Particle Swarm Optimization and Fuzzy C-means for Domain-independent Noisy Image Segmentation

by

Saeed Mirghasemi

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Doctor of Philosophy
in Computer Science.

Victoria University of Wellington
2017

Abstract

Image segmentation is considered to be one of the foremost image analysis techniques for high-level real-world applications in computer vision. Its task is to change or simplify the representation of an image in order to make it easier to understand or analyze. Although image segmentation has been studied for many years, evolving technology and transformation of demands make image segmentation a continuing challenge.

Noise as a side effect of imaging devices is an inevitable part of images in many computer vision applications. Therefore, an important topic in image segmentation is noisy image segmentation which requires extra effort to deal with image segmentation in the presence of noise. Generally, different strategies are needed for different noisy images with different levels/types of noise. Therefore, many approaches in the literature are domain-dependent and applicable only to specific images.

A well-recognized approach in noisy image segmentation uses clustering algorithms, among which Fuzzy C-Means (FCM) is one of the most popular. FCM is unsupervised, efficient, and can deal with uncertainty and complexity of information in an image. Dealing with uncertainties is easier with the fuzzy characteristic of FCM, and complexity of information is being taken care of by utilizing different features in FCM, and also combining FCM with other techniques.

Many modifications have been introduced to FCM to deal with noisy image segmentation more effectively. Common approaches include, adding spatial information into the FCM process, addressing the FCM initialization problem, and enhancing features used for segmentation. However, existing FCM-based noisy image segmentation approaches in the literature generally suffer from three drawbacks. First, they are applicable to specific domains and images, and impotent in others. Second, they don't perform well on severely noisy image segmentation. Third, they are effective on specific type and level of noise, and they don't explore the effect of noise level variation.

Recently, evolutionary computation techniques due to their global search abilities have been used in hybridization with FCM, mostly to address FCM stagnation in local optima. Particle Swarm Optimization (PSO) is particularly of interest because of its lower computational costs, easy implementation, and fast convergence, but its potential in this area has not been fully investigated.

This thesis develops new domain-independent PSO-based algorithms for an automatic non-supervised FCM-based segmentation of severely noisy images which are capable of extracting the main coherent/homogeneous regions while preserving details and being robust to noise variation. The key approach taken in the thesis is to explore the use of PSO to manipulate and enhance local spatial and spatial-frequency information.

This thesis introduces a new PSO feature enhancement approach in wavelet domain for noisy image segmentation. This approach applies adaptive wavelet shrinkage using evaluation based on FCM clustering performance. The results show great accuracy in the case of severe noise because of the enhanced features. Also, due to adaptivity, no parameter-tuning is required according to the type or volume of noise, and the performance is consistent under noise level variation.

This thesis presents a scheme under which a fusion of two different denoising algorithms for more effective segmentation is possible. This fusion retains the advantages of each algorithm while leaving out their drawbacks. The fusion scheme uses the noisy image segmentation system introduced above and anisotropic diffusion, the edge-preserving denoising algorithm. Results show greater accuracy and stability in comparison to the individual algorithms on a variety of noisy images.

This thesis introduces another PSO-based edge-preserving adaptive wavelet shrinkage system using wavelet packets, bilateral filtering, and a detail-respecting shrinkage scheme. The analysis of the results provide a comparison between the two feature enhancement systems. The first system uses wavelets and the second uses wavelet packets as a domain to enhance features for an FCM-based noisy image segmentation. Also, the highest segmentation accuracy among all the algorithms introduced in this thesis on some benchmarks belong to this system.

List of Publications

1. S. Mirghasemi, P. Andreae, M. Zhang and R. Rayudu, ”Severely noisy image segmentation via wavelet shrinkage using PSO and Fuzzy C-Means,” 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, 2016, pp. 1-8. doi: 10.1109/SSCI.2016.7850051

2. S. Mirghasemi, R. Rayudu and M. Zhang, ”A New Modification of Fuzzy C-Means via Particle Swarm Optimization for Noisy Image Segmentation,” Proceedings of the Australasian Conference on Artificial Life and Computational Intelligence (ACALCI 2016) Canberra, Australia. 2-5 February 2016. ACM. pp. 147-159.

3. S. Mirghasemi, R. Rayudu and M. Zhang, ”A heuristic solution for noisy image segmentation using Particle Swarm Optimization and Fuzzy clustering,” 2015 7th International Joint Conference on Computational Intelligence (IJCCI), Lisbon, 2015, pp. 17-27. (**Best Student Paper Award**)

4. S. Mirghasemi, R. Rayudu and M. Zhang, ”A feature-based region growing-merging approach to color image segmentation,” 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), Wellington, 2013, pp. 376-381. doi: 10.1109/IVCNZ.2013.6727044

5. S. Mirghasemi, R. Rayudu and M. Zhang, ”A new image segmentation algorithm based on modified seeded region growing and particle swarm optimization,” 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), Wellington, 2013, pp. 382-387. doi: 10.1109/IVCNZ.2013.6727045

Acknowledgments

There are many people to thank for their part in this work. My gratitude to my supervisors:

Pondy, for encouragement, advice, and his generosity to help.

Mengjie, for guidance, responsibility, and his financial support.

Ramesh, for being helpful and patient.

My special thanks to:

Suzan hall, ECS manager, for providing me with an extra scholarship.

ECS staff member, for bing there to help me out.

My lovely office mates, Harith Al-Sahaf and Binh Ngan Tran.

And my appreciation to:

My wife for being supportive and staying by my side through all this journey.

My mom, dad, brothers, father-in-law, and mother-in-law for sending love from a long distance.

Contents

Chapter 1

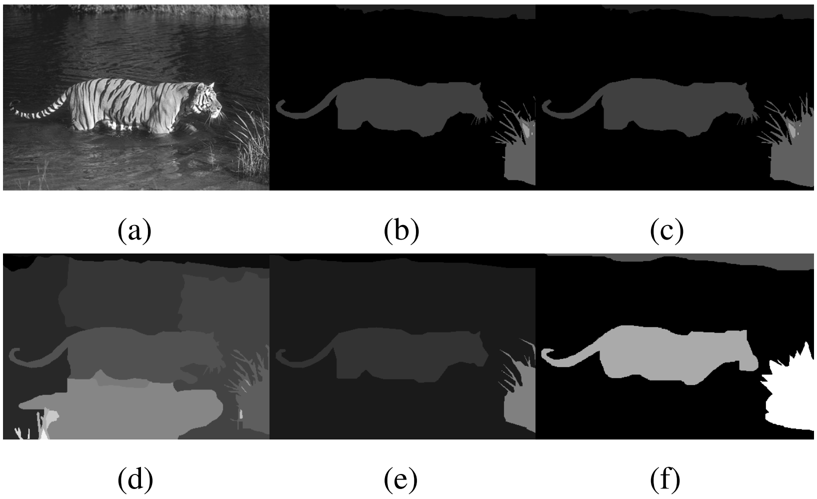
Introduction

1.1 Problem Statement

Image segmentation is generally one of the first stages in any attempt to analyze or interpret an image automatically. It bridges the gap between low-level image processing and high-level image processing. Low/mid-level image processing algorithms are usually designed to make the input image easier to process for the end application in high-level image processing. Image segmentation is a basic pre-processing step to deal with subsequent practical problems.

Image segmentation could be described at so many levels, and in many different ways. A descriptive definition is that: image segmentation is the procedure of dividing an image into its main components such as image salient regions, objects, or natural parts of objects in which the process of grouping pixels together happens between pixels that have similar attributes such as intensity, color, or texture [133]. Pure computer vision research often deals with relatively domain-independent considerations [40], and results are usable in a broad range of contexts. Thus, generally, specifying the final goal for an image segmentation algorithm is not an easy task, and is usually up to the application. Nevertheless, unsupervised image segmentation in this manner is an important part of image understanding and practical vision systems. Image segmentation can be subjective or objective depending on the final application [126]. But generally it is an objective task because the level of segmentation to which the division is carried out is dependent on the application. No unique general method could perform adequately for all applications [89]. Similarly, the segmentation procedure will stop when the regions or objects of interest have been detected. When designing a vision system, the segmentation algorithm is usually selected heuristically, and then is tuned carefully by an image processing expert with respect to application needs [89]. This means that the selected approach might not guarantee the optimal solution for a specific need while being good enough to resolve it to a great extent. Therefore, any of the generic approaches mentioned in next chapter (Section 2.1.2) can be chosen where a parameter-tuning can change the results to address the problem needs. Researchers need to use their knowledge of the domain to guide their choice. For example with very complex texture or noisy images using an edge-based algorithm is not wise, because it produces many fake and redundant edges that make the segmentation process difficult.

