31.6323
	42.8025-229.6551	225	65.1846	65.1576	89.6853	11.3885	23.8322
	229.6551-849.2246	15	38.6926	44.0215	39.9916	6.8495	8.0879
Mean Correlation(MCR)	0-0.44411	131	73.9172	68.9688	102.3032	6.8954	22.1767
	0.44411-0.8422	248	74.6088	74.6763	83.3029	12.7837	25.4944
	0.8422-1.0009	572	38.7822	39.627	71.2958	9.6097	32.7734
Mean Energy(ME)	0-0.052468	862	52.1036	51.9622	78.2644	10.2093	28.6898
	0.052468-0.13834	69	69.6951	68.3493	83.181	7.451	38.3753
	0.13834-0.27694	20	32.3637	35.6877	81.9304	12.7949	29.7812
Mean Homogeneity(MH)	0-0.32218	430	54.9069	56.1379	81.6625	10.4051	29.3043
	0.32218-0.47625	376	49.6765	48.3057	78.3659	10.322	29.3083
	0.47625-0.80553	145	55.4632	54.6142	70.7696	8.3804	30.0229

- HD values are lowest for the third range for all algorithms for the feature MC. Beside the smallest for algorithm *Onecut*, HD value for third range for algorithm *GCBS* is also very small.
- For feature MCR, HD values are lowest for third range for algorithms *GSC*, *GSCSeq* and *TRC*.
- For the feature MH, RMSD values, for each particular algorithm, are close for all three ranges.

5.2.3.1.2 Closed Contour Interactions Table 5.8 presents results for algorithm DRLSE which uses closed contour interactions. Some of the key points from these results are explained below:

- For three features, MI, SI and NMI, segmentation results are best for the low feature values among all three ranges in term of all three metrics Dice, RMSD and HD.
- Segmentation results in term of RMSD and HD are best for the high feature values among all three bins for the feature NSI.
- For the feature MC, best segmentation results in term of all three metrics Dice, RMSD and HD are for the high feature values.
- For the feature MCR, best value of Dice and worst values of RMSD and HD are for the high feature values among the three ranges.
- Segmentation results in term of Dice and HD are for the high feature values for the feature ME.

Table 5.8: Segmentation results for *DRLSE* algorithm that uses closed contour interactions

Feature	Range of Values	Number of Contours	Dice	RMSD	HD
Mean Intensity(MI)	0-35.3846	2535	0.60715	6.1221	18.9106
	35.3846-64.8199	749	0.56774	16.7056	43.4965
	64.8199-115.3481	110	0.12585	25.1819	123.8315
Stdev Intensity(SI)	0-6.1525	2122	0.69412	8.2596	21.256
	6.1525-12.094	988	0.41675	9.4564	33.2373
	12.094-35.1476	284	0.32931	13.8459	57.0245
NbrMeanIntensity(NMI)	0-35.6296	2464	0.60769	6.2307	19.1827
	35.6296-64.3135	813	0.5728	15.331	39.7912
	64.3135-113.4973	117	0.12962	25.5153	124.1216
NbrStdevIntensity(NSI)	0-6.5721	2169	0.68316	9.9613	25.6841
	6.5721-12.3621	923	0.38147	8.588	36.0536
	12.3621-38.0648	302	0.47793	4.2021	17.0605
Mean Contrast(MC)	0-41.9104	2348	0.57366	10.0929	31.7228
	41.9104-158.9534	947	0.6034	7.0084	19.2684
	158.9534-886.1909	99	0.6034	4.7152	14.205
Mean Correlation(MCR)	0-0.43463	570	0.57529	5.7615	18.853
	0.43463-0.83656	959	0.5781	6.0719	20.262
	0.83656-1.0005	1865	0.5876	11.6327	34.2956
Mean Energy(ME)	0-0.028037	2747	0.59629	9.4578	28.3096
	0.028037-0.068989	365	0.55884	6.5771	20.3144
	0.068989-0.28928	282	0.48299	8.5844	31.764
Mean Homogeneity(MH)	0-0.31497	1542	0.60574	7.6811	21.9902
	0.31497-0.46858	1295	0.5984	10.9698	33.4622
	0.46858-0.81314	557	0.48334	8.5311	30.3345

- For the feature MH, segmentation results in term of Dice, RMSD and HD are best for the low feature values.

5.2.3.1.3 Iso-Contour Interactions Table 5.9 shows results for algorithm *DRLSEIC* which uses closed iso-contour interactions. Some of the important observations from these results are explained below:

- Segmentation results are best in terms of RMSD and HD for the medium feature values and best in terms of Dice for the high feature values for the feature MI.
- For the feature, MC and SI, segmentation results are best in terms of all three metrics for the high feature values.
- Best Dice value is for the high feature values and best RMSD and HD values are for the low feature values for the feature NMI.
- For the feature, NSI, best Dice value is for the low feature values and best RMSD and HD values are for the high feature values.
- Dice and RMSD values are very close for all three ranges for the feature MCR.
- Segmentation results are best in term of all three metrics Dice, RMSD and HD for the low feature values for feature ME.

Table 5.9: Segmentation results for DRLSEIC algorithm that uses closed iso-contour interactions

Feature	Range of Values	Number of Contours	Dice	RMSD	HD
Mean Intensity(MI)	0-26.7439	783	0.7837	6.3126	14.0607
	26.7439-39.4167	154	0.6799	4.1431	7.6381
	39.4167-75.4812	335	0.80325	13.017	26.9442
Stdev Intensity(SI)	0-6.743	818	0.78249	8.1558	17.9547
	6.743-13.6407	356	0.74574	7.431	14.7365
	13.6407-32.1475	98	0.83542	6.3732	13.0508
NbrMeanIntensity(NMI)	0-33.6823	873	0.77373	6.2298	13.6073
	33.6823-49.2474	284	0.75713	10.8026	22.7647
	49.2474-76.0547	115	0.84301	12.478	24.9375
NbrStdevIntensity(NSI)	0-6.2592	911	0.80181	9.1232	19.8007
	6.2592-11.9467	277	0.72067	5.1565	9.8832
	11.9467-24.4362	84	0.68284	2.403	5.1906
Mean Contrast(MC)	0-47.254	927	0.78717	8.4024	17.6002
	47.254-159.425	314	0.73934	6.5164	14.809
	159.425-663.472	31	0.82506	3.4309	7.9584
Mean Correlation(MCR)	0-0.44631	191	0.76351	6.5221	13.9011
	0.44631-0.83175	348	0.76924	5.5477	11.4946
	0.83175-1	733	0.78296	9.2294	19.8594
Mean Energy(ME)	0-0.027805	1062	0.78773	7.5567	16.3985
	0.027805-0.071746	131	0.71525	8.6883	17.7959
	0.071746-0.25301	79	0.72362	9.8498	18.5533
Mean Homogeneity(MH)	0-0.31159	551	0.76563	6.803	15.0593
	0.31159-0.4656	542	0.80795	7.8838	16.9783
	0.4656-0.78875	179	0.7132	10.7263	20.7387

- For the feature MH, best Dice and best RMSD and HD values are for the medium and high feature values respectively.

5.2.3.1.4 Seed Point Interactions Table 5.10 shows results for algorithms *GCSP* and *GCnoSP* that use seed point interactions. Some of the key points from these results are stated below:

- For the feature SPI and NMI, Dice values are best for the high feature values for both the algorithms; for *GCSP* algorithm, both RMSD and HD values are best for the low feature values whereas these values are best for the medium feature values for *GCnoSP* algorithm.
- The best Dice values are for the low feature values and best RMSD and HD values are for the high feature values for both the algorithms for the feature NSI.
- The best segmentation results in term of Dice, RMSD and HD are for the high feature values for both the algorithms for the feature MC and MH.
- For the feature MCR, best RMSD and HD values are for the low feature values for both the algorithms whereas Dice values are best for the high and low feature values for algorithms *GCSP* and *GCnoSP*, respectively.
- Best Dice values are for the high feature values for both the algorithms and best RMSD and HD values are for the medium feature values for *GCSP* algorithm and for the high feature values for *GCnoSP* algorithm, respectively, for the feature ME.

Table 5.10: Segmentation results for algorithms GCSP and GCnoSP that use seed point interactions

Feature	Range of Values	Number of Contours	Dice		RMSD		HD	
			GCSP	GCnoSP	GCSP	GCnoSP	GCSP	GCnoSP
SP Intensity(SPI)	0-36	2517	0.619	0.534	10.836	17.510	23.859	36.863
	36-72	2532	0.741	0.575	11.264	11.691	25.419	25.903
	72-180	464	0.773	0.648	17.785	14.371	42.956	32.965
NbrMeanIntensity(NMI)	0-38.062	2645	0.617	0.532	10.859	18.005	23.921	37.842
	38.062-73.025	2459	0.748	0.574	11.351	10.337	25.605	23.379
	73.025-192.420	409	0.793	0.782	18.123	16.125	44.277	36.350
NbrStdevIntensity(NSI)	0-8.392	3363	0.724	0.618	12.131	17.380	27.214	37.208
	8.392-17.150	1658	0.634	0.462	10.996	13.583	25.166	28.970
	17.150-70.565	492	0.626	0.478	10.206	9.040	22.556	20.113
Mean Contrast(MC)	0-113.145	5189	0.689	0.578	11.869	16.033	26.739	34.359
	113.145-427.174	268	0.643	0.427	8.455	12.985	19.263	27.794
	427.174-1172.626	56	0.860	0.743	3.429	4.660	7.730	10.262
Mean Correlation(MCR)	0-0.473	447	0.705	0.605	7.439	10.772	17.487	24.172
	0.473-0.866	1744	0.593	0.502	10.866	14.559	23.729	31.574
	0.866-1.007	3322	0.736	0.571	12.575	17.292	28.640	36.497
Mean Energy(ME)	0-0.051	5131	0.683	0.549	11.752	15.595	26.404	33.516
	0.051-0.164	261	0.753	0.586	8.900	13.627	20.214	27.639
	0.164-0.330	121	0.767	0.741	11.775	11.778	29.652	27.530
Mean Homogeneity(MH)	0-0.325	2576	0.666	0.508	12.028	17.275	27.284	36.694
	0.325-0.498	2278	0.700	0.585	11.651	13.960	25.868	30.609
	0.498-0.830	659	0.732	0.632	9.897	12.727	22.965	26.501

In general, a lot of fine details about the segmentation performance of nine segmentation applications have been revealed through this case study of the proposed methodology. This has been possible due to categorization of interactions and use of sound statistical methods, which are two integral components of the proposed methodology. Conventional evaluation approaches would have presented the results in an aggregated form, which would have failed to discover the in-depth details of the results. For example, this evaluation methodology has provided the the opportunity to examine the relation between the feature values with the corresponding segmentation accuracies by partitioning the feature values into three ranges. As a result, we can find out, for example, which range of values for a particular feature is associated with best segmentation results in term of Dice, or which range of values for a particular feature is typically related with poor segmentation results. We also can look for the values of one or more particular features for which segmentation results are especially good for a particular SIS algorithm while overall segmentation results are poor for that algorithm. We also may be interested to know, for example, if there are any features for which segmentation results for a particular SIS algorithm is not affected at all, i.e., the segmentation results are similar irrespective of the feature values. Many other interesting details and surprising trends, concealed inside the segmentation results, can be unearthed through this methodology. Hence, this methodology of evaluation works like a data mining tool which can help to discover the underlying interesting patterns from the huge amount of segmentation results. For example, we have found that, for algorithm *GCSP*, segmentation results are best in term of Dice, RMSD and HD when values of the feature, *MC*, are in the third i.e. the highest range, which couldn't be discovered if the interactions were not categorized based on feature values. It is because a relatively small number of interactions were included in that range but corresponding mean values of Dice, RMSD and HD were in the high end, which could not be identified if the

mean were computed over the entire set of these metric values. Another example can be mentioned where we have observed that for the algorithms *GSC* and *GSCSeq*, mean RMSD values for all the ranges of the feature MH are fairly close, but for these same algorithms, mean RMSD values for all three ranges of the feature MI are significantly different, which couldn't be recognized by the existing approach of evaluation because the aggregate or overall mean RMSD values for both the features are very similar. Thus it can be concluded that this methodology can discover the underlying fine details and interesting trends hidden in the segmentation results of different SIS algorithms which could not be recognized otherwise due to the insufficient number of segmentations and consequent aggregate approach of result analysis.

5.2.4 Significance test of the results for the bins of image properties

Segmentation results are categorized according to the groups of feature values, which provides a comparative analysis about the relation of the image properties on the resulting segmentation. Several tables in the previous section presented the results based on the categories of the feature values which is actually not an indication, at least statistically, whether the observed effect was likely to be real or did not arise by chance. In order to find this impact, metric values for each of the categories based on feature values should be tested statistically. Ideally this is a hypothesis test where the null hypothesis is that the metric values for three groups of interaction come from the same distribution. This hypothesis could be tested by applying the classic one way ANOVA but as the values were tested using one-sample Kolmogorov-Smirnov test for the null hypothesis that the sample data comes from a standard normal distribution, against the alternative that it does not come from such a distribution and were found not to be normally distributed with $p = 0.0021$ with 5% significance level, a nonparametric alternative for classic one way ANOVA, the Kruskal-Wallis Test, has been chosen which doesn't require the data to be normally distributed. For each algorithm, three categories of Dice, RMSD and HD values based on the feature values are tested and corresponding p values are presented in tables, where shaded cells indicate significant p values i.e. the cells with $p < 0.05$.

Table 5.11 shows the p values of Kruskal-Wallis Test among the three groups of Dice, RMSD and HD values based on the categories of the eight feature values for the algorithms *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCBS* which uses brush stroke interaction mode.

Here, it can be observed that all the p values for the three groups of Dice, RMSD and HD values for the features MI, NMI and MCR are significant for all these five algorithms. This suggests that the three groups of Dice, RMSD and HD values for these three features for these five algorithms do not come from the same distribution i.e., these values are significantly different. This means the effects, of groups of interactions categorized based on these feature values, on segmentation results are different and are likely to be real. For the feature MC, all the p values except one are significant. MH has the least number of significant p values, which tells us that the segmentation results of these five algorithms are least affected by this feature in the region of image where the interactions are placed.

Table 5.12 shows the p values of Kruskal-Wallis Test among the three groups of Dice, RMSD and HD

Table 5.11: P values of the categories of feature values for brush stroke interaction

Algorithm	Metric	MI	SI	NMI	NSI	MC	MCR	ME	MH
GSC	Dice	1.45E-82	0.00069	3.91E-74	0.0026	6.06E-17	1.63E-26	7.56E-05	7.63E-05
	RMSD	8.77E-36	0.0324	7.04E-32	0.00016	0.00073	1.80E-27	0.0011	0.5856
	HD	2.20E-31	0.0762	1.22E-27	0.000144	6.39E-05	8.96E-26	0.00599	0.8889
GSCSeq	Dice	7.67E-64	0.00223	1.07E-54	0.000506	2.96E-17	8.19E-22	0.000587	6.37E-05
	RMSD	5.22E-30	0.0599	1.19E-27	5.75E-05	0.00443	2.75E-23	0.001	0.3958
	HD	1.95E-28	0.0506	4.07E-26	0.000176	0.00182	1.54E-21	0.0069	0.429
TRC	Dice	6.32E-14	0.1321	2.74E-12	4.35E-05	0.04604	1.96E-05	0.0992	0.01835
	RMSD	1.27E-10	0.0814	8.40E-11	0.389	0.00256	1.64E-12	0.00077	0.1168
	HD	3.03E-14	0.07435	7.11E-15	0.3579	1.69E-05	1.79E-12	0.6225	0.07426
Onecut	Dice	5.43E-94	0.04893	7.45E-81	2.89E-09	8.52E-18	2.21E-44	0.1651	2.48E-05
	RMSD	1.39E-32	0.0126	6.78E-35	0.1936	0.000403	2.01E-15	0.4421	0.4936
	HD	5.62E-07	0.00018	3.28E-07	0.2011	0.445	1.06E-07	0.06704	0.3826
GCBS	Dice	3.47E-25	0.2189	1.85E-20	0.1506	2.41E-08	2.72E-17	1.31E-05	0.5967
	RMSD	1.16E-29	0.00018	4.60E-25	7.48E-06	3.14E-11	5.53E-21	0.20932	0.7553
	HD	3.84E-31	3.62E-05	2.20E-26	2.30E-05	6.41E-09	1.17E-20	0.1187	0.9561

Table 5.12: P values of the bins of feature values for closed contour interaction

Algorithm	Metric	MI	SI	NMI	NSI	MC	MCR	ME	MH
DRLSE	Dice	2.66E-46	3.18E-167	7.08E-49	2.68E-136	0.8233	0.01346	0.000753	3.38E-09
	RMSD	6.44E-108	1.14E-07	6.11E-84	8.89E-45	8.90E-16	1.25E-32	0.10917	1.54E-15
	HD	3.62E-130	7.70E-29	3.48E-108	9.84E-33	2.54E-27	3.98E-24	0.12881	2.03E-21
DRLSEIC	Dice	4.51E-13	6.95E-09	5.33E-08	1.63E-18	1.60E-07	0.0271	0.2704	0.000119
	RMSD	3.09E-36	0.01591	8.50E-16	5.70E-39	0.0044968	4.68E-15	0.18481	0.001221
	HD	4.58E-47	5.35E-05	1.55E-17	1.69E-53	0.0012	3.40E-18	0.2547	0.00019

values based on the categories of the eight feature values for the algorithms *DRLSE* and *DRLSEIC*, which uses closed contour interaction mode.

These results of the significance test indicate that all the p values for the three groups of Dice, RMSD and HD values for the algorithms *DRLSE* and *DRLSEIC* for all the features except MC and ME are significant. This indicates that the three groups of Dice, RMSD and HD values for these two algorithms for the features MI, SI, NMI, NSI, MCR and MH do not come from the same distribution i.e., these values are significantly different. This suggests that the segmentation results of these two algorithms are likely to be affected by the values of these six features of the image region where the interactions are located. All the p values except one for the feature MC are significant, whereas only one p value for the feature ME is significant. Table 5.13 shows the p values of Kruskal-Wallis Test among the three groups of Dice, RMSD and HD values based on the categories of the seven feature values for the algorithms *GCSP* and *GCnoSP*, which uses seed point interaction mode.

This table of p values presents that all the p values are significant, which tells us that the three groups of Dice, RMSD and HD values for the algorithms *GCSP* and *GCnoSP* for all the features are significant. This is an indication that the effects of all seven features on the segmentation results are likely to be real.

So it can be concluded that for most of the algorithms, segmentation results are, in general, likely to

Table 5.13: P values of the bins of feature values for seed point interaction

Algorithm	Metric	SPI	NMI	NSI	MC	MCR	ME	MH
GCSP	Dice	6.79E-100	1.61E-104	3.03E-41	2.46E-22	2.08E-61	0.0315	0.00039
	RMSD	9.65E-74	4.30E-83	2.07E-16	2.12E-30	6.50E-51	1.13E-09	2.29E-13
	HD	5.41E-85	1.91E-95	7.46E-19	1.36E-35	9.99E-49	2.55E-11	1.41E-12
GCnoSP	Dice	3.19E-27	1.93E-55	2.44E-110	1.06E-73	4.02E-26	6.51E-05	5.46E-53
	RMSD	3.77E-40	5.85E-98	1.91E-88	8.97E-21	5.72E-15	0.00055157	3.38E-22
	HD	2.02E-43	3.20E-92	3.15E-77	4.70E-30	5.13E-13	2.84E-14	4.63E-24

be affected by some or all of the features. It is interesting to observe that the relation between the image features and segmentation results varies depending on the algorithm itself and the metric used for measuring the accuracy of segmentation.

5.3 Summary of results

This section presents the results briefly as a summary so that the entire results can be observed at a glance. These summarized results provide an overall picture to get a comparative view about the effect of different image features on the segmentation results. Figures 5.2 to 5.4 show the values of the Dice coefficient, RMSD and HD for five algorithms *GSC*, *GSCSeq*, *TRC*, *Onecut* and *GCBS* categorized into three ranges of values for eight features. It can be observed from this figure that, in general, Dice coefficients have variations for three ranges of values for almost all features for these five algorithms i.e., all these eight features have effect on the segmentation results for these five algorithms. Trends of the variations in the Dice coefficients for three ranges of feature values are different for the features i.e., for some features, Dice coefficients increase in the order of low, medium and high feature values for some algorithms, whereas for some other features, the opposite trend is observed for some algorithms and for some cases, mixed trends are observed. The extent of the variations in the values of Dice coefficients for three ranges of feature values also differ for these algorithms. Dice coefficients for algorithms *GSC*, *GSCSeq*, *TRC* and *GCBS* have variations for three ranges of values for the features MC, MI, NMI, MCR and ME. Dice coefficients for the algorithm *Onecut* vary significantly for three ranges of values for almost all features except SI.

RMSD values for the algorithms *GSC*, *GSCSeq* and *TRC* vary for three ranges of values for all features. RMSD values for the algorithm *Onecut* has slight variation for three ranges of feature values except MCR. Visible variations in the values of RMSD for three ranges of values for all features except MH are observed for the algorithm *GCBS*.

HD values for the algorithms *GSC* and *GSCSeq* have significant variations for the three ranges of values for almost all features except MH. HD values for the algorithm *TRC* varies for three ranges of feature values except NSI and ME. Small variations in the HD values for three ranges of values for all features except NSI and MH are observed for the algorithm *Onecut*. Variations in the HD values for three ranges of feature values are visible for all the features except MH for the algorithm *GCBS*.

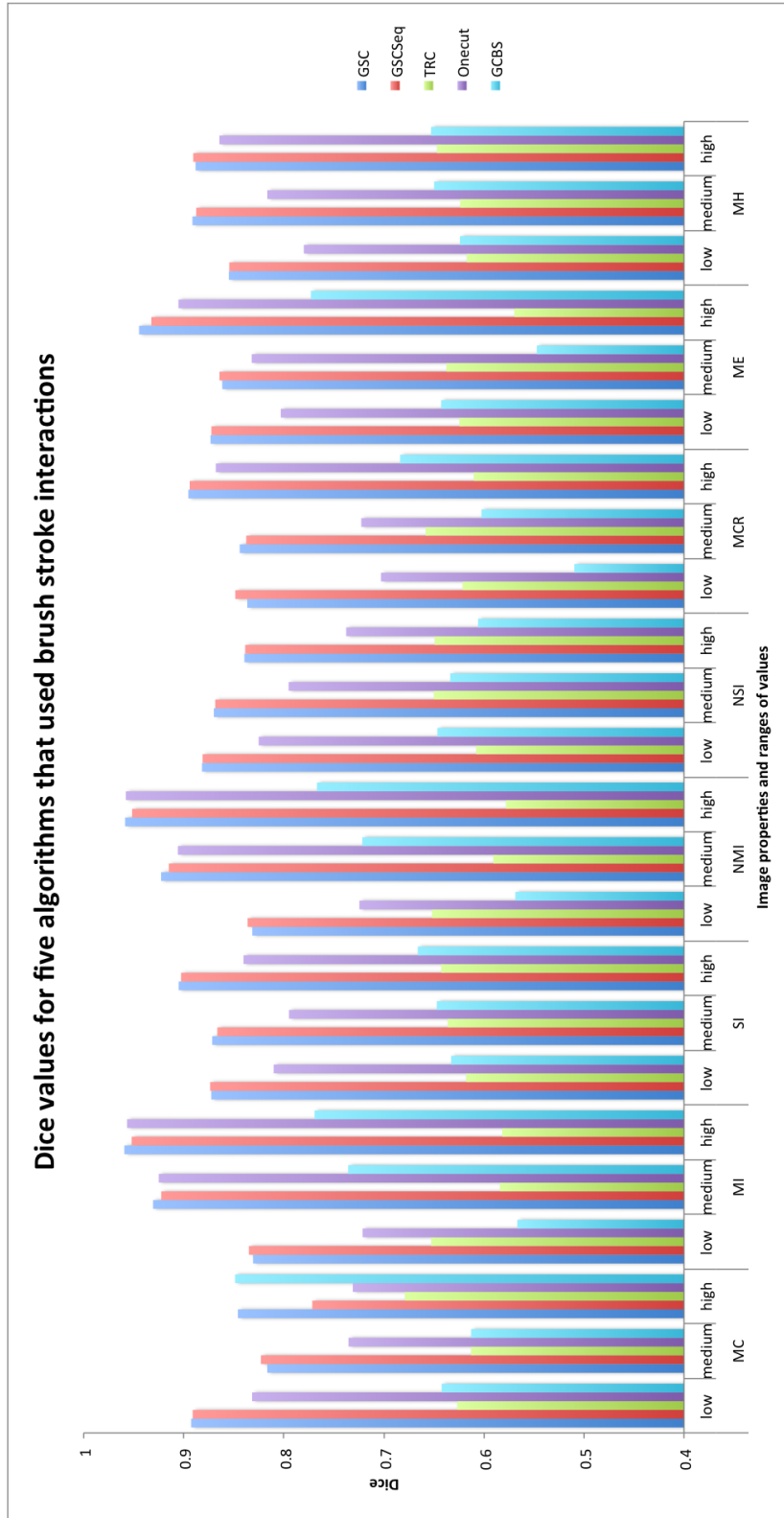


Figure 5.2: Dice values for five algorithms that used brush stroke interactions for three ranges of different feature values

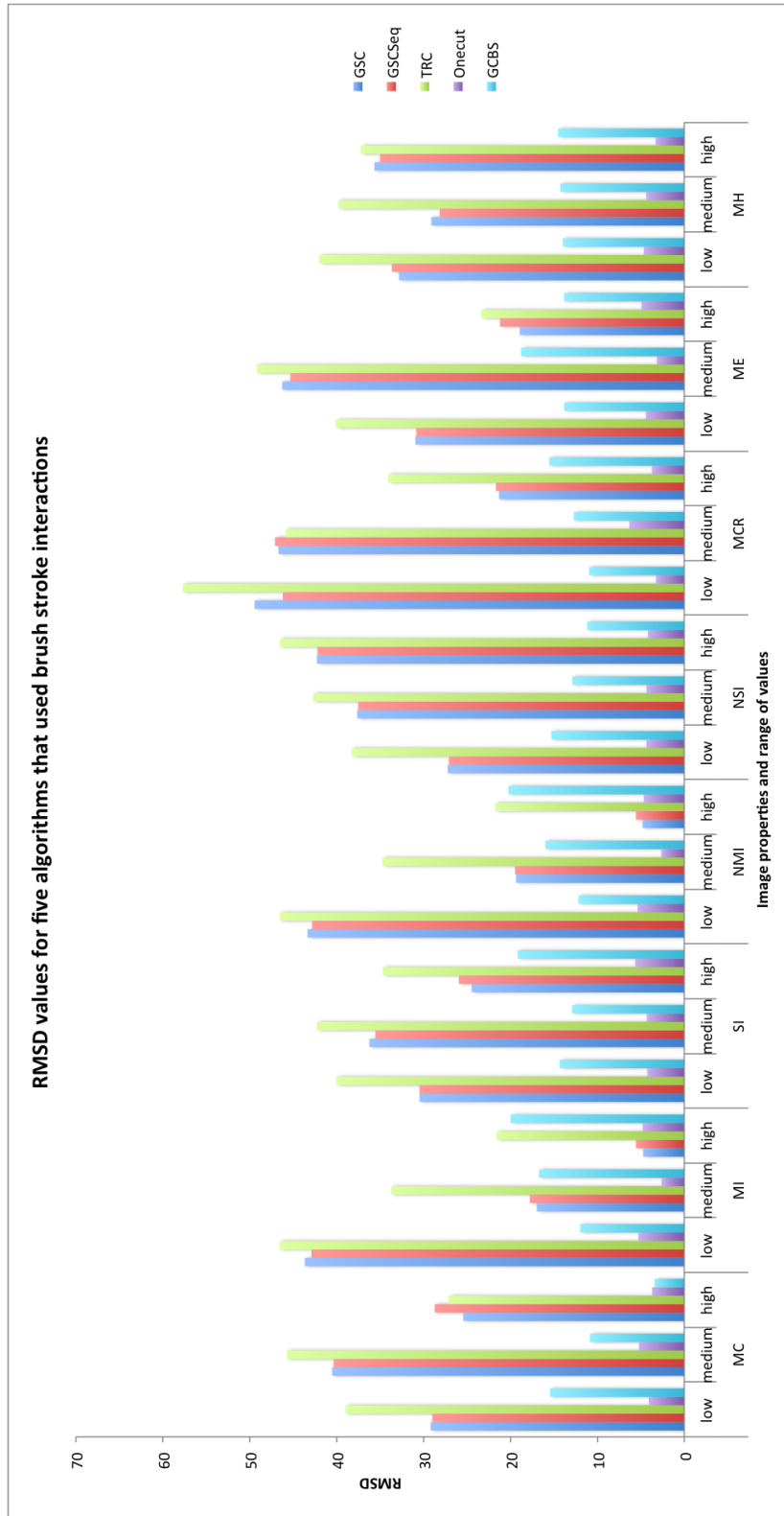


Figure 5.3: RMSD values for five algorithms that used brush stroke interactions for three ranges of different feature values