Image segmentation is known for being ill-defined as there are so many valid solutions as interpretations of a sample image. There is no unambiguous definition of what is/is not a region, and no objective measure exists to assess the segmentation quality [141]. Fig. 1.1 provides some segmentation results for the same image. Different results represent different regions as segments, and they are all considered acceptable depending on the application.


(a) (b) (c)

(d) (e) (f)

Figure 1.1: An image segmentation example. (a) The original image. (b-f) Several segmentation results.

Noisy image segmentation is an important topic in real world applications such as medical image processing [6, 118] and remote image analysis [120, 11]. It is among the non-trivial problems of computer vision, for it has to address two major issues at the same time. The first issue is how to denoise a noisy image to support segmentation which is not necessarily the same as simply denoising an image. The second issue is how to segment a noisy image properly considering the challenges related to segmentation itself and criteria commanded by the application. Segmentation of natural noisy images has been less considered given that segmentation is already a non-trivial task in noiseless natural images.

There is a domain-dependent noisy image segmentation field in the literature based on the famous clustering algorithm Fuzzy C-Means (FCM) [90, 127, 38, 74, 60]. This algorithm has great potential to deal with both segmentation-oriented denoising and segmentation at the same time. The fuzzy characteristics of FCM empower it to be effective in cases of noise and intensity inhomogeneities, and its clustering performance has been shown to provide reasonably good segments in some areas and applications [6, 118, 120, 11]. In FCM-based segmentation, segments are pixels inside the clusters identified by the algorithm. They represent main or high quality regions such as objects and coherent background/foreground regions. Fig. 1.2 provides a good FCM-based segmentation results for a noisy image.

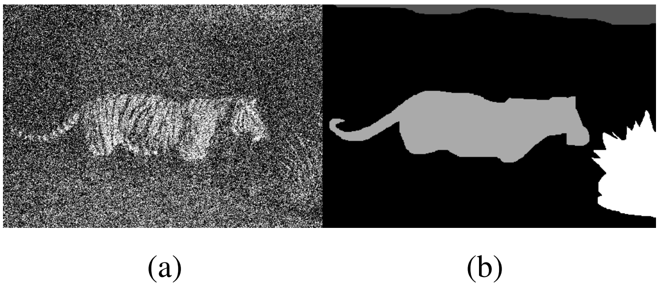


Figure 1.2: An FCM-based noisy image segmentation example. (a) The noisy image. (b-f) FCM-based segmentation result.

On the other hand, the common usage of FCM does not consider any spatial information which makes it super sensitive to noise [25]. Therefore, with a high-level of noise and other imaging artifacts, FCM loses its ability to perform image segmentation effectively. There have been many attempts in the literature to modify FCM in order to increase its abilities for noisy image segmentation, but none has been particularly satisfactory, especially for severely noisy images.

Particle Swarm Optimization (PSO), is one tool that has been utilized recently to improve FCM performance in coping with the problem of noisy image segmentation [13, 124, 145]. The PSO contribution has led to promising results but the best integration of PSO and FCM is yet to be found.

The overall goal of this thesis is to develop a new domain-independent PSO approach for an automatic non-supervised FCM-based segmentation of severely noisy images which is capable of extracting the main coherent/homogeneous regions while preserving details and being robust to noise variation. The key approach taken in the thesis is to explore the use of PSO to manipulate and enhance local spatial and spatial-frequency information.

1.2 Motivations

We have had three main motivations in mind in conducting this research.

First, although for other tasks there is considerable work dealing with severely noisy images [37, 111, 16, 31], image segmentation for severely noisy images has seldom been investigated in the literature. This puts forward the importance of introducing more effective methods for severely noisy image segmentation.

Second, the existing implementations of PSO in an FCM-based scheme are very limited, and they come along with drawbacks as we will show in the next chapter. The definitive utilization of PSO is yet to be investigated in such a paradigm. We aim at designing new schemes for an FCM-based noisy image segmentation scenario by effectively utilizing PSO.

Third, most existing algorithms in FCM-based noisy segmentation are focused on extracting local features from the spatial domain [60, 118, 38]. However, image denoising has proven to be more effective in the frequency domain. We try to demonstrate the importance of feature domain selection, and feature manipulation/enhancement on the noisy segmentation performance of FCM.

1.2.1 Challenges of Noisy Image Segmentation

There are a number of challenges related to noisy image segmentation. We briefly introduce the challenges below:

- **Preserving image texture**: texture is a complex structure of an image [65]. Homogeneous texture regions are confined within region boundaries and edges. Severe noise causes a lot of intensity fluctuation and artifacts, resulting in mixed texture regions around the boundaries. As a result, misplacement of region borders occurs, and segmentation accuracy decreases. Minimizing the mixture of homogeneous texture regions and misplacement of region boundaries is a big challenge in noisy image segmentation.

- **Preserving image details**: image details show themselves as edges, corners, and narrow/small regions that are perceptually important aspects of any image [87]. Due to the volume of noise, details could be partially/completely eliminated or over-smoothed. Preserving these details from being either occluded by noise, or deformed in the denoising process is challenging.

- **Eliminating artifacts**: noise causes fake artifacts in an image, and therefore denoising has to be intrinsically addressed in a noisy image segmentation algorithm. The denoising process has to be powerful enough to remove the artifacts to the extent that segmentation could be performed effectively.

- **Extracting main regions:** extraction of high quality regions such as compact/coherent regions or main objects in a segmentation process is a necessity for many applications such as image compression [111], object detection [30], and tracking [85]. This problem originates from the segmentation process and becomes even harder on noisy images due to existence of artifacts and intensity turbulences.

This thesis introduces new strategies to directly address detail preservation, artifact elimination, and main region extraction. Image benchmarks in this study are not texture-based images, therefore we do not directly look into texture preservation, although our strategies have good results on images with simple texture properties.

1.2.2 Why PSO?

There are reasons for exploring the use of PSO:

- **Novelty**: there have been a few studies on general-purpose FCM-based noisy image segmentation using PSO in recent years. These studies have shown that PSO has the potential to address some of the issues related to noisy image segmentation. However, a definitive and proper contribution is yet to be made. For the first time, this study uses PSO in a feature-enhancement scheme for FCM-based noisy image segmentation.

- **Representation**: PSO presentation is suitable for denoising and FCM-based image segmentation tasks where parameter tuning is of interest. PSO can be effectively used for denoising [20, 19, 143]. Also, given the nature of FCM clustering process as a continuous optimization problem, PSO can be used in different scenarios to improve its clustering performance. In PSO, particles are encoded in form of arrays in the swarm. This well matches adaptive image denoising or optimization-based clustering schemes. These schemes end up being low to medium-range dimensionality problems which is the appropriate condition for the search space for an effective PSO-based search space.

- **Simplicity**: in comparison to some other evolutionary computation methods, PSO is easier to implement, has fewer parameters to tune, and can converge faster [138].

Throughout this thesis PSO is used for enhancement of features for more effective segmentation. More specifically, PSO adaptively provides threshold values in a thresholding scheme in wavelet domain to enhance intensity features.

1.2.3 Why FCM?

FCM is one of the most widely used techniques for data clustering [130] that has been applied in many disciplines such as astronomy, geology, bioinformatics, and image analysis. In image analysis domain it has been applied to medical imaging [2], target recognition [106, 139], image segmentation [4, 106], image retrieval [76, 105], and detection tasks [96, 112]. It has been so popular in noisy image segmentation that we have a field of study as FCM-based noisy image segmentation. Below are the motivations to used FCM in this thesis to severely noisy image segmentation:

- **Practicality**: FCM is unsupervised, simple to implement, and fast. This makes the combination with other models and systems such as neural networks [113], evolutionary computation [53, 124], Markov random field [62] and support vector machine [67, 98] easy.

- **Diversity of features**: The clustering process happens based on a similarity metric between a feature vector attributed to each datapoint and cluster centers. Therefore, using a variety of features such as intensity, color, texture, and spatial, and spatial-frequency features is possible. This specification has made FCM popular in feature analysis and classifier designs [130].

- **Maneuverability**: A range of modifications and alteration can be done in the clustering process using FCM to make the final segmentation results more accurate. Different similarity metrics such as Euclidean, Mahalanobis, Manhattan, and kernel-induced could be used based on the dimensionality of the feature vector and high-level application [13, 125]. Different strategies could be thought of to modify the FCM objective function for better results [52, 129]. Also, type-2 fuzzy set (or higher-types fuzzy sets) can be utilized to deal with uncertainties present in noisy images more effectively [104].

- **Fuzzy characteristics**: The fuzzy behavior of FCM makes it robust for ambiguity, capable of retaining more information compared to hard clustering methods, and good performance in case of poor contrast, overlapping regions, noise and intensity inhomogeneity [13].