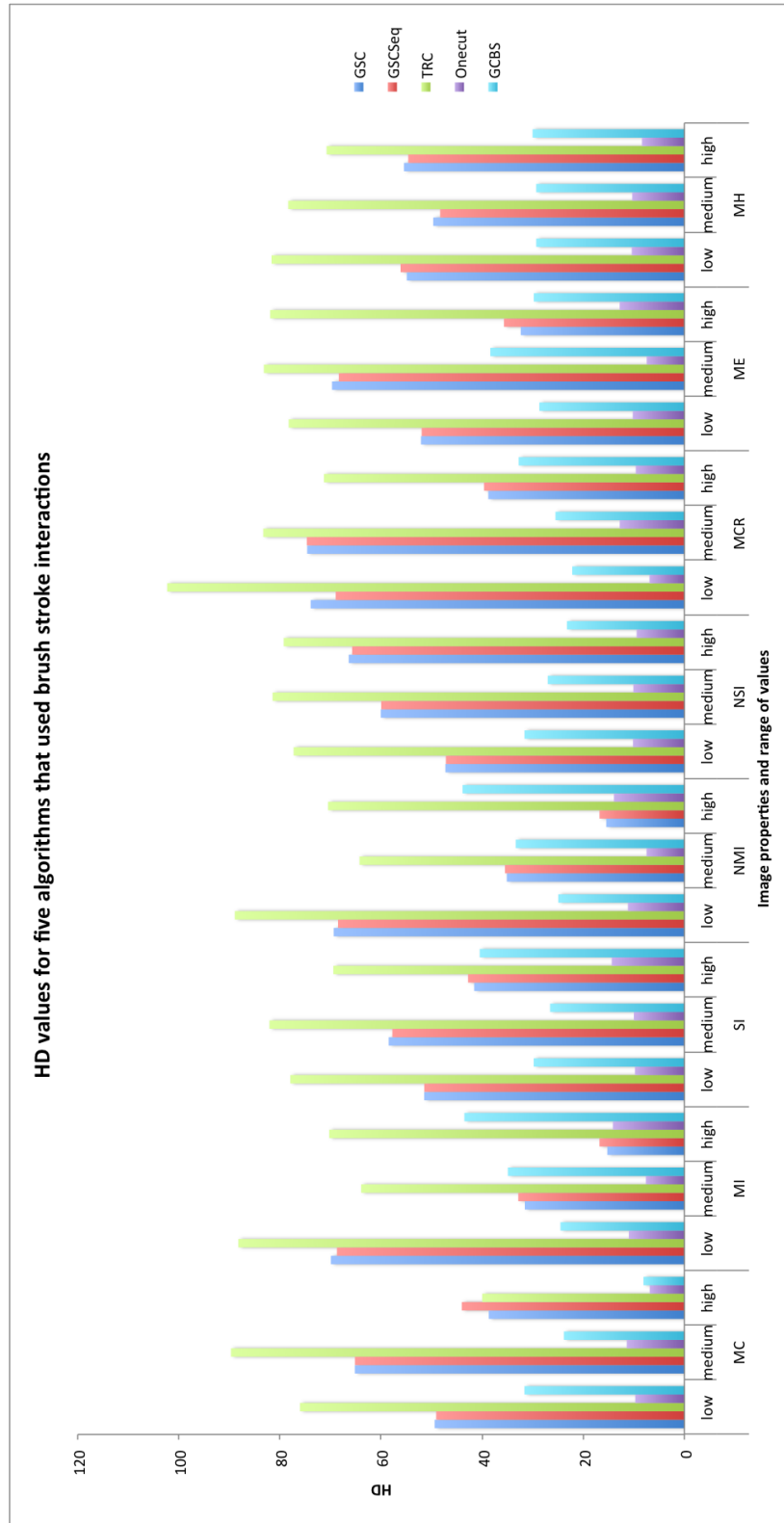


Figure 5.4: HD values for five algorithms that used brush stroke interactions for three ranges of different feature values

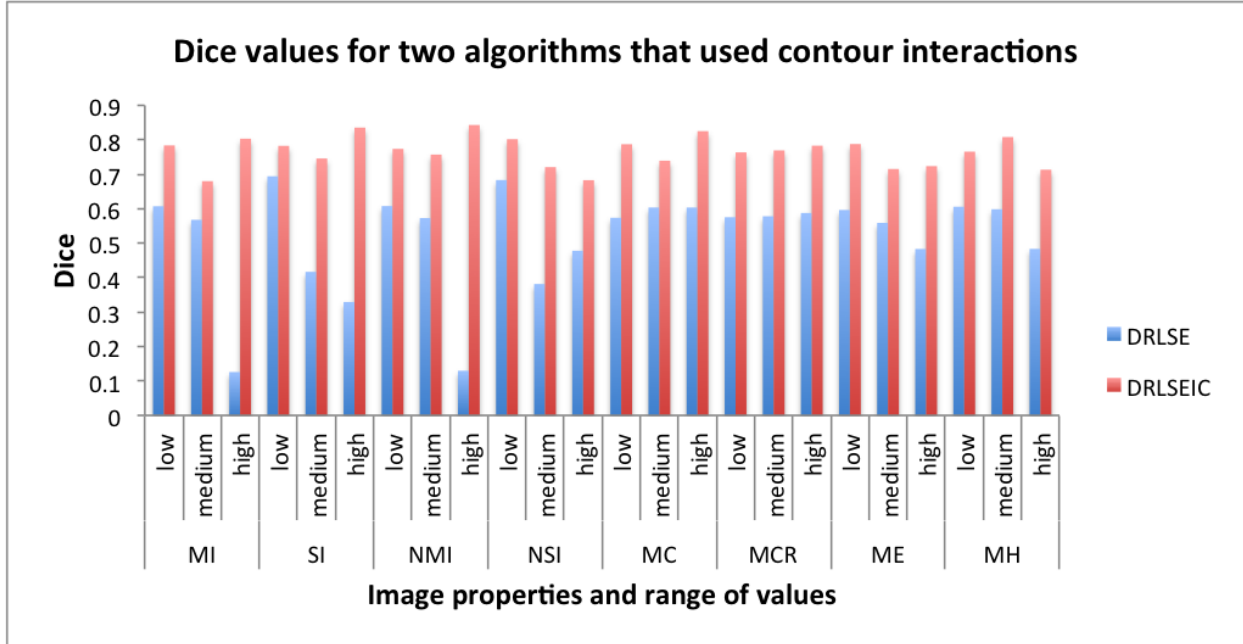


Figure 5.5: Dice values for two algorithms that used contour interactions for three ranges of different feature values

Figures 5.5 to 5.7 presents the values of Dice coefficient, RMSD and HD for two algorithms *DRLSE* and *DRLSEIC* categorized into three ranges of values for eight features. This figure shows that the Dice coefficients for three ranges of feature values varies a lot for the features MI, SI, NMI and NSI for the algorithm *DRLSE*, whereas variations for other features are very small. Variations in the Dice coefficients are very small for three ranges of values for all features for the algorithm *DRLSEIC* i.e., this algorithm is less affected by the feature values than the algorithm *DRLSE*.

Large variations in the RMSD values for three ranges of feature values are visible for all features except ME and MH for the algorithm *DRLSE*. These variations are large for all the features except SI, ME and MH for the algorithm *DRLSEIC*.

HD values for three ranges of feature values vary a lot for the features MI, NMI and SI, whereas other features have less visible effect on the HD values for the algorithm *DRLSE*. Overall variations in the HD values for three ranges of feature values for the algorithm *DRLSEIC* are small for all features except MI and NSI.

Hence, for contour interactions, image features have more visible effect on the segmentation results for the algorithm *DRLSE* than the algorithm *DRLSEIC*.

Figures 5.8 to 5.10 presents the values of Dice coefficient, RMSD and HD for two algorithms *GCSP* and *GCnoSP* categorized into three ranges of values for seven features. Slight variations in the Dice coefficients for three ranges of feature values for the features ME and MH and moderate variations for other features are observed for the algorithm *GCSP*. These variations are large for the features NMI and MC and moderate

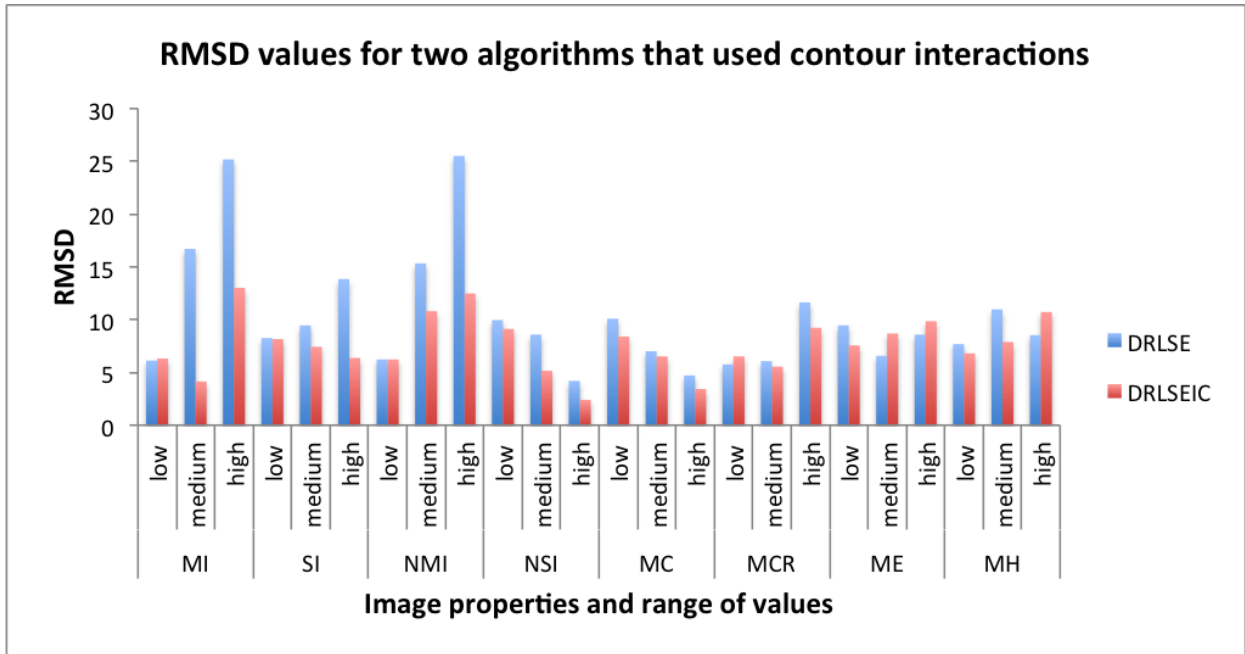


Figure 5.6: RMSD values for two algorithms that used contour interactions for three ranges of different feature values

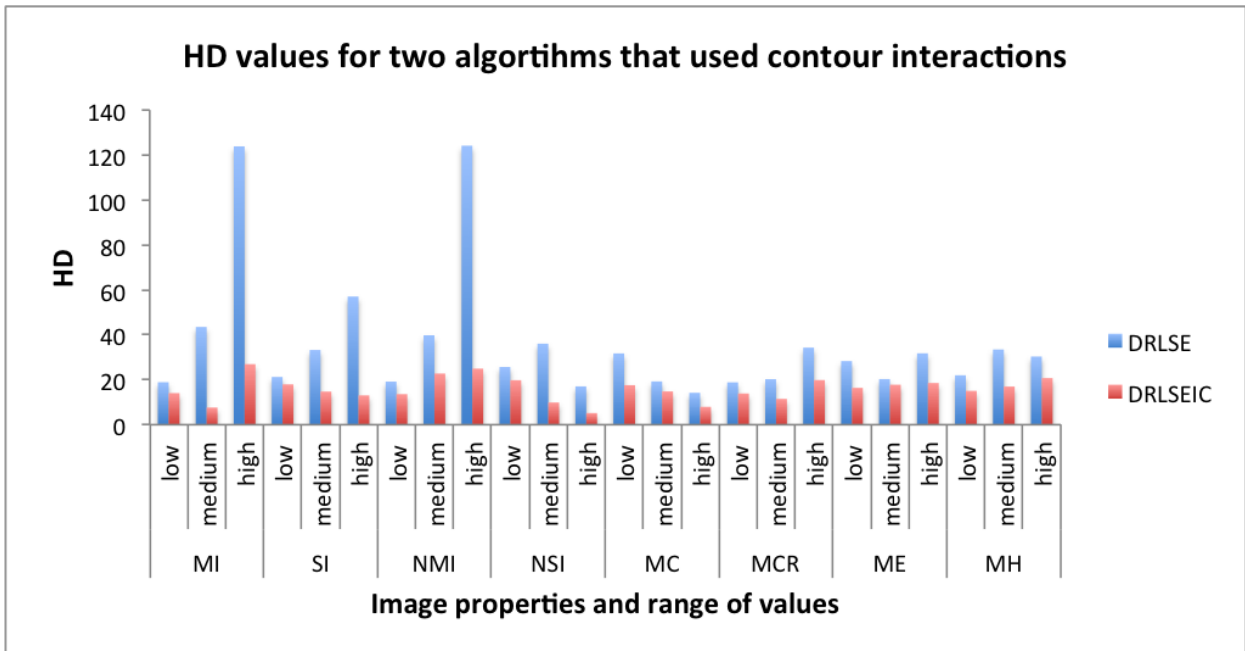


Figure 5.7: HD values for two algorithms that used contour interactions for three ranges of different feature values

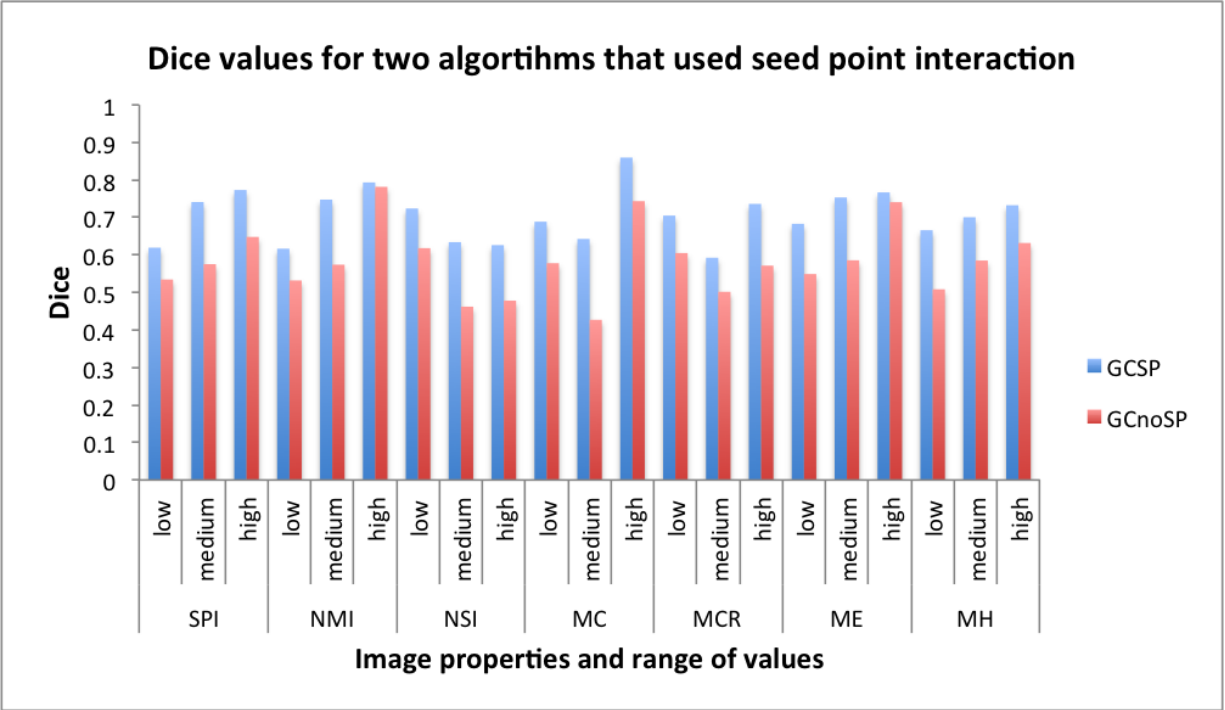


Figure 5.8: Dice values for two algorithms that used seed point interactions for three ranges of different feature values

for other features for the algorithm *GCnoSP*.

RMSD values for three ranges of feature values for the algorithm *GCSP* vary widely for the features SPI, NMI and MC, whereas moderate variations are observed for other features. These variations are large for the features NMI, MC, NSI, MCR and SPI and moderate for two other features for the algorithm *GCnoSP*.

Large variations in the HD values for three ranges of values for the features SPI, NMI and MC and moderate variations for other features are evident for the algorithm *GCSP*. HD values for the features SPI, NMI, NSI and MC vary largely for the algorithm *GCnoSP* whereas these variations are small for other features.

Considering these summarized results for seed point interactions, effects of the image features on these segmentation results are roughly similar. Overall results suggest that the extent of the effect of image features on the algorithm *GCnoSP* is slightly more than the algorithm *GCSP*.

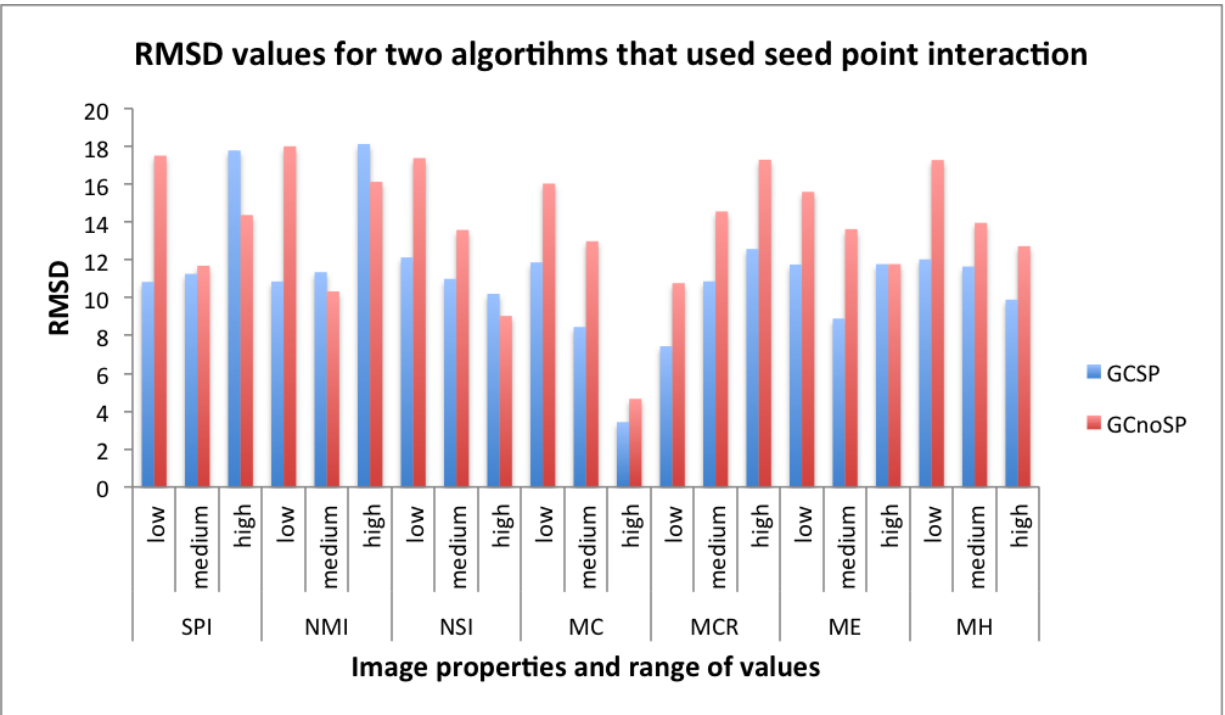


Figure 5.9: RMSD values for two algorithms that used seed point interactions for three ranges of different feature values

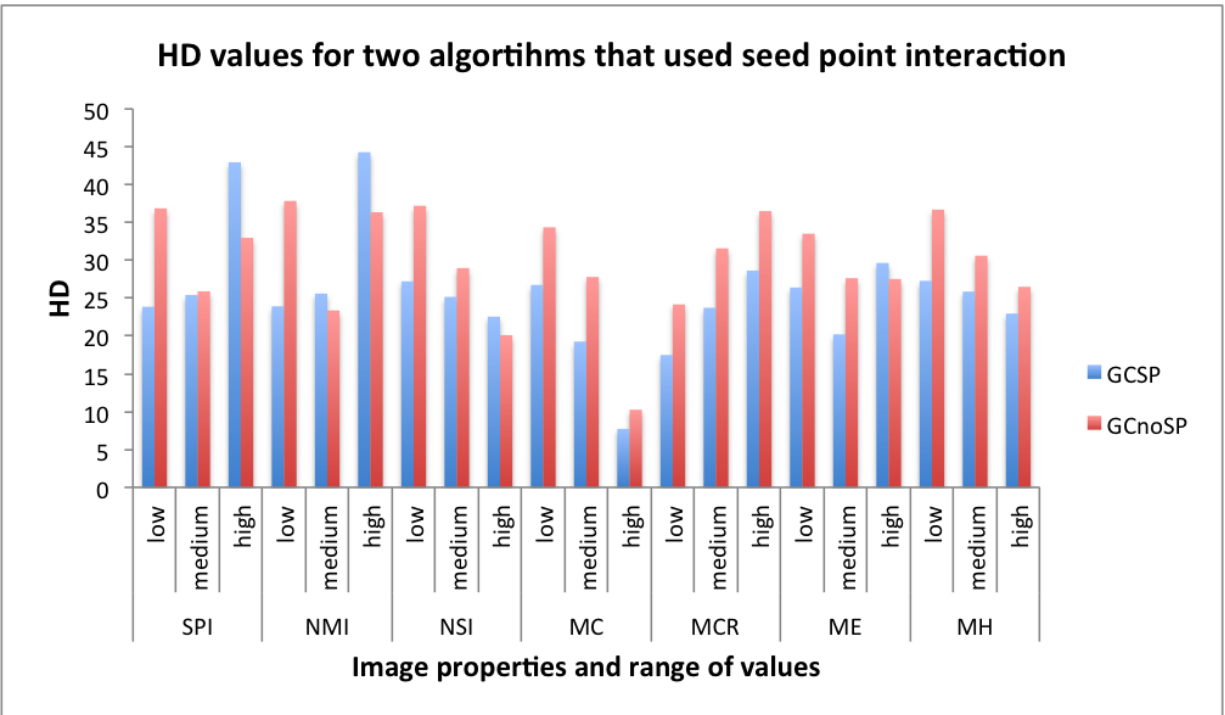


Figure 5.10: HD values for two algorithms that used seed point interactions for three ranges of different feature values

CHAPTER 6

DISCUSSION

Chapters 4 and 5 presented experimental results and analysis of two case studies to demonstrate the proposed methodology. This chapter discusses those results along with some other issues related with the methodology.

6.1 Significance of the results and observations

Chapters 4 and 5 presented, in depth, the quantitative results with analysis of the case studies. Here we consolidate those results and discuss their overall meaning. The following are the implications of the results and the observations of Chapters 4 and 5 as a whole:

- The generation of a large number of interactions across the entire object region by simulated interaction models is essential for evaluating the performance of SIS algorithms. Small numbers of interactions supplied by human users are not enough for this purpose due to the insufficient number of samples and natural inter-user variability.
- The categorization of interactions into different groups of interest is important to examine the existence of the effect of interactions on the segmentation results. In cases of any recognized impact, categorization is also essential to determine the extent of this impact. Large numbers of interactions, categorized into several groups, helps to discover subtle differences in the performances of the SIS algorithms. The categorization of interactions is not possible for the standard methods due to the small number of interactions inside the object region.
- It was shown that the spatial position of interactions can impact the segmentation results. These effects were found to be diverse depending on the object-size, interaction mode and the SIS algorithm in use. Knowing that this can occur is beneficial for human operators because they can potentially utilize such knowledge while providing interactions for an application to maximize outcomes.
- Image properties in the vicinity of interactions were also found to impact segmentation performance for some of the SIS algorithms, and these impacts were also found to be diverse depending on the specific image properties and the SIS algorithms studied. Interactions were categorized based on the image properties and variations in the results were also observed for different groups of interactions.

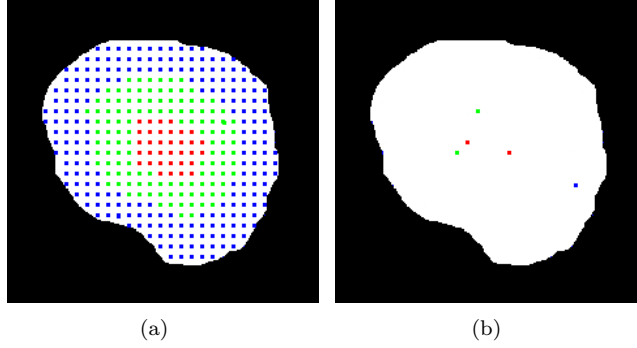


Figure 6.1: An object with seed points generated by the (a) simulated interaction models (b) human operators. Blue, green and red seed points are peripheral, intermediate and central seed points respectively.

Knowledge of these impacts can be useful to understand the effect of the image properties on the performances of SIS algorithms.

6.2 Efficacy of the methodology

This section explains the advantages of the proposed methodology over the standard methods by demonstrating the difference with the standard methods, in terms of the working principle and the corresponding outcome, using examples of three different interaction modes.

6.2.1 Seed point example

The proposed methodology uses simulated interaction models to generate large numbers of seed points to consider the segmentation results for the seed points across the entire object region to evaluate the performance of SIS algorithms. These large numbers of seed points are categorized into several interaction groups in order to verify the impact of the position of interactions on the segmentation results. But the standard methods use a small number of human operators to evaluate the segmentation performance of SIS algorithms. Moreover, interactions provided by these human operators are not free from inter-operator and intra-operator variability. Mean segmentation results computed from these small number of random and variable samples fail to represent the true picture regarding the performance of the SIS algorithm.

For this example, Figure 6.1 shows two objects, one with seed points provided by five instances of human operators (for this application one seed point is needed to segment an object) and one with seed points provided by the simulated interaction models. Table 6.1 shows the mean accuracies for these seed points overall and for the three groups of seed points. It can be observed that the accuracy values are different for the two methods and more importantly, overall mean values for the standard method are computed only from five samples whereas those values for the proposed methodology are computed using a large number of samples. Moreover, the proposed methodology can generate enough seed points for each group, whereas the

Table 6.1: Comparison between the proposed methodology and the standard method for seed point interactions.

	Proposed methodology	Standard method
Need human user?	No	Yes
Number of generated samples	Large	Very small
Categorization	Effective	Not functioning
Mean accuracy (Dice) (overall)	0.846	0.867
Mean accuracy (Dice) (central)	0.870	0.897
Mean accuracy (Dice) (intermediate)	0.861	0.881
Mean accuracy (Dice) (peripheral)	0.835	0.863

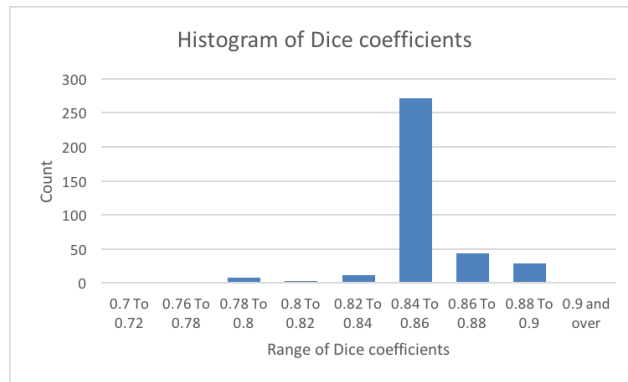


Figure 6.2: A histogram of accuracy values for the seed points generated by the simulated interaction model.

standard method has only one or two seed points. So, the accuracy values are not only different but also statistical significance is not testable for the standard methods due to the insufficient number of samples. Thus, the mean accuracy values, obtained from the standard method, do not convey the complete information regarding the performance of the algorithm. Hence, the evaluation of the performance of a SIS algorithm, by standard methods, may give an inaccurate and partial information about that algorithm. For example, a histogram, like the one in Figure 6.2, can be built from the large number of accuracy values (here it is Dice coefficients) for the proposed methodology, which can help us to get an idea about the overall distribution of the accuracy values, but there is no way that the 5 seed points in 6.1 can capture the shape or the summary statistics of that distribution in general.

Depending on the presence or absence of potentially bad inputs in the samples, standard methods may significantly underestimate or overestimate the segmentation performance, as the small number of samples are not enough to capture the variability in the interaction patterns provided by the human operators.

Standard methods could work fine if enough people are available to do the segmentations, but that is prohibitively expensive and the proposed methodology offers a more effective alternative for a tiny fraction of the financial and human effort costs.

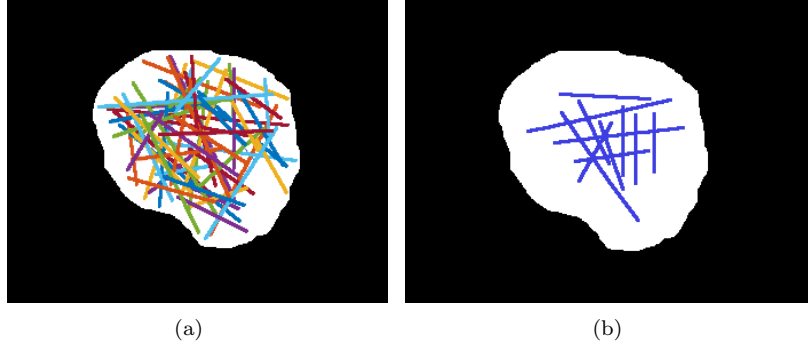


Figure 6.3: An object with brush strokes generated by the (a) simulated interaction models (b) human operators

Table 6.2: Comparison between the proposed methodology and the standard method brush stroke interactions.

	Proposed methodology	Standard method
Mean accuracy (RMSD) (overall)	17.578	15.784
Mean accuracy (RMSD) (central)	16.733	15.976
Mean accuracy (RMSD) (intermediate)	17.893	16.782
Mean accuracy (RMSD) (peripheral)	18.409	No value

6.2.2 Brush stroke example

For this example, Figure 6.3 shows two objects with brush strokes provided by human operators and by simulated interaction models. Table 6.2 shows the mean accuracies (here it is in RMSD) for these brush strokes overall and for the three groups of brush strokes. It is observed that the accuracy values are different for the two methods, and, more importantly, overall mean values for the standard methods are computed only from a few samples whereas those values for the proposed methodology are computed using a large number of samples. Moreover, the proposed methodology can generate enough brush strokes for each group whereas the standard methods have very few or even zero brush strokes for the intermediate and the peripheral regions. So, the accuracy values are not only different but also categorization is not effective for the standard methods. Thus the mean accuracy values obtained from the standard method fail to convey the complete information about the performance of the algorithm. Trends of the accuracy values, for different groups of interactions, can be determined for the proposed methodology; but for the standard method, observing trends, based on a small number of samples, is not sensible.

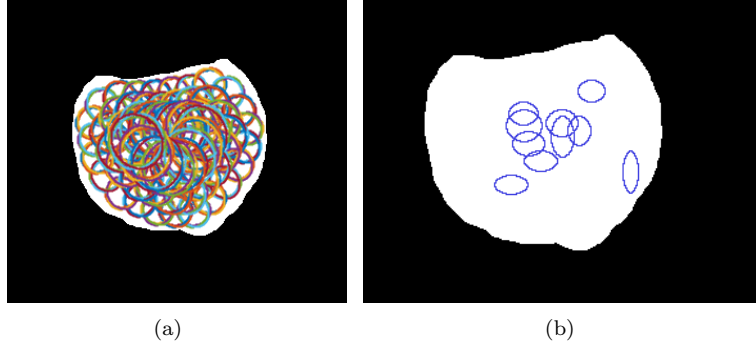


Figure 6.4: An object with closed contours generated by the (a) simulated interaction models (b) human operators

Table 6.3: Comparison between the proposed methodology and the standard method.