1.3 Research Questions

The research carried out in this thesis will try to answer the following main questions:

(i) *How can PSO be used in an FCM-based noisy image segmentation system for an effective enhancement of features?*

The importance of feature enhancement for denoising [93] or segmentation [110] algorithms is well established. To answer this research question, there are two main issues to be addressed. First, we need to find the type of feature(s) that has the ability to remove the artifacts caused by noise effectively. Second, a system has to be designed utilizing this feature and PSO for enhancement of segmentation results. While many of the existing algorithms are parameter dependent according to the type and volume of noise [38, 118, 25], is it possible to make the new system parameter-tuning-free from this perspective? Severely noisy image segmentation has been hardly investigated in the literature. High volume of noise makes the task extra challenging due to the high volume of artifacts created in an image. This challenge is even more serious on natural images with many descriptor details. Thereby, is the new system capable of effective segmentation while dealing with images corrupted with severe noise?

(ii) *How can details be preserved in a noisy image segmentation? How can we improve the results of the previously created system with a stress on detail preservation?*

Detail preservation as mentioned before is a serious challenge in noisy image segmentation particularly on images with fine details or severe noise. There are a number of possible ideas to address this research question. The most obvious one would be incorporating edge detection algorithms in the feature enhancement system [43, 71, 55]. The edge map produced by edge detection algorithms is a good source of information for the real location of edges. However, this incorporation in the case of severe noise is a problematic, because they produce many redundant edges causes by noise, and the proper application of edge enhancement based on edge detection is yet to be found.

Another idea to detail preservation is utilizing more effective features. Introduction [39, 44, 81] and utilization [137, 115, 3] of more effective features is more considered nowadays in this field. This leads to further questions such as whether using these features in our feature enhancement system result in a better performance.

Another intriguing idea towards retaining the details is considering edge-respectful denoising algorithms [69]. How is it possible to take advantage of the strengths of both the previous feature enhancing system and edge-respectful denoising algorithms in a way that drawbacks of the initials algorithms are addressed?

(iii) *How could a detail-preserving feature-enhancing system cope with noise type or volume variations while remaining consistently effective and parameter-tuning-*

free?

Aside from FCM-based segmentation algorithms that are parameter-dependent, edge detection and denoising algorithms are also required to be tuned for the volume or type of noise [144]. It is impossible to use denoising algorithms with a fixed parameter setting effectively in a situation that noise level variation is high. How is it possible to design a system that can predict the right parameter settings for a denoising algorithm? Does this system necessarily predicts the perfect parameter tuning for the denoising algorithm individually, or rather is it the incorporation with the feature enhancement system that produces the good results?

1.4 Research Objectives

This thesis addressee the above questions through a set of new algorithms and approaches for FCM-based noisy image segmentation. The objectives of these algorithms are as follows:

1. *Developing a new feature manipulation/enhancement system using PSO that can produce effective noisy segmentation results.* In this objective, FCM is used as its original form, and feature enhancement is the main focus. To this end, two points are taken into consideration. First, the proper choice of feature that conveys proper spatial information. Second, finding the proper feature enhancement scheme that can improve the results. Wavelet domain that represents spatial and frequency information at the same time, and also have great denoising properties is the core focus in this objective.

2. *Developing a new feature enhancement system based on the previous objective with more effort towards detail preservation.* We look at a scheme which requires not only strong noise suppression, but also avoids over-smoothing and preserve the details as much as possible. More specifically, this objectives addresses how the trade-off between a strong noise suppression system (produced in the previous objective) and another algorithm with strong detail preservation abilities should be made. Utilizing a proper secondary algorithm and an aggregation scheme for the fusion of the systems will play an important role in this objective.

3. *Developing an effective new feature manipulation system for noisy image segmentation using PSO and other richer features.* Given the previous objectives, this objective tries to apply the same strategy utilizing other types of transformations in the frequency domain. For instance, wavelet packets provide a more accurate sub-band resolution of high-frequency parts of a signal which noise mostly tends to occupy. In this objective, based on the properties of the new feature, new enhancement scheme also has to be developed.

1.5 Major Contributions

This section briefly outlines the main contributions of this thesis which corresponds to the objectives in the previous section.

1. This thesis introduces a PSO feature enhancement approach in wavelet domain for noisy image segmentation. This approach applies adaptive wavelet shrinkage using evaluation from FCM clustering performance. The PSO-based process helps to enhance intensity features for a clustering-based denoising, and also provides adaptivity for the system that can perform well on different noisy images with different noise volumes and range/spatial properties. Furthermore, the algorithm applies edge enhancement based on the Canny edge detector to the coefficients in wavelet domain. This results in even further effectiveness for noisy image segmentation and detail preservation. The algorithm shows consistent performance over a range of noise levels and different noise types without the need for parameter tuning. In addition this algorithm shows promising results on severely noisy images.

2. This thesis presents a scheme under which a fusion of two different denoising algorithms for more effective segmentation is possible. This fusion retains the advantages of each algorithm while leaving out their drawbacks. This fusion scheme uses the noisy image segmentation system introduced in the previous contribution and anisotropic diffusion, the edge-preserving denoising algorithm. The first algorithm is based on enhancing features in the wavelet domain, and the second algorithm is enhancing features in the spatial domain. After the fusion of the de-

noising results, greater accuracy and stability is observed in comparison to the individual algorithms on variety of noisy images.

3. In line with the wavelet shrinkage algorithm introduced in the first contribution, this thesis introduces another wavelet-based shrinkage system, which uses a richer feature domain: wavelet packets. This system also uses adaptive PSO-based shrinkage of wavelet coefficients. This contribution provides a comparison between feature enhancement schemes using wavelets and wavelet packets for FCM-based noisy image segmentation. In accordance with the importance of detail-preservation in noisy segmentation systems, this thesis extends the created wavelet packet-based system adding two steps. The first step is a preprocessing denoising algorithm based on bilateral filtering. Bilateral filtering as another edge-preserving denoising algorithm, paves the way for an even more effective thresholding of wavelet packets coefficients. The second step is a sigmoid-based shrinkage scheme which introduces more accuracy to the thresholding process. The results show that the extended algorithm is more effective than its predecessor, and shows the highest accuracy in comparison to some other state-of-the-art algorithms, and the algorithms introduced in this thesis so far.

1.6 Organization of the Thesis

Chapter design is as follows. Chapters 3-5 are the contribution chapters, and Chapters 2 and 6 are literature survey and conclusions respectively.

- **Chapter 2**: provides a general description of images segmentation, noise phenomena, noisy image segmentation and FCM-based noisy image segmentation. We also summarize the related work introduced in the latter to tackle the problem of FCM-based noisy image segmentation. This chapter also introduces the generic PSO and FCM algorithms as tools we have used throughout our contribution chapters

- **Chapter 3**: provides the first contribution chapter. This chapter introduces a new adaptive wavelet-based shrinkage for severely noisy image segmentation. Adaptive thresholding is fulfilled using PSO with a clustering criterion for segmentation performance evaluation. More specifically, PSO explores a search space to look for threshold values which have optimum clustering performance when applied to coefficients in the multi-scale wavelet domain. Next, the approach extended using edge enhancement on coefficients from a specific sub-band. Extensive experiments are presented in this chapter using a real and a synthetic dataset each being corrupted with with two types of common noises: Gaussian and Salt & Pepper with different levels of noise to test the consistency of the proposed methods.

- **Chapter 4**: having created two powerful FCM-based noisy image segmentation algorithms in the previous chapter, in this chapter we address one significant drawback related to them. The algorithms suffer from over-smoothing of boundaries. This chapter utilizes fusion of denoising algorithms. We use the edge-preserving Anisotropic Diffusion in spatial domain, and the adaptive wavelet thresholding algorithm in the previous chapter in frequency-spatial domain. We show how the denoising results of both the algorithms could be aggregated in wavelet domain that has better performance than both by preserving more details related to edges and boundaries. The algorithm is designed so that it is functional over the noise variation. Similar to the previous chapter, extensive experiments are carried out using the two datasets. The real dataset has got more images in this chapter, and we used Gaussian noise as the source of noise.

- **Chapter 5**: Similar to what was proposed in Chapter 3, in this chapter we introduce an adaptive shrinkage of wavelet packets coefficients. We provide analysis to show how the algorithm is different to that of Chapter 3. Next, as an extension, we add a preprocessing denoising step based on Bilateral Filtering to better functionality of wavelet shrinkage. Also, we use a different shrinkage function to preserve more details in an image. Besides the experiments presented to evaluate the new algorithms both quantitatively and qualitatively, this chapter also compares the effect of feature enhancement using wavelets and wavelet packets.