	Proposed methodology	Standard method
Mean accuracy (HD) (overall)	72.676	65.042
Mean accuracy (HD) (central)	46.337	46.469
Mean accuracy (HD) (intermediate)	50.691	48.973
Mean accuracy (HD) (peripheral)	97.772	90.568

6.2.3 Closed contour example

For this example, Figure 6.4 shows two objects with closed contours provided by five typical human operators and by the simulated interaction models. Table 6.3 shows the mean accuracies (here it is in HD) for these closed contours overall and for the three groups of closed contours. It can be observed that the interactions are generated across the entire object region for the proposed methodology, whereas a large area has no interaction at all for the standard method. Thus, the accuracy values for the proposed methodology are obtained by considering the interactions from all areas whereas those values for the standard method are computed based on interactions only from a few regions of the object. Hence, the proposed methodology can generate enough closed contours for each group whereas the standard methods have very few or even zero closed contours for the intermediate and the peripheral regions. So, the accuracy values are not only different but also not plausible from the viewpoint of statistical soundness due to the insufficient number of samples for the standard methods. Thus the mean accuracy values, obtained from the standard method, do not convey complete and accurate information regarding the performance of the algorithm.

6.3 Relation between accuracy and spatial position of interactions

6.3.1 Impact of position of interactions on accuracy

One of the important findings of this work is that the position of interactions has impact on the segmentation results. Although users of SIS algorithms, while providing interactions, can roughly notice these impacts by observing the variations in the results, this is not definitive. Without any formal study, it is not possible to validate and evaluate these impacts, as the impacts may be variable for different positions of interactions and trends of these variations also may not be similar. Moreover, these impacts may vary depending on the shapes and sizes of the objects, interaction modes and the algorithms in use. The proposed methodology provides a systematic and comprehensive way to consider all these factors as a whole. This work has revealed the in-depth details regarding these impacts which can help us to grasp an overall picture with respect to different algorithms and groups of interactions. This work has found enough evidence to validate the existence of the impact of position of interactions on the segmentation results. This general finding leads to the more specific question of whether this impact is same for all the interactions. The answer has been found to be “no”, because these impacts vary depending on the position of the interactions. In general, the trends of variation in the results improve as interactions move from the peripheral region towards the central region of the object, but the magnitudes of these variations are not always the same. It varies for different images and even for a single image, it varies for different objects as well. Although all the images of the dataset were ultrasonic imaging, the image properties of the images in terms of texture, brightness, noise, intensity distribution were not same across images and objects within an image. As the SIS algorithms use these image properties for segmentation, the existence of differences in the magnitudes of these variations is not surprising, but these subtle differences could not be observed for a small number of segmentations, because the variations in the image properties among the different object regions could not be significant enough to cause the differences in the segmentation results. These impacts again vary for different algorithms and interaction modes. For a particular algorithm and interaction mode, these variations with respect to the position of interactions are sometimes very subtle. This may be because there are differences in the working principles of the SIS algorithms i.e., different SIS algorithms use different number of image properties in different ways for segmentation. Moreover, different sets of pixels are collected for different interaction modes; for example, a single pixel is stored for a seed point, whereas a set of pixels are collected for a brush stroke or a closed contour. Though for each seed point, a set of neighbouring pixels are also collected, but still the sets of collected pixels for the interactions of different modes are also different, due to the differences in the forms of these interaction modes. As a result, intensity distribution or other image properties for the interactions, of different modes, may be different. Hence, the combined effect of different SIS algorithms and the interaction modes may be strong enough to produce those subtle differences in the segmentation results. This work shows that the algorithms *GCnoSP*, *GCBS* and *Onecut* are less affected by the position of interactions and the

largest impacts are visible for the algorithms *DRLSE* and *DRLSEIC*. More specifically, large variations in the values of Dice coefficients are observed for the two algorithms *DRLSE* and *DRLSEIC*, which indicate that, the segmentation results are more affected by the positions of interactions for these two algorithms. Large variations in the values of RMSD are observed for the algorithms *GSC*, *GSCSeq* and *TRC*, which are the indications that, accuracy can vary a lot based on the choice of interaction location, which is exactly same for HD values, for the same algorithms and this is why we need to sample the set of interactions densely when evaluating algorithms.

But these general trends are not exactly the same for all interactions modes i.e., for a particular algorithm, variations in the results for different interaction groups are not the same for all interaction modes. The reasons behind these differences may lie in the working principles of these algorithms, as the first three algorithms are from the same family of graphcut, which use an appearance model whereas the last two are from the family of level set algorithms which work on the evolution of the contour of the segmented region. Appearance models of grayscale images seem to be more suitable for the graphcut family of algorithms than for the level set algorithms. So, one of the main differences of this proposed methodology from the standard methods is the capability of this methodology to reveal the subtle variations in the impacts for the position of interactions in conjunction with different algorithms and interaction modes.

6.3.2 Significance of this impact

Evidence of the impact of position of interaction on the segmentation results has validated the concept that the position of interaction inside the object region is important and it has effect on the segmentation results. Having this knowledge, users now know that interactions should be placed carefully inside the object region in order to obtain the best possible segmentation. Variations in the impacts for different interaction modes and different algorithms give users a comparative view which can guide them to adapt their usual practice of providing interactions.

6.4 Relation between accuracy and image properties

6.4.1 Existence of association between image properties and accuracy

Like position of interactions, image properties also have effect on the segmentation results, although, not all image properties affect all algorithms. Several image properties (i.e., features of the image regions) were computed for all the algorithms and the corresponding segmentation results were analyzed to inspect the impact of those features on the results. In order to analyze the effect, interactions were categorized based on the feature values. Some of the features have no significant effect on the segmentation results for some of the algorithms, but some features have significant impact on the results for some of the algorithms and those impacts are diverse. Most of the features used here originated from the variations of intensity values

of the image and some of the algorithms use intensity values in different ways for segmentation. That is why the effect of some of the features on the segmentation results are very likely. As the feature values across the image regions are not homogenous, interactions have different feature values, depending on the particular regions of the image where the interactions are placed and are categorized based on these values. As the distribution of the feature values across the image regions do not follow any regular pattern, large numbers of interactions are essential to accommodate all the variabilities for proper study of the effect on the segmentation results. This is the point where the standard methods fail due to the insufficient number of randomly placed interactions. This work shows that, in general, the extent of the impacts varies a lot depending on different features for different algorithms. For example, segmentation results get better when the values of mean contrast vary from low to high. The likely explanation for this effect is that for the image regions with high contrast, foreground objects become more distinguishable from the background; so the algorithms which use the variations of intensity values for separating objects from background are likely to perform well. This is not surprising, but algorithms are not normally compared by their performance in at different contrast (or other feature) levels, which is something that our method enables. Segmentation performance, for some of the algorithms, also got better when the values of mean intensity of the neighbouring pixels (NMI) increased. This can be explained by considering that ultrasound images contain some scattered dark spots in the background, usually marked as background by the users, and the follicle regions are also blackish, and are sometimes attached with those dark spots. This makes the technique of separating the follicles from the dark spots harder, especially for the algorithms which rely on the intensity distribution for segmentation. For this, slightly brighter follicles are easier to separate from the background, resulting in better segmentations. For the feature NSI (mean standard deviation of the intensity values of neighbouring pixels), segmentation results, for some of the algorithms, got worse in the order of these feature values from low to high. The explanation for this trend may be that the larger variations of intensity values, inside the follicle area, make these objects more non-homogenous, causing the segmentation to be harder. Again, this is expected, but our method enables us to quantify and compare an algorithm's sensitivity to variance to other algorithms.

Hence, it is clear that each algorithm is affected by some of the features, but not by all the features. The reasons behind these different effects lie in the working principles of the algorithms. Different algorithms use image properties in different ways and that is why different features have different impacts on the outcomes of the algorithms. For example, some algorithms use local variations of intensity values, whereas some algorithms use intensity values in a larger scale, such as global texture features. Consequently, all the features are not equally connected with all the algorithms which eventually introduce the differences in the outcomes. From the viewpoint of the users, SIS algorithms should be designed with the minimal sensitivity to image properties, because the users do not like to pay attention to the image properties while providing interactions for an SIS application. Although it is obvious that the effects of image properties cannot be totally eliminated, efforts should be made to minimize it. Even if the effects cannot be minimized, efforts

should be directed towards minimizing algorithm sensitivity to those features that cannot easily be perceived by users. Thus the methodology provides a way to characterize the interactions in terms of image features, which creates the opportunity to categorize the interactions and, thereby, examine the impacts on the results, and compare segmentation algorithms in new ways.

6.4.2 Significance of this association

The relation of the image features to the segmentation results, discovered by the proposed methodology, can make the users more aware of how they should be placing the interactions inside the image regions. If the users are informed about the effects of the features, they can use this knowledge to provide the interactions more carefully to obtain the best possible segmentation. Diverse impacts of the features on the results for different algorithms can be used as the clues to discover the underlying basis for the segmentation performance of the algorithms. In order to understand the extent of the impact of each individual feature, an algorithm can be tested on the images having different values of each particular feature, while values of other features are kept constant, and corresponding change in the result can be observed to verify the effect and then this process should be iterated for all the features, which have an effect on that particular algorithm, according to the evaluation by the proposed methodology. Using this knowledge of the effect of each individual feature, users can choose the regions inside the object which are supposed to produce better segmentation. For this, users should be able to recognize the presence and magnitude of the features from the image visually. Features which have effects on the results but are not perceivable visually should be identified and investigation is required to understand how the working principle of the algorithm can be modified to minimize the effect of those features on the algorithm. In this way, algorithms can be redesigned and evaluated again by the methodology to assess the improvement. Iteration of this process can help to design the algorithm such that only the visually perceivable features are used, so that the users can apply that knowledge to maximize the outcome.

6.4.3 Open questions

The number and types of the image features used for this work were selected considering the image dataset in use. One can argue that some other image features could be more appropriate or the number of image features could be different, etc. This intuition points to the natural question: how can we understand that these features are the most appropriate? How do we know how many features are enough? These questions eventually lead to the more structured query: is it possible to develop a mechanism which can determine the number and kinds of image features considering the image dataset and the algorithms in use? Before addressing these issues, it should be noticed that the effect on each algorithm is the combined effect of several features, where the number of features are variable. So, the effect of each individual feature is still unknown. In order to understand this individual effect, an algorithm can be tested on the images having different values of each particular feature, while values of other features are kept constant and corresponding change in the

result can be observed to verify the effect, and then this process should be iterated for all the features. Once the individual effect is known, the working principle of an algorithm can be tested by successively excluding a feature from the set of features until the performance does not degrade. This will give us an idea about the minimum number of required features for the algorithm. This process can be iterated using different combinations of features, even including features which were not tested before. This can suggest the most appropriate set of features to maximize the performance of the algorithm. This whole procedure should be tested on different types of images to determine the required set of features for a particular type of image.

6.5 Training human operators to use SIS algorithms

6.5.1 Effect of the methodology on users

Evidence of potential impact of interaction positions and image properties on the resulting segmentations, in terms of the variations in the segmentation results for different groups of interactions, implies that the position of interaction inside the image region is important to maximize the segmentation results. The extent of these variations in the results suggests that interactions should be placed in the regions which potentially produce better segmentations instead of placing randomly inside the object region. Hence, one of the effects of this work is that, the positional and image feature sensitivities that our work elucidates can be used to inform best practices for training users to provide the interactions for maximizing the segmentation performance by choosing the potentially good regions and avoiding the bad ones.

6.5.2 Significance of this effect

The information about the interaction patterns in terms of potential capability to produce good or poor segmentation results is useful because the users can use this information for producing good segmentation results. But, on the other hand, it may be a concern for the users whether this knowledge will add any extra cognitive load on the users because users now are not free to place the interactions anywhere inside the image regions, rather they have to apply this knowledge while placing the interactions. Another concern may be whether using this knowledge will make the users slower (i.e., reduce throughput) while using an image segmentation application. A user study could be conducted to find the answers of the following specific questions:

- How much extra cognitive load will the said knowledge add on users? Increase of cognitive load could be determined from the extra time spent by the users for segmenting the same set of images using the same software for providing the interactions, applying the knowledge and without the knowledge. In order to measure the cognitive load subjectively, the most commonly used assessment tool NASA task load index (TLX) tool [56] can be used.
- Will this affect throughput? The answer to the first question would answer this question as well because

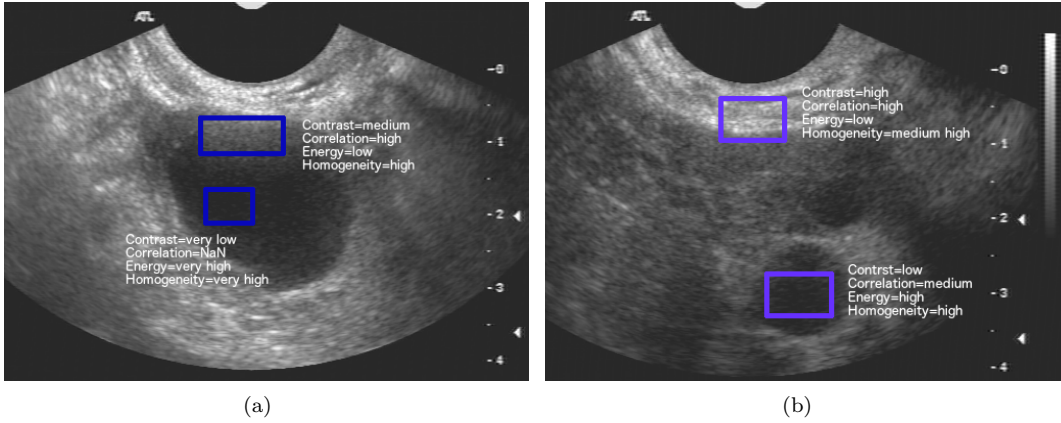


Figure 6.5: Two images with two rectangular areas, for each, showing the qualitative feature values for these areas.

the evidence of extra time spent for providing the interactions using the knowledge, may make the users slow and thereby decreasing the throughput.

- Will this be really beneficial despite the cognitive load on the users? Answer of this question could be found by considering the extra cognitive load, throughput and the gain in the accuracy. If the amount of extra load is not excessive and if the gains in the accuracy are found to be significant, that will indicate a real benefit for the users.

Depending on the findings of the user study, a trade-off may be necessary for the balance the extra workload with the gain in accuracy and the throughput. Whether the extra workload is acceptable or not can be determined by comparing the time spent for providing the interactions using and not using the knowledge. If the extra time spent is small compared to the time spent for the standard approach, then it should be acceptable. In this case, the environment, where the SIS applications are usually used should also be considered to check whether the extra time is allowable in that environment.

Another important issue here is, how to train the users to recognize the important image properties and to utilize the good areas, and avoid the bad ones while providing interactions. As previously mentioned, user training could include training in recognizing potentially good and poor areas in terms of image properties, but the problem is to recognize those areas in the image. In this regard, users will have to rely on their visual perception. So, users need to be trained from the images with the demonstration of different image properties like the following in figure 6.5.

In these two images, two rectangular areas in each image are shown with the qualitative measures of four image features for those areas, from which users can get idea about the visual perception of the image properties, and can be trained to recognize the potentially good and bad areas.

6.6 Selection of interaction modes and SIS algorithms to be used in an application

6.6.1 Choice of algorithm and interaction mode

The proposed methodology has been demonstrated by two case studies through implementation of seven different algorithms using three different interaction modes, making a total of nine segmentation applications, to assess the efficacy of the methodology. The evaluation of the relative performances of different algorithms have provided the opportunity to assess the suitability of an algorithm for a particular segmentation application. This work has also opened up the opportunity to assess whether a particular interaction mode has any effect on the performance of an algorithm, or in other words, which interaction mode is more appropriate for an algorithm.