- **Chapter 6**: Conclusion to this thesis, summarizing the achievements, and suggestions to future work is discussed here.

Chapter 2

Literature Review and Methodology

2.1 Basic Concepts

2.1.1 Computer Vision

“Computer vision is the enterprise of automating and integrating a wide range of processes and representations used for vision perception” [40]. Important aspects include techniques like image processing and pattern recognition for geometric and cognitive processing of images. More specifically, it is a field that includes methods for acquiring, processing, analyzing, and understanding images in order to produce numerical or symbolic information in terms of decisions [72]. A theme in the development of this field has been to duplicate the abilities of human vision by electronically perceiving and understanding an image. Usually, the goal is to comprise a computer system that is closely modeled after the human visual system. Billions of neural cell connections within the brain form the human vision, yet human vision is fallible. Illusions and ambiguities are encountered all the time. People have a limited attention span, which makes them susceptible to distractions. People are also inconsistent. Individuals themselves often exhibit different sensitivities during the course of a day or from day to day. Similarly, there are inconsistencies from person to person, from shift to shift, and so on. The eye’s response may also be a performance limiter. Another issue associated with people includes their ability to adapt to changes. This can be either bad or good. People are flexible in that they easily move from inspecting one product to another. On the other hand, if in the course of inspecting a product produced in sheet form, the color, for example, changes slowly over during a day, depending on the subtlety of the change, it is likely that color change could go undetected. This brief reasoning makes the importance of computer vision in practical aspects of human life more clear.

2.1.2 Image Segmentation

A preliminary pure mathematical definition for image segmentation, but not accurate for this ill-defined topic could be presented as follows [59]:

Assuming R as the entire spatial region enclosed by an image. One could see image segmentation as a process of partitioning R into n subregions, $R_1, R_2, ..., R_n$ such that:

(a) $\bigcup_{i=1}^{n} R_i = R$.
(b) R_i is a connected set of pixels $i = 1, 2, ..., n$.
(c) $R_i \cap R_j = \varnothing$ for all i and j, $i \neq j$.
(d) $Q(R_i) =$ TRUE for $i = 1, 2, ..., n$.
(e) $Q(R_i \cup R_j) =$ FALSE for any adjacent regions R_i and R_j.

where, $Q(R_k)$ is a logical predicate defined over the pixels set R_k, and $\varnothing$ is the null set. The symbols $\cup$ and $\cap$ represent set union and intersection, respectively. Two regions R_i and R_j are said to be adjacent if their union forms a connected set. Condition (a) indicates that the segmentation must be complete; every pixel must be in a region. Condition (b) requires that points in a region be connected in some predefined sense (e.g. the points must be 4- or 8-connected). Condition (c) indicates that the regions must be disjoint. Condition (d) deals with the properties that must be satisfied by the pixels in a segmented region. For example, $Q(R_i)$= TRUE if all pixels in R have the same intensity level. Finally, condition (e) indicates that the two adjacent regions, R_i and R_j, must be different in the sense of the predicate Q.

Segmentation has two objectives. The first objective is to decompose the image into parts for further analysis. In simple cases, the environment might be well-controlled (in terms of noise interference, complexity of regions/background and color/gray-level distribution) so that the segmentation process reliably extracts only the parts that need to be further analyzed. An example is an algorithm which segments a human face from a color video image. The segmentation is reliable, provided that the person's clothing or room background does not have the same color components as a human face. In complex cases, such as extracting a complete road network from a grayscale aerial image, the segmentation problem can be very difficult and might require applying a great deal of domain building knowledge. The second objective of segmentation is to perform a change of representation. The pixels of the image must be organized into higher-level units that are either more meaningful or more efficient for further analysis (or both).

Based on the mathematical definition given in this chapter, generally, a good complete segmentation must satisfy the following criteria:

1. All the existing pixels in an image have to be assigned to regions.

2. There is only one region for each pixel.

3. Each region has to be a connected set of pixels.

4. Regarding the certain predicate, each region should be uniform.

5. Any merged pair of adjacent regions has to be non-uniform.

Image segmentation algorithms from a very general view can be divided into two groups *bottom-up* and *top-down*. The bottom-up approaches first divide an images into small parts such as pixels, super pixels, and small regions, and then merge the ones that represent a coherent region or object. They attempt to create uniform regions based on a predefined criteria such as intensity, color, or texture. Therefore, the results of segmentation may depend greatly on these criteria. Among this group's techniques are thresholding, region growing, and region splitting and merging. Top-down approaches use information about shape, color or texture about an object prior to segmentation. These approaches rely on class-specific information, and therefore the application is limited to images from a specific class. The intention in top-down approaches is to partition an image based on sudden changes and discontinuities. Approaches based on point, edge and contour are in this group.

A critical issue is whether or not segmentation can be performed for many different domains using general bottom-up methods that do not use any special knowledge from the domains. According to [54], "the image segmentation problem is basically one of the psychophysical perceptions, and therefore not susceptible to a purely analytical solution".

Another more detailed division which roughly covers algorithms in this filed, categories image segmentation algorithms into six main categories [131]: histogram thresholding-based methods, clustering-based methods, edge detection-based methods, region-based methods, graph-based methods, and hybrid methods.

Thresholding: Image segmentation algorithms based on thresholding are popular due to their simplicity and efficiency in some kind of images. The thresholding procedure is done on the information obtained from the histogram of the image. However, the most serious problem related to the traditional histogram-based thresholding is that they are incapable of separating different regions having the same intensity but located in different locations of an image. Using these approaches on images with unimodal histogram, especially when the target area is much smaller than the background area is not possible [132]. Although, thresholding-based approaches occasionally have been used in color image segmentation [63, 12], mostly for medical image processing, their usages is often investigated on grayscale image segmentation. Overall, thresholding-based methods do not work well in images without recognizable peaks and valleys, and cannot consider spatial relationships. Therefore, their applicability in non-trivial image datasets is not still feasible.

Clustering: The process of putting a set of pattern vectors into several subsets of similar objects based on certain homogeneity criteria is called clustering which is generally an unsupervised process. Many clustering algorithms have been proposed [68] with the ability to be applied in image segmentation. The Clustering methods usually treat an image as a set of multi-dimensional data. Usually, over segmentation is a problem in this category which is needed to be taken care of. Also, Feature selection is critically important in final segmentation results. Two commonly used algorithms in this category are k-means [84] and fuzzy c-means [17]. K-means partitions a set of data into K clusters, with variable cluster centers in an iterative procedure. The goal is to find cluster centers in a way that the sum of the distances between all the members of clusters and their centers is minimum. This is fulfilled via an objective function. The final segmentation results depend on the number of clusters, and the position of cluster centers. Although, the algorithm guaranteed to converge, it may not give the optimum solution. On the other hand, in fuzzy clustering (also known as soft clustering), each datapoint has a degree of membership to all clusters rather than belonging solely to one cluster. The procedure of clustering is very similar to k-means. The aim is to minimize an objective function based on the mentioned distance and the degree of membership of each datapoint. Color, intensity, texture, and location could be used as features for image segmentation using clustering methods.

Edge detection: In this category, the idea is to detect edges in an image using one of the existing methods. The popular edge detection methods are Sobel, Robert, Prewitt, Laplacian, and Canny edge detectors [28, 109, 108]. They all return a binary image as the output in which edges have been detected. This binary image is then used to identify the region boundaries. However, serious problems are related to these methods. It is not easily possible to detect region boundaries in the form of a closed curve. Also, in noisy images, or images with too many edges, the edge detection algorithms are not working properly. They either produce a lot of extra edges, or miss the real edges [131].

Graph-based methods: In this category there is a fully connected graph in which nodes represent pixels, and there is a link between every pair of pixels. Also, there is a similarity value for each link. The idea is to cut the graph into segments representing different regions. To do so, the links with the lowest similarity are deleted. The interpretation of an image as a graph, whose arcs are defined by some adjacency relations, provides different topologies to exploit optimum connectivity between pixels for effective delineation. Two well-known main approaches in this field are normalized cut [114], and graph cut-based methods [23, 24, 33]. Among disadvantages related to normalized cut, one could mention needing to chose number of segments, high storage requirement, time complexity, and being biased towards partitioning into equal segments.

Region Growing: Region-based techniques include region growing, region splitting, region merging, and others. As a bottom-up technique, the region growing algorithms, group pixels into homogeneous regions. In an iterative procedure neighboring pixels are grouped together based on a defined predicate. This happens when a pixel satisfies this homogeneity criterion enough. After it was included in the region, the region attributes such as mean color/intensity and area have to be updated. The classic way to perform region growing is the Seeded Region Growing (SRG) in which the procedure starts with selecting a set of seeds as the initial members of regions, and then other pixels will be added to regions gradually. Several approaches have been proposed in this domain with different viewpoints on similarity criteria, and selection of seeds' locations. The main advantages of utilizing region growing methods rely on color similarity, and pixel proximity in the image [107].

Region growing satisfies the 2nd, 3rd, and 4th criteria of a good complete segmentation, mentioned earlier in this section, but not the others. The first criteria is not satisfied because, in general, the number of seeds may not be sufficient to create a region for every pixel. The 5th criterion may not hold because the regions grown from two nearby seeds are always regarded distinct, even if those seeds are defined within a potentially uniform part of an image.

The top-down split-and-merge algorithm considers initially the entire image to be a single region and then iteratively splits each region into subregions or merges adjacent regions. This continues until all regions become uniform, or the desired number of regions establishes.

There are problems with regional segmentation of any form:

1. "Meaningful" regions may not be uniform: surface properties of a solid body will vary in brightness or color dependent on the existence of slowly varying gradients due to lighting conditions.

2. Lighting or curvature affect the appearance, e.g. a sphere illuminated by a point light source may have intensities varying from pure white to black, yet is a single surface. It is very unusual in practice for an image to be composed of uniform regions of similar intensity, or color, or texture etc.

Hybrid Methods: All methods in each of the above mentioned categories have their own limitations. Therefore, proposing improved algorithms which are having these limitations removed or minimized is the goal here. By applying two methods hierarchically, it is hoped that the hybrid method improves segmentation results. For instance, a hybrid method utilizing adaptive thresholding and region growing based on low-degree polynomial fitting has been proposed in [42]. Edge-based region growing algorithms are also common hybridization attitudes [80, 99]. Another common hybridization trend is the combination of clustering methods with region growing approaches [1, 78, 116]. In this domain, the aim mostly is to overcome the discontinuity in clustering segmentation.

2.1.3 Noise

Image noise is random variation in intensity or color components of an image, which is usually caused by capturing or transmitting devices and is undesirable because it adds spurious information to the image. Both analogue and digital imaging devices can cause noise. Two types of common noise which are more considered in the literature are Gaussian and Salt & Pepper noise.

- **Gaussian Noise:** is an statistical noise that the value of noise is modeled using a Gaussian distribution independent of the intensity of pixels. Gaussian noise in digital images mainly arises via capturing devices during image acquisition (such as sensor noise caused by poor illumination), high temperature, or transmission (such as noise existing in electronic circuits). It usually causes everything look smoothed and blurry in an image.

- **Salt & Pepper Noise:** impulse or fat-tail distributed noise, which sometimes is referred to as *salt and pepper* noise, can be produced by malfunctioning pixels in camera sensors, faulty memory locations in hardware, analog-to-digital converter errors or bit errors in a transmission [22]. This means, images are usually damaged by impulsive noises during acquisition or transmission. It appears as sparsely occurring white and black pixels. Since the corrupted pixel by impulsive noise contains no information about the present image, impulse noisy image segmentation is a challenging issue.

Fuzzy C-Means (FCM) and Particle Swarm Optimization (PSO) are the two main tools that we have used throughout this thesis to introduce new approaches for noisy image segmentation. The next two sections briefly describes them.

2.2 Fuzzy C-Means

Fuzzy C-Means (FCM) was first introduced in [48], and then extended in [64] by Bezdek. It is a clustering algorithm in which all the datapoints are considered to belong to all the clusters with a degree of membership. Datapoints (pixels in our application), are represented as a set, $X = \{x_1, x_2, ..., x_N\}$, where x_i is a feature vector in a p-dimensional vector space associated with each pixel. The aim is to find C cluster centers in a way that the following objective function is minimized:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} d^2(x_i, v_j) \tag{2.1}$$

where N and C are the number of pixels and clusters respectively, m is the weighting exponent, $d(.,.)$ is the distance metric, and $d(x_i, v_j)$ is the distance between pixel x_i and cluster centre v_j which uses the Euclidean metric in our approach. u_{ij} is a value specifying the degree of membership of pixel i to cluster j which needs to satisfy:

$$\begin{aligned} &\forall i \in \{1,2,...,N\}: \quad \sum_{j=1}^{C} u_{ij} = 1 \\ &\forall i \in \{1,2,...,N\}, \forall i \in \{1,2,...,C\}: \quad u_{ij} \in [0,1] \end{aligned} \tag{2.2}$$

Using Lagrange multipliers the two following updating equations are obtained which are necessary but not enough to have Eq. (2.1) at its minimum:

$$u_{ji}^{k+1} = \frac{1}{\sum_{l=1}^{C} \left(\frac{d(x_i, v_j^k)}{d(x_i, v_l^k)} \right)^{2/m-1}} \tag{2.3}$$

$$v_j^k = \frac{\sum_{i=1}^{N} \left(u_{ij}^k \right)^m x_i}{\sum_{i=1}^{N} \left(u_{ij}^k \right)^m} \tag{2.4}$$

v and u are updating iteratively through the following steps where k is the iteration index. These equations repeat until the algorithm converges, meaning a certain degree of accuracy is obtained. At the end, the cluster centers and degrees of membership are obtained:

1. Set values of C, ε, and m.

2. Randomly initialize the centers of clusters, v_j^0

3. Unless the required accuracy is obtained, repeat the following steps for all pixels and clusters:

4. For pixel i and cluster j calculate u_{ji}^k using v_j^{k-1} according to Eq. 2.3.

5. Update v_j^k using Eq. 2.4

6. Test if accuracy is obtained: $\parallel v_j^k - v_j^{k-1} \parallel < \varepsilon$.

evolutionary computation

2.3 Particle Swarm Optimization

This section provides a very brief introduction to evolutionary algorithms first, then discusses particle swarm optimization.

2.3.1 Evolutionary Computation

An area of computational intelligence is Evolutionary Computation (EC) in which the techniques are inspired by biological mechanisms [10]. There are two main categories in this area [10]: Evolutionary Algorithms (EAs) and Swarm Intelligence (SI). Since EC techniques make no assumption about the search EAs follow natural selection and survival of the fittest solutions, and SI is inspired by the collective intelligence of a group of agents that resembles the social behavior of animals such as birds and fish [15, 21]. EC techniques perform well in many optimization problem due to their global search abilities. This characteristic of EC techniques is related to the usage of potential populations of solutions in a search space rather than selecting a single solution in the search space. As a result, EC techniques have been successively applied to a variety of applications in different fields [51].

2.3.2 Particle Swarm Optimization

Particle Swarm Optimization is a computational optimization algorithm from the category of SI introduced in [50, 70]. Due to efficiency, robustness, and simplicity [51] the technique has been modified many times, for general and specific applications. The search algorithm is motivated by the social behaviors of organisms. Particularly, choreography of birds flock led to the design of PSO. The algorithm is initialized with a swarm of potential solutions in a multidimensional space. Each solution, also known as particle, has the ability to move. Therefore, particle i has two parameters as x and v which specify its location and speed in the search space, respectively. During the movement, each particle updates its position and velocity according to its own experience, and that of its neighbors. i is in interactive communication with neighboring particles in order to find the best position (final solution). The best so-far position of each particle is called *pbest*, and the best so-far position in the whole swarm is called *gbest*. What really determines the goodness of *pbest*, *gbest*, and basically all particles is a fitness function which is an essential part of PSO algorithm. The fitness function specifies the nature of the optimization problem, and is designed according to the application. Briefly, assuming a D-dimensional search space the ith particle is represented by $\boldsymbol{X_i = (x_{i1}, x_{i2} ... x_{iD})}$ and $\boldsymbol{V_i = (v_{i1}, v_{i2}, ..., v_{iD})}$ as D-dimensional arrays for the positions and velocities. $\boldsymbol{x}$ and $\boldsymbol{v}$ are updated using these two equations:

$$v_{id}^{k+1} = w \times v_{id}^{k} + c_1 r_1 (pbest_d - x_d^k) + c_2 r_2 (gbest_d - x_d^k) \tag{2.5}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{2.6}$$

where $d = 1,2,...,D$, $i = 1,2,...,N$, are the sizes of dimension and swarm, c_1 and c_2 are positive constants, r_1 and r_2 are random numbers, uniformly distributed in the interval [0,1], $k = 1,2,...$, denotes the iteration number, $pbest_d$ and $gbest_d$ represent *pbest* and *gbest* in the dth dimension, and ω is inertia weight which controls the influence of previous velocities on the new velocity. Larger inertia weights indicate larger exploration through the search space while smaller values of the inertia weight restrict the search on a smaller space [51]. Typically, PSO starts with a larger ω, and the decreases gradually over the iterations. We have adopted the following equation for ω to simulate its descending property:

$$\omega = (\omega_{initial} - \omega_{final}) \times \frac{(k_{max} - k)}{k_{max}} + \omega_{final} \tag{2.7}$$

where $w_{initial}$, is the preliminary value of w, w_{final} is the final value of w, k is the iteration number, and k_{max} is the maximum number of iterations.