6.6.2 Open questions

The opportunity for comparing the algorithms in terms of relative performances has also raised some questions which need to be addressed. One question may be whether it is possible to determine the best possible algorithm for a particular application and for a particular dataset? Knowing that there is evidence of a relationship between the performance of an algorithm and the interaction mode from [144], natural curiosity leads to the question: is it always possible to determine which interaction mode is best for a particular algorithm? The answer to this question can be found by evaluating the performance of an algorithm for a particular segmentation problem using different interaction modes. Even knowing this answer, it is not known whether the effects of the interaction modes on the performance of a particular algorithm are independent. So, the next relevant question may be: are there any inherent benefits to a particular interaction mode that are independent of the underlying algorithm? Knowing this answer can help to distinguish between the combined impacts of the interaction modes and the algorithms on the segmentation performance. In order to answer this question, several algorithms can be applied to a particular segmentation problem by combining several interaction modes with each of the algorithms. Then, examining the corresponding segmentation performances, it is possible to get the answer but not always guaranteed.

6.6.3 Future work

In order to determine the best possible algorithm for a particular segmentation application, commonly used SIS algorithms should be classified into several groups. Then each type of algorithm can be tested for different types of images and different types of objects to be segmented, and thus a general method can be developed to determine the best possible algorithm depending on the type of the image and the interaction mode. In the same fashion, each type of algorithm can be tested for different types of interaction modes and thus a

general technique can be devised to find out whether the performance of the algorithm differs depending on the interaction mode.

6.7 Background interactions

6.7.1 Need for automatic generation

For SIS algorithms, like foreground interactions, background interactions are also supplied by users for marking the background regions of the image. Foreground interactions were generated by the simulated interaction models but background interactions were generated manually. Attempts for automatic generation of background interactions were not successful due to the differences in the noise, texture, brightness and intensities of the pixels in the background area. The judicious choice of the locations for the placement of brush strokes in the background region is essential in order to capture the variability in the said image properties. This cannot be achieved when background interactions are generated randomly in the background area and thus, are not assured to be positioned in the areas which can ensure to capture all kinds of variability. This results in producing some very poor segmentations which are not even suitable to be included in the process of evaluation. As a result, the number of admissible segmentations turns out to be very small, or even zero, especially for very small-sized objects. According to the methodology, each image is segmented once for all the interactions generated within the foreground objects, while interactions in the background region remain constant. As the interactions in the background region are constant for a particular image, these interactions should not be placed in the areas which lead to extremely poor segmentation, thereby resulting in inadequate number of segmentations required for the evaluation process. As the automatically generated interactions can be placed anywhere within the background region, this enhances the possibility of producing extremely poor segmentation. In order to develop this methodology as a complete tool for automatic evaluation of SIS algorithms, background interactions also need to be generated automatically. Manual generation of background interactions is cumbersome and time consuming, which imposes a practical limitation on using this methodology as a tool for automatic evaluation of SIS algorithms.

6.7.2 Future work

Programmatic generation of interactions requires techniques for characterizing the background area in terms of image properties. Programmatically generated interactions, in terms of the image properties, need to be collectively equivalent to the background area. This means that the mean feature values computed from the entire background area should be very similar to that computed considering the total pixels of all the interactions. An algorithm is required to determine the number and locations of interactions in the background region of the image provided that the image properties of that interactions are equivalent to that of the background area. This could be done by generating background interactions randomly with the

following initialization:

- Number of interactions=1
- Width of the interaction=1 pixel
- Length of the interaction= smallest from a range determined from empirical study

Values of image features could be computed from the pixels of the interactions and compared to that of the background area to verify the similarity. This process could iterate by incrementing the values of these three parameters within a range and each time feature values could be compared to check for the similarity with that of the background area. Once the feature values become similar, then the iteration should stop and the generated interactions can be used as the background interactions. This technique may not look very concrete because the number of iterations may vary within a wide range and may be computationally expensive but could be a good point to start. On the whole, it is challenging but essential for the automatic generation of background interactions.

CHAPTER 7

CONCLUSION

This thesis has introduced a novel way of evaluating the performance of semiautomatic and interactive segmentation (SIS) algorithms. In this chapter we summarize our contributions and state some of the challenges and open questions in research on this topic.

7.1 Contributions

This thesis has proposed a methodology for evaluating the segmentation performance of SIS algorithms which is a significant step toward the development of a framework to automatically get an extensive evaluation of any SIS algorithm without employing human operators. This is a very significant achievement because the use of human operators to segment images for evaluating the segmentation imposes practical limitations on the number of segmentations which is not sufficient for comprehensive analysis of an algorithm's performance. In addition with this insufficient number of samples, natural inter-user variability is another major problem which makes the evaluation even more difficult and unreliable. Use of simulated interaction models to generate the interactions programmatically has made it possible to replace the human operator for providing user interactions. Use of simulated interaction models has solved the problem of an insufficient number of user interactions by generating a large number of interactions. It also has been capable to capture the variabilities in the set of user interactions by ensuring the presence of interactions everywhere inside the region of the foreground object using the method of sampling at a regular interval. The capability of the simulated interaction models to generate the interactions of all commonly used modes has made it practically useful.

Introducing the concept of categorizing the interactions according to different criteria for the purpose of in-depth analysis of the segmentation results is another contribution. This idea has provided the possibility of extensive evaluation well beyond that of conventional aggregate evaluation. Differences in the existence and extent of impact of interactions on segmentation, depending on the relative position or image properties, can be determined using the categorization of interactions. This allows characterization of the interactions into different groups and thereby relate these groups of interactions to the potential capability to produce good or poor segmentation. This information can be used to guide the users in order to avoid the bad choices of interactions for obtaining the best possible segmentation for a particular SIS algorithm.

Use of appropriate statistical methods for analysis of segmentation result is a significant step to achieve reliability in the process of evaluation because conclusions based on an analysis that lacks the proper statistical evidence may be unfounded. This methodology incorporates guidelines to evaluate the segmentation results using series of proper statistical tests that ensures the validity of the conclusion drawn based on the analysis.

Our case studies demonstrated that we were able to show that among the algorithms tested for the follicle image dataset, algorithms *Onecut*, *GCBS*, *GCnoSP* were superior, while considering both mean accuracy and consistency of the performance among the interaction categories. Algorithms *GSC* and *GSCSeq* were superior, if mean accuracy is considered more important than consistency. These results hold only for this particular application and set of algorithms; however, the methodology is such that other algorithms and interaction modes could easily be added for even richer comparison, and other datasets can be easily studied yielding similar kinds of results specific to that dataset.

7.2 Challenges

This study is a step toward the automatic extensive evaluation of SIS algorithms. Two case studies demonstrated how the proposed methodology can be applied to evaluate the segmentation performance for seven algorithms and three interaction modes for a particular type of images. Although these two case studies used ultrasound images, this methodology can be applied for any type of images provided that the image properties should be selected considering the types of the images. In order to upgrade this proposed methodology to a framework, all types of SIS algorithms and existing interaction modes should be included to make it generalized.

Two case studies here have used a particular data set containing 33 ultrasound images of ovarian follicles. Ovarian follicles have a particular type of shape and this methodology has been designed to work for a class of shape which not only includes the follicles but also a large number of objects in the real world. Still, this methodology will not work for some of the shapes, especially for the objects with non-star shape. One of the special cases for the objects, not supported by this methodology, is the objects that have one or more long thin elongated parts. Even if objects of this shape are segmented reasonably well by the SIS algorithms, still there will be problem to generate a sufficient number of interactions and categorize them, especially categorization with respect to the position of interactions.

This methodology has used several texture features for characterizing and categorizing interactions for extensive evaluation of segmentation performance. These features were suitable for the dataset used in the case study, but may not be appropriate for any type of dataset. So, a technique should be incorporated into the methodology which will select the proper number and kinds of features for a particular dataset. This can be challenging as there is no definite procedure yet to determine the perfect combination of features for any dataset.

Developing a tool for fully automatic evaluation using this methodology will face some challenges in

the sphere of practical implementation too. Especially for large datasets, generation of a large number of interactions may be computationally infeasible for practical use. Using statistical methods for result analysis also may be expensive and complicated when it will be integrated into a single application.

7.3 Open questions

Simulated interaction models are capable of generating large numbers of interactions programmatically only for foreground regions because these regions have limited area and interactions can be generated anywhere inside the regions. For background regions, manually generated interactions have been used, as programmatic generation of these interactions was not successful due to the reasons explained in the Section 3.1.1.2. Generating interactions manually for background region appears to be a cumbersome and time consuming task but not as much as it sounds to be because number of these interactions is not large and same set of interactions are used for a particular image for all instances of segmentations. Still, the obvious question is:

- Is it possible to generate the interactions for the background region automatically, provided that these will capture the variations of the image properties for the background region? If it is not possible, then how can we tackle the combinatorial problem of combining sets of background interactions with sets of foreground interactions, and having to keep the background strokes constant as a control?

In order to determine the number, size and density of interactions for all objects of interest, several heuristic formulas have been used based on the empirical studies of the particular dataset used. Determining the parameter values for these heuristic formulas, separately for each individual dataset, is not ideal as it is tedious and time consuming. So, it is natural to find out the answer to the following question:

- Is it possible to develop models for determining the values of these parameters that will work for any dataset without empirical study?

In spite of these challenges and unsolved issues, this methodology looks promising, as it can be very much beneficial for the users of image segmentation, if it can be upgraded to an application for practical use for extensive evaluation of segmentation performance of SIS algorithms.

REFERENCES

- [1] Adobe Systems Incorporation. *Adobe Photoshop User Guide*, 2002.
- [2] Shawn Andrews, Ghassan Hamarneh, and Ahmed Saad. Fast randomwalker with priors using precomputation for interactive medical image segmentation. In *Proceedings of the 13th international conference on Medical image computing and computer-assisted intervention: Part III*, pages 9–16. Springer Berlin Heidelberg, 2010.
- [3] Christopher J. Armstrong, Brian L. Price, and William A. Barrett. Interactive segmentation of image volumes with live surface. *Computers and Graphics*, 31(2):212–229, April 2007.
- [4] M. Stella Atkins and Blair T. Mackiewich. Fully automatic segmentation of the brain in mri. *IEEE Transactions on Medical Imaging*, 17(1):98–107, February 1998.
- [5] Angela R. Baerwald, Gregg P. Adams, and Roger A. Pierson. Characterization of ovarian follicular wave dynamics in women. *Biology of Reproduction*, 69(3):1023–1031, September 2003.
- [6] Xue Bai and Guillermo Sapiro. Distancecut: Interactive segmentation and matting of images and videos. In *IEEE International Conference on Image Processing*, volume 2, pages 249–252, 2007.
- [7] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 14–21 October 2007.
- [8] Xue Bai and Guillermo Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International Journal of Computer Vision*, 82(2):113–132, April 2009.
- [9] William A. Barrett and Eric N. Mortensen. Interactive live-wire boundary extraction. *Medical Image Analysis*, 1(4):331–341, 1997.
- [10] Stefan Bauer, Lutz-P. Nolte, and Mauricio Reyes. Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization. In *MICCAI 2011 Proceedings of the 14 th International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume Part III, pages 354–361, 2011.
- [11] Nir Ben-Zadok, Tammy-Raviv, and Nahum Kiryati. Interactive level set segmentation for image-guided therapy. In *IEEE International Symposium on Biomedical Imaging : From Nano to Macro*, pages 1079–1082, June 28–July 1 2009.
- [12] Wu Bingrong and Xie Mei. An interactive segmentation of medical image series. In *International Seminar on Future BioMedical Information Engineering*, pages 7–10, 2008.
- [13] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *European Conference on Computer Vision*, pages 428–441, May 2004.
- [14] Yuri Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings of the International Conference on Computer Vision (ICCV’01)*, pages 105–112, 2001.
- [15] Yuri Boykov and Gareth Funka Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

- [16] Jelena Bozek, Mario Mustra, Kresimir Delac, and Mislav Grgic. A survey of image processing algorithms in digital mammography. In Mislav Grgic, Kresimir Delac, and Mohammed Ghanbari, editors, *Recent Advances in Multimedia Signal Processing and Communications*, volume 231 of *Studies in Computational Intelligence*, pages 631–657. Springer Berlin Heidelberg, 2009.
- [17] Christopher E. Byrum, James R. MacFall, H. Cecil Charles, Venkata R. Chitilla, Orest B. Boyko, Lucy Upchurch, Jean S. Smith, Pradeep Rajagopalan, Theodore Passe, Dennis Kim, Stavra Xanthakos, K. Ranga, and R. Krishnan. Accuracy and reproducibility of brain and tissue volumes using a magnetic resonance segmentation method. *Psychiatry Research: Neuroimaging*, 67:215–234, 1996.
- [18] Stefan Cagnoni, A.B. Dobrzeniecki, R. Poli, and J.C. Yanch. Genetic algorithm-based interactive segmentation of 3d medical images. *Image and Vision Computing*, 17(12):881–895, October 1998.
- [19] Joshua E. Cates, Aaron E. Lefohn, and Ross T. Whitaker. Gist: An interactive, gpu-based level set segmentation tool for 3d medical images. *Medical Image Analysis*, 8(3):217–231, September 2004.
- [20] Vikram Chalana and Yongmin Kim. A methodology for evaluation of boundary detection algorithms on medical images. *IEEE Transactions on Medical Imaging*, 16(5):642–652, October 1997.
- [21] Tony F. Chan and Luminita A. Vese. Active contours without edges. In *IEEE Transaction on Image Processing*, volume 10, pages 266–277, February 2001.
- [22] Terrence Chen, Wei Zhang, Sara Good, Kevin S. Zhou, and Dorin Comaniciu. Automatic ovarian follicle quantification from 3d ultrasound data using global/local context with database guided segmentation. In *IEEE 12th International Conference on Computer Vision*, pages 795–802, Kyoto, Japan, 2009.
- [23] H.D. Cheng, Xiaopeng Cai, Xiaowei Chen, Liming Hu, and Xueling Lou. Computer-aided detection and classification of microcalcifications in mammograms: a survey. *Pattern Recognition*, 36(12):2967–2991, 2003.
- [24] Stephanie J. Chiu, Cynthia A. Toth, Catherirne Bowes Rickman, Joesph A. Izatt, and Sina Farsiu. Automatic segmentation of closed-contour features in ophthalmic images using graph theory and dynamic programming. *Biomedical Optics Express*, 3(5):1127–1140, May 2012.
- [25] Samiksha Chugh and S. Mahesh Anand. Semi automated tumor segmentation from mri images using local statistics based adaptive region growing. *International Journal of Information and Electronics Engineering*, 2(1):7–11, January 2012.
- [26] Benjamin A. Cohen, Irina Barash, Danny C. Kim, Matthew D. Sanger, James S. Babb, and Hersh Chandarana. Intraobserver and interobserver variability in renal volume measurements in polycystic kidney disease using a semiautomated mr segmentation algorithm. *American Journal of Roentgenology*, 199:387–393, 2012.
- [27] Laurent D. Cohen. On active contour models and ballons. *CVGIP: Image Understanding*, 53(2):211–218, 1991.
- [28] Laurent D. Cohen and Isaac Cohen. A finite element method applied to new active contour models and 3d reconstruction from cross sections. In *International Conference on Computer Vision*, pages 587–591, 4–7 December 1990.
- [29] Laurent D. Cohen and Ron Kimmel. Global minimum for active contour models. *International Journal of Computer Vision*, 24:57–78, 1997.
- [30] Luis A. Consularo, Roberto M. Cesar Jr, and Isabelle Bloch. Structural image segmentation with interactive model generation. In *IEEE International Conference on Image Processing*, volume 6, pages 45–48, 2007.
- [31] Daniel Cremers, Oliver Fluck, Mikael Rousson, and Shmuel Aharon. A probabilistic level set formulation for interactive organ segmentation. In *SPIE Proceedings on Medical Imaging*, 2007.