2.4 Related Work

2.4.1 Noisy Image Segmentation

Among the applications of image segmentation, there are domains where segmentation needs to be done on real noisy images. For instance, medical images suffer from intensity inhomogeneities and noise. Also, natural images have additive Gaussian noise. Therefore, noisy image segmentation has become widely used in the field of medical image processing. Segmenting brain CT scan images [32], ultrasound images [7], and magnetic resonance images (MRI) [**?**] are examples of medical noisy image segmentation. Another field that needs to handle noisy images is remote image analysis with applications in segmenting Synthetic Aperture Radar [120], and satellite [11] images.

2.4.2 FCM-based Noisy Image Segmentation

The traditional applications of FCM to image segmentation fail to produce accurate noisy segmentation results as the objective function did not consider any spatial information. In this manner, the first notable attempt to overcome this weakness is [6] known as FCM_S. The method was proposed to conquer the intensity inhomogeneities present in the segmentation of MRI images by allowing the labeling of a pixel to be affected by its immediate neighborhood. Since FCM_S was computationally expensive, FCM_S1 and FCM_S2 were proposed [38] to improve both efficiency and effectiveness by using a pre-calculated mean and median filtration of the surrounding window for FCM_S1 and FCM_S2 respectively. EnFCM was proposed [118] as another modification of FCM. It uses a linearly weighted filter applied to the noisy image, and then FCM is performed on the intensity histogram of the image. The fact that the number of intensities in an image is usually much fewer than the number of pixels in an average-sized image made EnFCM quite efficient. The algorithms mentioned so far have a tuning parameter, termed α, which has to be large enough to suppress the effect of noise, and has to be small enough to preserve the details in an image. Since these methods are parameter dependent, their utilization was narrowed down to certain types and volumes of noise. FGFCM [25] was therefore proposed later to reduce the parameter dependency of the former modifications. FGFCM proposes a non-linear filtering factor which still has two tuning parameters termed as λ_s and λ_g, but it was shown that the dependency of FGFCM to these parameters is much less than that of the previous methods to α. Motivated by the strengths of all the previous methods, the parameter-free FLICM was proposed [74]. A new fuzzy factor was introduced into the objective function that considers gray and spatial information simultaneously. Although the method is parameter-free and performs better than its predecessors, the segmentation results are not accurate in the case of multi-intensity noisy images [52]. Also, FLICM is problematic when it comes to identifying the class of boundary pixels [129], and severely noisy image segmentation. To partially overcome the shortcomings of FLICM [146] is proposed in which pixel relevance is introduced into the fuzzy factor of FLICM. This way non-local information can be considered to improve FLICM performance in case of high-level noise.

A new scheme is introduced in [56] to incorporate local spatial information into hard c-means (HCM) algorithm. The paper adds a weighted Kullback-Leibler divergence function of the local spatial membership function to fuzzify HCM. A new objective function is introduced in [57] composing of two parts. The first part is a modified version of FCM and the second part is a weighted regularization version of FCM. In both parts distance metrics are modified to take into account the distances in a pixel neighborhood. Both [56] and [57] are tested on simple synthetic and real images corrupted with single-level low-volume noise of Gaussian and Salt & Pepper noise. To tackle the common issues with the approaches adding spatial penalty term in FCM, a auto-tuning-parameter algorithm is proposed in [60]. The approach detects noise pixel from variance of gray levels around that pixel, and then uses this information to to balance a denoising term and a detail preserving term in the objective function. This paper analyses the effect of noise level variation on low levels for real and synthetic images. A forth order partial differentiation equation filter to deal with cancer detection in microscopic biopsy is proposed in [75]. The approach improves FCM segmentation results solves the problem of blocky artifacts and also edge preservation by incorporating denoising and segmentation in one package. A modified version of FCM is proposed in [79] in which L_p norm is used instead of L_2 for distance calculation. Also, spatial and color information are incorporated into the objective function to deal with noise and outliers. The method considers severe Gaussian noise and has been tested on gray and color images. The algorithm proposed in [147] introduces a new strategy based on peak detection to initialize the cluster centers. Also, pixel relevance and spatial information is added into the FCM objective function, and to further improve the results by detecting the misclassified pixels and putting them into the correct clusters. The method is tested on synthetic and real images with high levels of noise.

2.4.3 PSO for FCM-based Noisy Image Segmentation

Compared with the application of EC methods in fields such as object classification or edge detection, there are not many works in noisy image segmentation using EC techniques. PSO is one of the major paradigms that have been used for noisy image segmentation. This section reviews some typical works in this area.

Recently, researches have been carried out to combine heuristic algorithms particularly PSO with FCM. The common approach deals with the initialization problem of FCM. FCM performance is influenced by the initial position of the cluster centers, and improper initialization of cluster centers can make FCM to fall into a local optima. PSO can effectively be used to find good cluster centers for FCM effective performance. Potential solutions (particles) in PSO as a tool for good initialization of FCM are possible values for cluster centers within the intensity diversity of pixels. They take the objective function of a FCM-based clustering method as the fitness function, and then try to find the optimum positions of the cluster centers that minimize the objective function the most. This eliminates the updating formula for cluster centers, but the fuzzy membership updating formula is needed to obtain the value of degree of membership. Knowing that, FCM-based clustering methods find their cluster centers in an optimization process, the algorithms in this approach are only significantly more effective than their predecessors when other accuracy steps are also added [124, 13].

PSO for FCM Initialization

In the following we briefly mention the studies in which PSO is used for a better initialization of FCM. Using gray-level and neighboring spatial information, a noisy image segmentation algorithm is introduced in [150]. First, a two-dimensional histogram is constructed using gray-level and spatial information. The histogram uses each pixel and its neighboring pixels' intensity information where mean or median filtering is successively applied (two times) to an image to create the information in the two dimensions. Next, predator-prey PSO is used to obtain FCM cluster centers. The algorithm is tested on synthetic images with low-level Gaussian noise.

To optimize the clustering performance of FCM, PSO and Markov random filed are hybridized using spectral and spatial information [82]. The similarity metric in FCM in modified using new spatial information provided by Markov random filed. The method is tested only synthetic single-level noisy images. An image segmentation method of thermal wave inspection is proposed in [61]. The methods uses homomorphic filtering to enhance the noisy images, and then a PSO-derived FCM is applied for segmentation using histogram information. Images have limited diversity and are composed of geometrically simple regions.

Multi-scale FCM is combined with PSO for ultrasound image segmentation. Multi-scale FCM for segmentation of magnetic resonance images [145] uses a series of images with different levels of spatial information. First, speckle reducing anisotropic diffusion is applied to remove noise from an ultrasound image and construct a multiple scales images. Next, PSO is used to look for the global optima of cluster centers in multiple-scale FCM. The method is tested on synthetic and real images corrupted with speckle and Gaussian noise. A three-step algorithm is proposed in [13] based on another modification to FCM [38]. The first step deals with a proper initialization of the objective function of the algorithm in [38] which has added some spatial information to FCM objective function. The second step, segments the image with the properly initialized FCM-based clustering algorithm using Mahalanobis instead of Euclidean distance to calculate the similarity metric. Third, a postprocessing step to refine the segmentation results of the previous step by reclassifying misclassified pixels. The method is tested on low-level synthetic and simulated magnetic resonance brain noisy images.

The same authors extend their work to a multi-objective FCM combining Pareto-optimal clusters [14]. Multi-objective PSO is used to produce a set of Pareto-optimal solutions based on two fitness metrics: a region-based fitness and an edge-based metric. The region-based metric improves what has been proposed in [13], and the edge-based fitness uses some contour statistics and number of connected components to find proper segmentation solutions. To select the final solution to multi-objective optimization, a new scheme is presented to combine all solutions to better segmentation results. An

FCM-based magnetic resonance brain image segmentation is proposed in [5]. To divide the noisy brain region into required number of clusters the number of boundary points are determined from the dataset. Soft-computing techniques such as PSO, genetic algorithm and bacteria foraging optimization are applied to optimize the boundary points. The method is applicable to specific low-noise magnetic resonance brain images.

PSO is incorporated with EnFCM and FGFCM to improve the performance of these algorithms [124]. Also, to improve the convergence speed of the proposed algorithms, the optimal-selection-based suppression strategy [149] is applied to the membership values. Differential evolution (DE) is used along with PSO address the initialization problem of FCM when benefiting from the characteristic of both PSO and DE [83]. PSO has a good performance in local search and fast convergence, and DE has a good performance in global search. The method shows to have anti-noise abilities in segmentation of low level real images. Another algorithm that deals with other issues related to FCM as well as the initialization problem is IKPCM [91]. Inspired by Kernel Possibilistic C-Means (KPCM), IKPCM takes into account a few aspects of FCM clustering by integrating initialization step, distance metric, spatial information, and outlier rejection into their algorithm. The initialization problem is taken care of by using PSO, the membership function of KPCM is modified using outlier rejection, spatial information is provided by using a neighboring window around the pixel under consideration, and level set is used for segmentation. The method has been tested on synthetic and real images of low-level noise.

PSO Hybridization with FCM in Other Schemes

There are a few other research works available combining PSO with FCM for problems other than effective initialization which we briefly mention here. A new parameter tuning algorithm using PSO and GA is introduced in [53] to tune the degree of attraction of neighboring pixels for a modification of FCM called improved FCM (IFCM). IFCM [113] introduces two factors into FCM to determine the contribution of neighboring pixels in segmentation results. The first factor considers difference between features of the pixels under consideration and the neighboring pixels, and the second factor uses the relative location of the neighboring pixels. Tuning the parameters related to these two factors plays an important role in the performance of IFCM. [53] uses PSO, GA, and a hybridization of PSO and GA based on the strengths of the two algorithms for parameter tuning in IFCM. The method is tested on synthetic and simulated/real magnetic resonance images corrupted with low level of noise.

New similarity metrics are proposed in [119, 121] for FCM in which PSO is used to weight different features in the new similarity metric. [121] uses neighboring pixels' texture and relative position information to introduce the new similarity metric. Grey-level co-occurrence matrix and wavelets are used to extract texture information from a neighboring window around each pixel. Then PSO is used to optimize the parameters related to features in a new similarity metric. The method is tested on synthetic texture and real SAR (Synthetic Aperture Radar) images. The other paper [119], uses similar strategy by using super pixels. Super pixels are extracted using information from different scales of wavelet transform, then a new similarity metric is proposed to determine the influence of similar and neighbor superpixels in FCM clustering. PSO is used to optimize the parameters related to the new similarity metric. The method is tested on synthetic, SAR and color natural image segmentation. One drawback of the the two mentioned algorithms is their huge computational cost.

2.5 Chapter Summary

Noisy image segmentation is a filed that has being applied in real-world application. More development and exploration is needed in this field in order to fulfill all the needs properly. Based on the provided literature review in this chapter in the area of FCM-based noisy image segmentation, there are limitations related to the existing approaches, and also gaps to be addressed.

- Many of the existing approaches are parameter-dependent for the level of noise.
- Severely noisy image segmentation has not been explored properly.
- The effect of noise level variation has not been investigated.
- The number of noisy images is often too few that it is not possible to make a correct conclusion out of the results for similar cases.
- The exiting PSO contributions is not either effective or efficient enough.

This thesis aims to tackle these issues by developing new algorithms using PSO.

Chapter 3

Adaptive Wavelet Shrinkage for Severely Noisy Image Segmentation

3.1 Introduction

Tackling the problem of severely noisy image segmentation requires specific tools and strategies. There are many FCM-based image segmentation methods that do not show promising results on severely noisy image segmentation. While there has been much research on modifying the objective function of FCM [6, 38, 118, 25, 74, 129], or new similarity metrics for FCM [94, 95] to improve the performance, there has not been a stress on the importance of feature choice and manipulation in the literature. Also, noisy image segmentation algorithms that do not require any assumptions about the noise are important in different applications in which the noise type or volume is unknown. There are also domains where noise is a mixture of different types of noise, or information about the noise is not available. This chapter proposes two new FCM-based noisy image segmentation algorithms using PSO, wavelet transform, and edge detection. The algorithms require no parameter tuning for the type or volume of noise, and show stable and accurate results on different images corrupted with different noises at different volumes.

3.1.1 Chapter Goals

This chapter targets the problem of noisy image segmentation from the feature analysis point of view. It explores the potentials of PSO and wavelet-based thresholding for noisy image segmentation. We demonstrate the effect of appropriate choice of features and feature processing on FCM performance with superior results compared to other state-of-the-art methods. Following are the chapter objectives briefly:

- Investigating the effect of proper choice of features, and feature manipulation on severely noise removal.
- Studying the effectiveness of wavelet features and wavelet-based thresholding at providing proper spatial information, removing noise, and segmentation results.
- Exploring the abilities of PSO in feature manipulation, and proposing an approach needing no parameter tuning which performs adaptively based on the type or volume of noise for each noisy image.
- Investigating the effect of number of wavelet transformation scales and the type of wavelet thresholding in conjunction with PSO for denoising matters.
- Evaluating the goodness of clustering fitness metrics in the PSO-based feature manipulating system.
- Exploring the effect of edge detection methods on preserving more details, and more accurate segmentation results.
- Comparing the performance of the proposed methods with other FCM-objective-function-modified methods on severely noisy image segmentation problems.

3.1.2 Chapter Organization

The remainder of this chapter is as follows. Section 3.2 is dedicated to the methodology of the proposed methods. Section 3.3 introduces the foundation of the new proposed severely noisy image segmentation algorithm, and an extended version of it based on the concept of edge enhancement. The datasets, comparison algorithms, and evaluation metrics are provided in Section 3.4. Section 3.5 is devoted to results and discussions, and Section 3.6 concludes this chapter.

3.2 Methodology

This section briefly explains two techniques we have utilized in the construction of the new algorithms in this chapter: wavelets and Canny edge detector.

3.2.1 Wavelets

Wavelet transform is a technique which provides multi-resolution representations for image analysis [86]. In comparison to other frequency-based techniques such as Fourier transform, it has the advantage of providing information from both frequency and space domains which is essential for dealing with non-stationary signals like images. The main difference between different wavelets is the choice of the basis (ψ) and scaling (φ) functions. Wavelet analysis is in fact the correlation between the signal under consideration and the basis functions. One of the primary properties of wavelet transforms is its sparsity when applied to real-world signals. This means that they typically contain a few large coefficients encompassing the majority of the energy of the signal. The rest are unimportant coefficients that carry no significant information. This feature of wavelets is quite favorable for image denoising. Discrete Wavelet Transform (DWT) is a hierarchical subband system in which subbands are obtained from logarithmically spacing the frequency domain. To apply wavelet decomposition on an image, two-dimensional Discrete Wavelet Transform (2D-DWT) is applied by applying 1D-DWT along the rows, and then the columns of an image. After applying 2D-DWT on an image, it is decomposed into four subbands. These four subbands are the results of applying high-pass and low-pass filters in vertical and horizontal directions, and are named HH1, LH1, HL1, and LL1, or diagonal (D), horizontal (H), vertical (V) and approximation (A) coefficients respectively. Fig. 3.1 shows a single-scale two-dimensional filter bank that could produce four subbands. In this figure, h_φ is a scaling vector, and $h_\varphi(n)$ is the set of scaling function coefficients. Also, h_ψ is the wavelet vector, and $h_\psi(n)$ are the wavelet function coefficients. $*$ means convolution, and $2\downarrow$ means a reduced-resolution approximation of the image/subband which is done by down-sampling the convolved results by a factor of 2.

LL1 (or A) can be further decomposed to give another set of coefficients at the second scale (Fig. 3.2-b). Fig. 3.2 shows how further decomposition takes place in

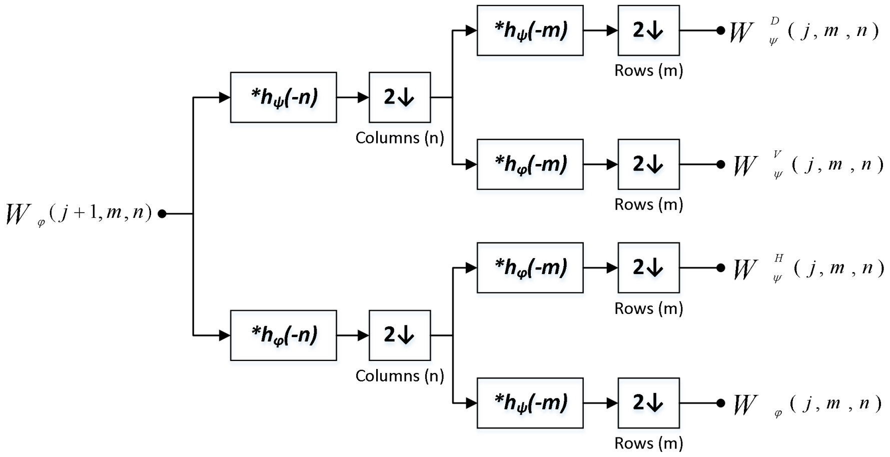


Figure 3.1: A two-dimensional, four-band filter bank for subband image coding (the Figure adapted from [58]).

a three-scale 2D-DWT providing an example using a sample image. Mathematically, decomposition scales could be infinite, but after a few scales, the subbands carry no useful information. The coefficients at each scale can be used as features representing texture, or be thresholded for denoising, or even as features for image compression.