- [32] William R. Crum, Oscar Camara, and Derek L. Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE Transactions on Medical Imaging*, 25(11):1451–1461, November 2006.
- [33] Claudia Dach, Christian Held, Jens Wenzel, Sophia Gerlach, Roland Lang, Ralf Palmisano, and Thomas Wittenberg. Evaluation of an interactive cell segmentation for fluorescence microscopy based on the graph cut algorithm. In *Microscopic Image Analysis with Applications in Biology*, Heidelberg, Germany, September 2 2011.
- [34] Claudia Dach, Christian Held, Jens Wenzel, Sophia Gerlach, Roland Lang, Ralf Palmisano, and Thomas Wittenberg. Evaluation of an interactive cell segmentation for fluorescence microscopy based on the graph cut algorithm. In *Microscopic Image Analysis with Applications in Biology*, September 2 2011.
- [35] Erik B. Dam, Jenny Folkesson, Poala C. Pettersen, and Claus Christiansen. Semi-automatic knee cartilage segmentation. In *Proc. SPIE Medical Imaging 2006: Image Processing*, volume 6144, March 15 2006.
- [36] Piali Das, Olga Veksler, Vyachelav Zavadsky, and Yuri Boykov. Semiautomatic segmentation with compact shape prior. In *Image and Vision Computing*, volume 27, pages 206–219, January 2009.
- [37] Jeron de Bresser, Marileen P. Portegies, Alexander Leeman, Geert Jan Biessels, L. Jaap Kappelle, and Max A. Viergever. A comparison of mr based segmentation methods for measuring brain atrophy progression. *NeuroImage*, 54(2):760–768, January 2011.
- [38] Marleen de Bruijne, Bram van Ginneken, Max A. Viergever, and Wiri J. Niessen. Interactive segmentation of abdominal aortic aneurisms in cta images. *Medical Image Analysis*, 8:127–138, 2004.
- [39] Lee Raymond Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [40] Lei Ding and Alper Yilmaz. Interactive image segmentation using probabilistic hypergraphs. *Pattern Recognition*, 43(5):1863–1873, May 2010.
- [41] A.B. Dobrzeniecki and N.D. Levitt. Interactive and intuitive segmentation of volumetric data: The segmentview system and the kooshball algorithm. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 540–543, 1995.
- [42] Olivier Duchenne and Jean Yves Audibert. Segmentation by transduction. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 1–8, 23–28 June 2008.
- [43] Bovenkamp EG, Dijkstra J, Bosch JG, and Reiber JH. User-agent cooperation in multiagent ivus image segmentation. *IEEE Transactions on Medical Imaging*, 28(1):94–105, January 2009.
- [44] A. J. Einstein, J. Gil, S. Wallenstein, C. A. Bodian, M. Sanchez, D. E. Burstein, H.-S. Wu, and Z. Liu. Reproducibility and accuracy of interactive segmentation procedures for images analysis in cytology. *Journal of Microscopy*, 188:136–148, 1997.
- [45] Johannes Feulner, S. Kevin Zhou, Alexander Cavallaro, Sascha Seifert, Joachim Hornegger, and Dorin Comaniciu. Fast automatic segmentation of the esophagus from 3d ct data using a probabilistic model. In *MICCAI '09 Proceedings of the 12th International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume Part I, pages 255–262, 2009.
- [46] Gerald Friedland, Kristian Jantz, and Raul Rojas. Sioux: Simple interactive object extraction in still images. In *IEEE International Symposium on Multimedia*, 12–14 December 2005.
- [47] Jurgen Fripp, Stuart Crozier, Simon K. Warfield, and Sebastien Ourselin. Automatic segmentation and quantitative analysis of the articular cartilages from magnetic resonance images of the knee. *IEEE Transactions on Medical Imaging*, 29(1):55–64, January 2010.

- [48] Feng Ge and Song Wang. New benchmark for image segmentation evaluation. *Journal of Electronic Imaging*, 16(3), 2007.
- [49] Guido Gerig, Matthieu Jomier, and Miranda Chakos. Valmet: A new validation tool for assessing and improving 3d object segmentation. In *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '01)*, pages 516–523. Springer-Verlag London, UK, 2001.
- [50] Lena Gorelick, Olga Veksler, Yuri Boykov, and Claudia Nieuwenhuis. Convexity shape prior for segmentation. In *Computer Vision – ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 675–690. Springer International Publishing, 2014.
- [51] Hayit Greenspan, Amit Ruf, and Jacob Goldberger. Constrained gaussian mixture model framework for automatic segmentation of mr brain images. *IEEE Transactions on Medical Imaging*, 25(9):1233–1245, September 2006.
- [52] Varun Gulshan, Carsten Rother, Antonio Criministi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *Conference on Vision and Pattern Recognition*, 2010.
- [53] K. Hameeteman, M.A. Zuluaga, M. Freiman, L. Joskowicz, O. Cuisenaire, L. Flórez Valencia, M.A. Gülsün, K. Krissian, J. Mille, W.C.K. Wong, M. Orkisz, H. Tek, M. Hernández Hoyos, F. Benmansour, A.C.S. Chung, S. Rozie, M. van Gils, L. van den Borne, J. Sosna, P. Berman, N. Cohen, P.C. Douek, I. Sánchez, M. Aissat, M. Schaap, C.T. Metz, G.P. Krestin, A. van der Lugt, W.J. Niessen, and T. van Walsum. Evaluation framework for carotid bifurcation lumen segmentation and stenosis grading. *Medical Image Analysis*, 15(4):477–488, August 2011.
- [54] S.M. Rafizul Haque, Mark Eramian, and Kevin Schneider. Evaluation of interactive segmentation algorithms using densely sampled correct interactions. In *Image Analysis and Processing-ICIAP 2013*, volume 8156 of *Lecture Notes in Computer Science*, pages 191–200, Naples, Italy, 11–13 September 2013. Springer Berlin Heidelberg.
- [55] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural features for image classification. *IEEE Transaction on Systems, Man and Cybernetics*, SMC-3(6):610–621, November 1973.
- [56] Snadra G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Hancock, P.A. and Meshkati, N. (Eds.), Human Mental Workload*, pages 139–183, 1988.
- [57] Hu He, David McKinnon, Michael Warren, and Ben Upcroft. Graphcut-based interactive segmentation using colour and depth cues. In *Australasian Conference on Robotics and Automation*, Brisbane Australia, 1–3 December 2010.
- [58] Frank Heckel, Olaf Konrad, Horst Karl Hahn, and Heinz-Otto Peitgen. Interactive 3d medical image segmentation with energy-minimizing implicit functions. *Computers and Graphics*, 35(2):275–287, April 2011.
- [59] Ravindra S. Hegadi and Basavaraj A Goudannavar. Interactive segmentation of medical images using grabcut. *International Journal of Machine Intelligence*, 3(3):168–171, 2011.
- [60] Edwin Heijman, Jean-Paul Aben, Cindy Penners, Petra Niessen, Rene Guillaume, Guillaume van Eys, Klaas Nicolay, and Gustav J. Strijkers. Evaluation of manual and automatic segmentation of the mouse heart from cine mr images. *Journal of Magnetic Resonance Imaging*, 27(1):86–93, 2008.
- [61] William E. Higgins, Joseph M. Reinhardt, and Werner L. Sharp. Semiautomatic construction of 3d medical image-segmentation processes. In *SPIE Proceedings on Conference on Visualization in Biomedical Computing*, volume 2359, pages 59–71, 1994.

- [62] Kevin P. Hinshaw, Russ B. Altman, and James F. Brinkley. Shape-based models for interactive segmentation of medical images. In *Proceedings of SPIE Conference on Medical Imaging*, volume 2434, pages 771–780. SPIE, 1995.
- [63] P.S. Hiremath and Jyothi R. Tegnoor. Automatic detection of follicles in ultrasound images using active contours method. *International Journal of Service Computing and Computational Intelligence*, 1(1):26–30, 2011.
- [64] Daniel P. Huttenlocher, Gregory A. Klanderman, Gregory A. KI, and Rucklidge William J. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.
- [65] Won-Ki Jeong, Johanna Beyer, Markus Hadwiger, and Hanspeter Pfister. Scalable and interactive segmentation and visualization of neural processes in em datasets. *IEEE Transaction on Visualization and Computer Graphics*, 15(6):1505–1514, November/December 2009.
- [66] jeremie Anquez, Elsa D. Angelini, and Isabelle Bloch. Automatic segmentation of head structures on fetal mri. In *IEEE International Symposium on Biomedical Imaging : From Nano to Macro*, pages 109–112, June 28–July 1 2009.
- [67] Xiaoyi Jiang, Andree Grobe, and Kai Rothaus. Interactive segmentation of non-star-shaped contours by dynamic programming. *Pattern Recognition*, 44(9):2008–2016, September 2011.
- [68] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *Proceedings of First International Conference on Computer Vision (ICCV)*, pages 259–267, 1987.
- [69] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [70] Michael R. Kaus, Simon K. Warfield, Arya Nabavi, Peter M. Black, Ferenc A. Jolesz, and Ron Kikinis. Automated segmentation of mr images of brain tumors. *Radiology*, 218(2):586–591, 2001.
- [71] Oliver Klar. Interactive gpu-based segmentation of large medical volume data with level-sets. In *Central European Seminar on Computer Graphics*, 2007.
- [72] Stefan Klein, Uulke A. van der Heide, Irene M. Lips, Marco van Vulpen, Marius Staring, and Josien P. W. Pluim and. Med. *Medical Physics*, 35(4):1407–1417, April 2008.
- [73] Matthew A. Kupinski and Mary. Automated seeded lesion segmentation on digital mammograms. *IEEE Transactions on Medical Imaging*, 17(4):510–517, August 1998.
- [74] F. Iellamo, J.M. Legramante, G. Raimondi, F. Castrucci, M. Massaro, and G. Peruzzi. Evaluation of reproducibility of spontaneous baroreflex sensitivity at rest during laboratory tests. *Journal of Hypertension*, 14(9):1099–1104, September 1996.
- [75] Bing Nan Li, Chee Kong Chui, Stephen Chang, and Sim Heng Ong. A new unified level set method for semi-automatic liver tumor segmentation on contrast-enhanced ct images. *Expert Systems with Applications*, 39(2012):9661–9668, 2012.
- [76] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D. Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE Transaction on Image Processing*, 19(12):3243–3254, December 2010.
- [77] Hongdong Li and Chunhua Shen. Interactive color image segmentation with linear programming. *Machine Vision and Applications*, 21(4):403–412, 2010.
- [78] Hua Li, Anthony Yezzi, and Laurent D. Cohen. 3d brain segmentation using dual-front active contours with optional user interaction. *International Journal of Biomedical Imaging*, 2006:1–17, 2006.
- [79] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Transaction on Graphics*, 23(3):303–308, August 2004.

- [80] Jianming Liang, Tim McInerney, and Demetri Terzopoulos. Interactive medical image segmentation with united snakes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 116–127, 1999.
- [81] Yugang Liu and Yizhou Yu. Interactive image segmentation based on level sets of probabilities. *IEEE Transaction on Visualization and Computer Graphics*, 18(2):202–213, February 2012.
- [82] Pechin Lo, Bram van Ginneken, Joseph M. Reinhard, Ta runashree Yavarna, Pim A. de Jong, Benjamin Irving, Catalin Fetita, Margarete Ortner, Rômulo Pinho, Jan Sijbers, Marco Feuerstein, Anna Fabijanska, Christian Baue, Reinhard Beiche, Carlos S. Mendoza, Rafael Wiemker, Jaesung Lee, Anthony P. Reeves, Silvia Born, Oliver Weinheimer, Eva M. van Rikxoort, Juerg Tschirren, Ken Mori, Benjamin Odry, David P. Naidich, Ieneke Hartmann, Eric A. Hoffman, Matthias Prokop, Jesper H. Pedersen, and Marleen de Bruijne. Extraction of airways from ct (exact’09). *IEEE Transactions on Medical Imaging*, 31(11):2093–2107, November 2012.
- [83] Kongkuo Lu and William E. Higgins. Interactive segmentation based on the live wire for 3d ct chest image analysis. *International Journal of Computer Assisted Radiology and Surgery*, 2:151–167, 2007.
- [84] D. Maleike, M. Nolden, H.-P. Meinzer, and I. Wolf. Interactive segmentation framework of the medical imaging interaction toolkit. *Computer Methods and Programs in Biomedicine*, 96(1):72–83, October 2009.
- [85] Takashi Matsuyama. Expert systems for image processing-knowledge-based composition of image analysis processes. *Computer Vision, Graphics and Image Processing*, 48(1):22–49, October 1989.
- [86] Kevin McGuinness and Noel E. O’Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010.
- [87] Artem Mikheev, Gregory Nevsky, and Siddarth Govindan. Fully automatic segmentation of the brain from t1-weighted mri using bridge burner algorithm. *Journal of Magnetic Resonance Imaging*, 27(6):1235–1241, June 2008.
- [88] Akshaya Mishra, Alexander Wong, Wen Zhang, David Clausi, and Paul Fieguth. Improved interactive medical image segmentation using enhanced intelligent scissors (eis). In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE EMBC)*, 20–24 August 2008.
- [89] E. Mortensen and W. Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198, 1995.
- [90] Eric N. Mortensen and William A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384, September 1998.
- [91] Eric N. Mortensen, Bryan Morse, William A. Barrett, and Jayaram K. Udupa. Adaptive boundary detection using live-wire two-dimensional dynamic programming. In *Proceedings of Computers in Cardiology*, pages 635–638, Durham, North Carolina, 11–14 October 1992. IEEE Computer Society Press.
- [92] Emmanouil Moschidis and Jim Graham. A systematic performance evaluation of interactive image segmentation methods based on simulated user interaction. In *IEEE International Symposium on Biomedical Imaging : From Nano to Macro*, pages 928–931, 14–17 April 2010.
- [93] Emmanouil Moschidis and Jim Graham. Evaluation of a framework for on-line interactive segmentation of similar 3-d images based on a single example. In *Medical Image Understanding and Analysis*, 2011.
- [94] Emmanouil Moschidis and Jim Graham. Propagating interactive segmentation of a single 3d example to similar images: An evaluation study using mr images of the prostate. In *IEEE International Symposium on Biomedical Imaging : From Nano to Macro*, pages 1472–1475, March-April 2011.

- [95] Keelin Murphy, Bram van Ginneken, Joseph M. Reinhardt, Sven Kabus, Kai Ding, Xiang Deng, Kunlin Cao, Kaifang Du, Gary E. Christensen, Vincent Garcia, Tom Vercauteren, Nicholas Ayache, Olivier Commowick, Grégoire Malandain, Ben Glocker, Nikos Paragios, Nassir Navab, Vladlena Gorbunova, Jon Sparring, Marleen de Bruijne, Xiao Han, Mattias P. Heinrich, Julia A. Schnabel, Mark Jenkinson, Cristian Lorenz, Marc Modat, Jamie R. McClelland, Sébastien Ourselin, Sascha E. A. Muenzing, Max A. Viergever, Dante De Nigris, D. Louis Collins, Tal Arbel, Marta Peroni, Rui Li, Gregory C. Sharp, Alexander Schmidt-Richberg, Jan Ehrhardt, René Werner, Dirk Smeets, Dirk Loeckx, Gang Song, Nicholas Tustison, Brian Avants, James C. Gee, Marius Staring, Stefan Klein, Berend C. Stoel, Martin Urschler, Manuel Werlberger, Jef Vandemeulebroucke, Simon Rit, David Sarrut, and Josien P. W. Pluim. Evaluation of registration methods on thoracic ct: The empire10 challenge. *IEEE Transactions on Medical Imaging*, 30(11):1901–1920, November 2011.
- [96] Hannes Nickisch, Carsten Rother, Pushmeet Kohli, and Christoph Rhemann. Learning an interactive segmentation system. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2010)*, pages 274–281, 2010.
- [97] Jifeng Ning, Lei Zhang, David Zhang, and Chengke Wu. Interactive image segmentation by maximal similarity based region merging. *Pattern Recognition*, 43(2):445–456, February 2010.
- [98] Neil Nirkbeck, Dana Cobzas, Martin Jagersand, Albert Murtha, and Tibor Kesztyues and. An interactive graph cut method for brain tumor segmentation. In *Workshop on Computer Vision*, pages 1–7, University of Alberta, AB, Canada, 7–8 December 2009.
- [99] Alexandre Noma, Ana B.V. Graciano, Roberto M. Cesar Jr, Luis A. Consularo, and Isabelle Bloch. Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition*, 45(2012):1159–1179, 2012.
- [100] S.D. Olabarriaga and A.W.M. Smeulders. Interaction in the segmentation of medical images: A survey. *Medical Image Analysis*, 5(2):127–142, June 2001.
- [101] Nikhil R Pal and Sankar K Pal. A review on image segmentation techniques. In *Pattern Recognition*, volume 26, pages 1277–1294, 1993.
- [102] Nicolas Passat, Benoit Naegel, Francois Rousseau, Meriam Koob, and Jean-Louis Dietemann. Interactive segmentation based on component-trees. *Pattern Recognition*, 44(10–11):2539–2554, October 2011.
- [103] Sayan D. Pathak, Vikram Chalana, David R. Haynor, and Yongmin Kim. Edge-guided boundary delineation in prostate ultrasound images. *IEEE Transactions on Medical Imaging*, 19(12):1211–1219, 2000.
- [104] Bozidar Potocnik and Damjan Zazula. Automated analysis of a sequence of ovarian ultrasound images. part i: segmentation of single 2d images. *Image and Vision Computing*, 20:217–225, 2002.
- [105] Marcel Prastawa, John H. Gilmore, Weili Lin, and Guido Gerig. Automatic segmentation of mr images of the developing newborn brain. *Medical Image Analysis*, 9(5):457–466, 2005.
- [106] Alexis Protiere and Guillermo Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Transaction on Image Processing*, 16(4):1046–1057, April 2007.
- [107] Milloni R, Sbrignadello S, Tura A, Iori E, Murphy E, and Tessari P. The inter- and intra-operator variability in manual spot segmentation and its effect on spot quantitation in two-dimensional electrophoresis analysis. *Electrophoresis*, 31(10):1739–1742, May 2010.
- [108] Anjana Rajkumar, Jose Dolz, Hortense A. Kirisli, Sonja Adebahr, Tanja Schimek-Jasch, Ursula Nestle, Laurent Massoptier, Edit Varga, Pieter Jan Stappers, Wiro J. Niessen, and Yu Song. User interaction in semi-automatic segmentation of organs at risk: a case study in radiotherapy. *Journal of Digital Imaging*, 29(2):264–277, April 2016.

- [109] Nathalie Richard, Michel Dojat, and Catherine Garbay. Automated segmentation of human brain mr images using multi-agent approach. *Artificial Intelligence in Medicine*, 30(2):153–175, February 2004.
- [110] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.
- [111] Laszlo Rusko, Gyorgy Bekes, and Marta Fidrich. Automatic segmentation of the liver from multi- and single-phase contrast-enhanced ct images. *Medical Image Analysis*, 13(6):871–882, December 2009.
- [112] M. Sadeghi, G. Tien, G. Hamarneh, and M.S. Atkins. Hands-free interactive image segmentation using eyegaze. In *SPIE Proceedings on Medical Imaging 2009: Computer-Aided Diagnosis*, February 2009.
- [113] Jakob Santner, Thomas Pock, and Horst Bischof. Interactive multi-label segmentation. In *Asian conference on Computer vision*, volume 1, pages 397–410, 2010.
- [114] G.E. Sarty, W. Liang, M. Sonka, and R. A. Pierson. Semi-automated segmentation of ovarian follicular ultrasound images using a knowledge-based algorithm. *Ultrasound in Medicine and Biology*, 24(1):27–42, January 1998.
- [115] Michiel Schaap, Coert T. Metz, Theo van Walsum, Alina G. van der Giessen, Annick C. Weustink, Nico R. Mollet, Christian Bauer, Hrvoje Bogunović, Carlos Castro, Xiang Deng, Engin Dikici, Thomas O’Donnell, Michel Frenay, Ola Friman, Marcela Hernández Hoyos, Pieter H. Kitslaar, Karl Krissian, Caroline Kühnel, Miguel A. Luengo-Oroz, Maciej Orkisz, Örjan Smedby, Martin Styner, Andrzej Szymczak, Hüseyin Tek, Chunliang Wang, Simon K. Warfield, Sebastian Zambal, Yong Zhang, Gabriel P. Krestin, and Wiro J. Niessen. Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms. *Medical Image Analysis*, 13(5):701–714, October 2009.
- [116] Andrea Schenk, Guido Prause, and Heinz-Otto Peitgen. Efficient semiautomatic segmentation of 3d objects in medical images. In S. L. Delp, A. M. DiGioia, and B. Jaramaz, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2000*, volume 1935 of *Lecture Notes in Computer Science*, pages 186–195. Springer-Verlag Berlin Heidelberg, 2000.
- [117] Juan Shan. *A Fully Automatic Segmentation Method For Breast Ultrasound Images*. PhD thesis, Utah State University, Logan, Utah, 2011.
- [118] David W. Shattuck, Gautam Prasad, Mubeena Mirza, Katherine L. Narr, and Arthur W. Toga. Online resource for validation of brain segmentation methods. *NeuroImage*, 45(2):431–439, April 2009.
- [119] Dinggang Shen, Yiqiang Zhan, and C. Davatzikos. Segmentation of prostate boundaries from ultrasound images using statistical shape model. *IEEE Transactions on Medical Imaging*, 22(4):539–551, 2003.
- [120] Zhou Shoujun, Yang Jian, Wang Yongtian, and Chen Wufan. Automatic segmentation of coronary angiograms based on fuzzy inferring and probabilistic tracking. *Biomedical Engineering Online*, 9(40), 2010.
- [121] G.J. Sivewright and P.J. Elliot. Interactive region and volume growing for segmenting volumes in MR and CT images. *Medical Informatics*, 19(1):71–80, 1994.
- [122] Dirk Smeets, Dirk Loeckx, Bert Stijnene, Bart De Dobbelaer, Dirk Vandermeulen, and Paul Seutens. Semi-automatic level set segmentation of liver tumors combining a spiral-scanning technique with supervised fuzzy pixel classification. *Medical Image Analysis*, 14(1):13–20, February 2010.
- [123] Y. Song, A.J. Bulpitt, and K. Brodlie. Efficient semi-automatic segmentation for creating patient specific models for virtual environments. In *MICCAI workshop CVII*, pages 22–34, 2008.
- [124] Tobias Stammberger, Felix Eckstein, Markus Michaelis, Karl-Hans Englmeier, and Maximilian Reiser. Interobserver reproducibility of quantitative cartilage measurements: Comparison of b-spline snakes and manual segmentation. *Magnetic Resonance Imaging*, 17(7):1033–1042, 1999.

- [125] Jean Stawiaski, Etienne Decenciere, and Francois Bidault. Interactive liver tumor segmentation using graph-cuts and watershed. In *Workshop on 3D Segmentation in the Clinic: A Grand Challenge II. Liver Tumor Segmentation Challenge. MICCAI, 2008*, New York, USA., 2008.
- [126] Sebastian Steger and Georgios Sakas. Fist: Fast interactive segmentation of tumors. In Hiroyuki Yoshida, Georgios Sakas, and Marius George Linguaru, editors, *Abdominal Imaging. Computational and Clinical Applications, Third International Workshop, Held in Conjunction with MICCAI 2011, Revised Selected Papers*, volume 7029 of *Lecture Notes in Computer Science*, pages 125–132. Springer-Verlag Berlin Heidelberg, 2012.
- [127] Avan Suinesiaputra, Brett R. Cowan, Ahmed O. Al-Agamy, Mustafa A. Elattar, Nicholas Ayache, Ahmed S. Fahmy, Ayman M. Khalifa, Pau Medrano-Gracia, Marie-Pierre Jolly, Alan H. Kadish, Daniel C. Lee, Ján Margeta, Ján Margeta, Simon K. Warfield, and Alistair A. Young. A collaborative resource to build consensus for automated left ventricular segmentation of cardiac mr images. *Medical Image Analysis*, 18(1):50–62, January 2014.
- [128] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *Proceedings of “International Conference on Computer Vision ” (ICCV)*, pages 1769 –1776, Sydney, Australia, 2013.
- [129] Demetri Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438, July 1988.
- [130] Garry A. Tew, Markos Klonizakis, James Moss, Alan D. Ruddock, John M. Saxton, and Gary J. Hodges. Reproducibility of cutaneous thermal hyperaemia assessed by laser doppler flowmetry in young and older adults. *Microvascular Research*, 81(2):177–182, March 2011.
- [131] Sheila Timp and Nico Karssemeijer. A new 2d segmentation method based on dynamic programming applied to computer aided detection in mammography. *Medical Physics*, 31(5):958–971, 2004.
- [132] Shidong Tong, H. Neale Cardinal, Raymond F McLoughlin, Donal B Downey, and Aaron Fenster. Intra- and inter-observer variability and reliability of prostate volume measurement via two-dimensional and three-dimensional ultrasound imaging. *Ultrasound in Medicine and Biology*, 24(5):673–681, June 1998.
- [133] M. Tscherepanow, F. Zollner, M. Hillebrand, and F. Kummert. Automatic segmentation of unstained living cells in bright-field microscope images. In *Proceedings of the 3rd international conference on Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry*, pages 158–172, 2008.
- [134] J. K. Udupa, Wei S. Samarasekara, Y. Miki, M.A. van Buchem, and R.I. Grossman. Multiple sclerosis lesion quantification using fuzzy-connectedness principles. *IEEE Transactions on Medical Imaging*, 16(5):598–609, October 1997.
- [135] Jayaram K. Udupa, Vicki R. LeBlanc, Ying Zhuge, Celina Imielinska, Hilary Schmidt, Leanne M. Currie, Bruce E. Hirsch, and James Woodburn. A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging and Graphics*, 30(2):75–87, 2006.
- [136] Jayaram K. Udupa, Supun Samarasekara, and William A. Barrett. Boundary detection via dynamic programming. In *Proceeding of SPIE Conference on Visualization in Biomedical Computing*, volume 1808, pages 33–39, Chapel Hill, North Carolina, September 22 1992.
- [137] Bram van Ginneken, Samuel G. Armato III, Bartjan de Hoop, Saskia van Amelsvoort-van de Vorst, Thomas Duindam, Meindert Niemeijer, Keelin Murphy, Arnold Schilham, Alessandra Retico, Maria Evelina Fantacci, Niccolò Camarlinghi, Francesco Bagagli, Ilaria Gori, Takeshi Hara, Hiroshi Fujita, Hiroshi Fujita, Gianfranco Gargano, Roberto Bellotti, Sabina Tangaro, Lourdes Bolaños, Francesco De Carlo, Piergiorgio Cerello, Sorin Cristian Cheran, Ernesto Lopez Torres, and Mathias Prokop. Comparing and combining algorithms for computer-aided detection of pulmonary nodules in computed tomography scans: The anode09 study. *Medical Image Analysis*, 14(6):707–722, December 2010.

- [138] Tuomo Vehkomaki, Guido Gerig, and Gabor Szekely. A user-guided tool for efficient segmentation of medical image data. In *First Joint Conference Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery*, volume 1205 of *Lecture Notes in Computer Science*, pages 685–694, Grenoble, France, 19–22 March 1997.
- [139] Olga Veksler. Star shape prior for graph-cut image segmentation. In *European Conference on Computer Vision*, 2008.
- [140] Dan Wang, Canxiang Yan, Shiguang Shan, and Xilin chen. Active learning for interactive segmentation with expected con dence change. In *Asian conference on Computer vision 2012*, 2012.
- [141] Guotai Wang, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Tom Doel, Maria Klusmann, Anna L. David, Jan Deprest, Tom Vercauteren, and Sebastien Ourselin. Slic-seg: A minimally interactive segmentation of the placenta from sparse and motion-corrupted fetal mri in multiple views. *Medical Image Analysis*, 2016.
- [142] Jan Wassenberg, Wolfgang Middelmann, and Peter Sanders. An efficient parallel algorithm for graph-based *Computer Analysis of Images and Patterns*, 5702:1003–1010, 2009.
- [143] Andreas Wimmer, Grzegorz Soza, and Joachim Hornegger. Two-stage semi-automatic organ segmentation framework using radial basis functions and level sets. In *3D Segmentation in the Clinic - A Grand Challenge MICCAI 2007 Workshop Proceedings*, pages 179–188, Brisbane Australia, 2007.
- [144] Mingfang Wu. Effects of user strokes on image processing techniques. Master’s thesis, The University of York, 2015.
- [145] Kai Xie, Jie Yang, Z.G. Zhang, and Y.M. Zhu. Semi-automated brain tumor and edema segmentation using MRI. 56(2005):12–19, 2005.
- [146] Shengzhou Xu, Hong Liu, and Enmin Song. Marker-controlled watershed for lesion segmentation in mammograms. *Journal of Digital Imaging*, 24(5):754–763, October 2011.
- [147] Pingkum Yan, Xiaobo Zhou, Mubarak Shah, and Stephen T. C. Wong. Automatic segmentation of high-throughput rna fluorescent cellular images. *IEEE Transaction on Information Technology in Biomedicine*, 12(1):109–117, January 2008.
- [148] Jianhua Yao and David chen. Live level set: A hybrid method of livewire and level set for medical image segmentation. *Medical Physics*, 35(9):4112–4120, August 2008.
- [149] Yading Yuan, Maryellen L. Giger, Hui Li, Kenji Suzuki, and Charlene Sennett. A dual-stage method for lesion segmentation on digital mammograms. *Medical Physics*, 34(11):4180–4193, 2007.
- [150] Paul A. Yushkevich, Joesph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage*, 31(3):1116–1128, July 2006.
- [151] Yefeng Zheng, Adrian Barbu, Bogdan Georgescu, Michael Scheuring, and Dorin Comaniciu. Four-chamber heart modeling and automatic segmentation for 3-d cardiac ct volumes using marginal space learning and steerable features. *IEEE Transactions on Medical Imaging*, 27(11):1668–1683, November 2008.
- [152] Yingxuan Zhu, Samuel Cheng, and Amrit Goel. Interactive segmentation of medical images using belief propagation with level sets. In *Proceedings of the International Conference on Image Processing*, pages 4113–4116, 26–29 September 2010.