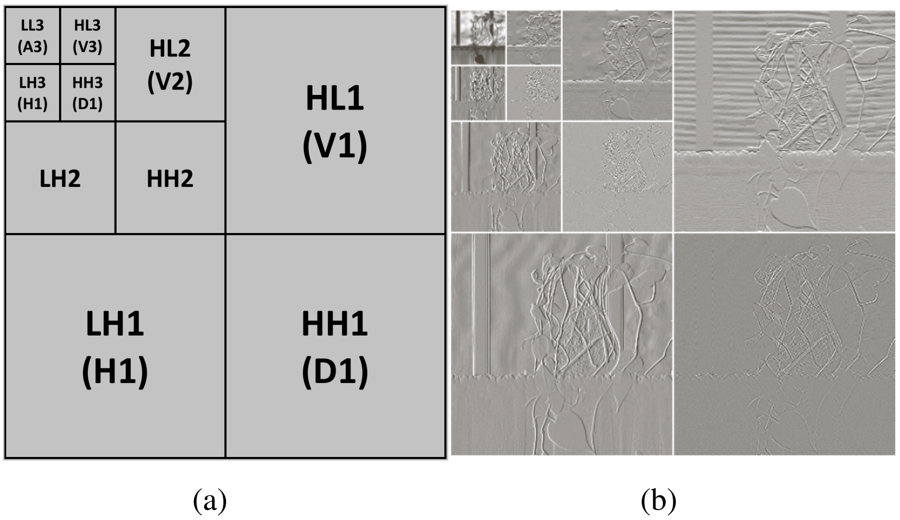


Figure 3.2: Three scale, 2D-DWT. (a) the resulting two-scale decomposition; (b) A sample image transformed using this configuration.

Applying 2D-DWT to an image $f(x,y)$ of size $M \times N$ is performed according to the block diagram provided in Fig. 3.1, using the following formulas:

$$
\begin{aligned}
W_{\varphi}(j_0,m,n) &= \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\varphi_{j_0,m,n}(x,y) \\
W_{\psi}(j,m,n) &= \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\psi^{i}_{j,m,n}(x,y), i=\{H,V,D\}
\end{aligned}
\tag{3.1}
$$

in which j_0 is an arbitrary starting scale, and the $W_{\varphi}(j_0,m,n)$ coefficients define an approximation of $f(x,y)$ at scale j_0. The $W_{\psi}(j,m,n)$ coefficients add horizontal, vertical, and diagonal details for scales $j \geqslant j_0$. We normally set $j_0 = 0$ and $N = M = 2^J$ so that $j = 0,1,2,...,J-1$, and $m = n = 0,1,2,...,2^j - 1$. $\varphi_{j,m,n}$ and $\psi^{i}_{j,m,n}(x,y)$ are the scaled and translated basis functions:

$$
\begin{aligned}
\varphi_{j,m,n} &= 2^{j/2}\varphi\big(2^{j}x-m, 2^{j}y-n\big) \\
\psi^{i}_{j,m,n} &= 2^{j/2}\psi^{i}\big(2^{j}x-m, 2^{j}y-n\big), i=\{H,V,D\}
\end{aligned}
\tag{3.2}
$$

where i indicates the term *directional wavelets* assuming the values H, V and D. φ and ψ are the separable scaling function and separable directionally sensitive wavelets respectively.

3.2.2 Canny Edge Detector

The Canny edge detector [28] is an effective edge detection algorithm which we use in our segmentation approach. It has three main objectives:

1. Find all the edges with no spurious responses. This means the detected edges have to be as close as possible to the true edges.

2. Localize the edges in a way that the distance between a point marked as an edge and the center of the true edge should be minimum.

3. Return only one point for each true edge point. This could be fulfilled by minimizing the number of local maxima around the true edge.

If $f(x,y)$ denotes the input image, and $G(x,y)$ denotes the Gaussian function:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.3}$$

First, a smoothed image, $f_s(x,y)$, is formed by convolving G and f:

$$f_s(x,y) = G(x,y) * f(x,y) \tag{3.4}$$

Next it computes the gradient magnitude, M, and direction, α as:

$$M(x,y) = \sqrt{(\frac{\partial f_s}{\partial x})^2 + (\frac{\partial f_s}{\partial y})^2} \tag{3.5}$$

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right] \tag{3.6}$$

$M(x,y)$ and $\alpha(x,y)$ are matrices of the the same size as the image $f(x,y)$. The edges in the image correspond to the high values of $M(x,y)$. Since gradient is used to create $M(x,y)$, it contains wide ridges around local maxima. To localize the edges, these ridges have to be thinned by keeping the large values of gradient. The Canny edge detector uses a *nonmaxima suppression* to thin the broad edges in $M(x,y)$ into ridges that are only one pixel wide. This approach specifies a number of discrete orientations of the gradient vector, and quantizes all possible edge directions into them. The result is a nonmaxima suppressed image which is thresholded to reduce false edge points. To make sure false edges are removed as much as possible, and minimum number of actual valid edge points are eliminated, Canny's algorithm uses *hysteresis thresholding* with two threshold values. The final step is a connectivity analysis to fill the gaps in the thresholded map, and link the edges.

3.3 PSO-based Wavelet Shrinkage for Noisy Image Segmentation

This section proposes an FCM-based severely noisy image segmentation algorithm and an extension to it using PSO-based wavelet shrinkage and Edge enhancement. The fundamental algorithm introduces an adaptive PSO-based wavelet shrinkage, and is called Wavelet Shrinkage-based Fuzzy C-Means (WS-FCM). The extended version adds an edge enhancement technique to the fundamental algorithm, and is called Edge-enhanced Wavelet Shrinkage-based Fuzzy C-Means (EWS-FCM). WS-FCM introduces a new way of combining PSO, FCM, and wavelets for effective noisy image segmentation.

PSO is used to adaptively search for optimal threshold values based on the noise and intensity properties of each image to threshold wavelet coefficients. FCM is used to evaluate the performance of threshold values, and perform the final segmentation. Fig. 3.3 shows the block diagram of the proposed WS-FCM and EWS-FCM. The flowchart inside the blue dashed line shows the block diagram of the proposed WS-FCM, and the flowchart inside the red dashed line shows the flowchart of the proposed EWS-FCM. For EWS-FCM, the block diagram has four main stages. First, an image is transformed into the wavelet domain. Second, the detail coefficients at different scales are thresholded in a PSO search procedure until the best set of threshold values are found. Third, the Canny edge detector is applied to just the approximation coefficients to create an edge map. This map is aggregated with the approximation coefficients. Fourth, the thresholded coefficients along with the aggregated approximation coefficient are used to reconstruct the image, and finally the resultant denoised image is used for an FCM-based segmentation.

Eqs. 3.7 and 3.8 show the process formulations of the steps in WS-FCM and EWS-FCM respectively for the input image X:

$$\begin{aligned}
&(A,H,V,D) \leftarrow W(X) \\
&\theta_{best} \leftarrow PSO\Big((H,V,D),\theta\Big) \\
&(H_t,V_t,D_t) \leftarrow Th\Big((H,V,D),\theta_{best}\Big) \\
&\hat{X} \leftarrow W^{-1}(H_t,V_t,D_t,A) \\
&X_s \leftarrow FCM(\hat{X})
\end{aligned} \tag{3.7}$$

$$\begin{aligned}
&(A,H,V,D) \leftarrow W(X) \\
&\theta_{best} \leftarrow PSO\Big((H,V,D),\theta\Big) \\
&(H_t,V_t,D_t) \leftarrow Th\Big((H,V,D),\theta_{best}\Big) \\
&A_e \leftarrow a \times A + (1-a) \times Canny(A) \\
&\hat{X} \leftarrow W^{-1}(H_t,V_t,D_t,A_e) \\
&X_s \leftarrow FCM(\hat{X})
\end{aligned} \tag{3.8}$$

where W is the wavelet transformation, A, H, V and D are the approximation, horizontal, vertical, and diagonal coefficients at different scales. PSO is the PSO search procedure, and θ_{best} is the best set of threshold values obtained from the PSO search. Th is the thresholding function using threshold value θ_{best}. *Canny* is the Canny edge detector, a is a constant, and A_e is an edge-enhanced version of the approximation coefficients. W^{-1} is the reverse wavelet providing the reconstructed image, $\hat{X}$, and FCM is FCM clustering creating the segmented image, X_s.

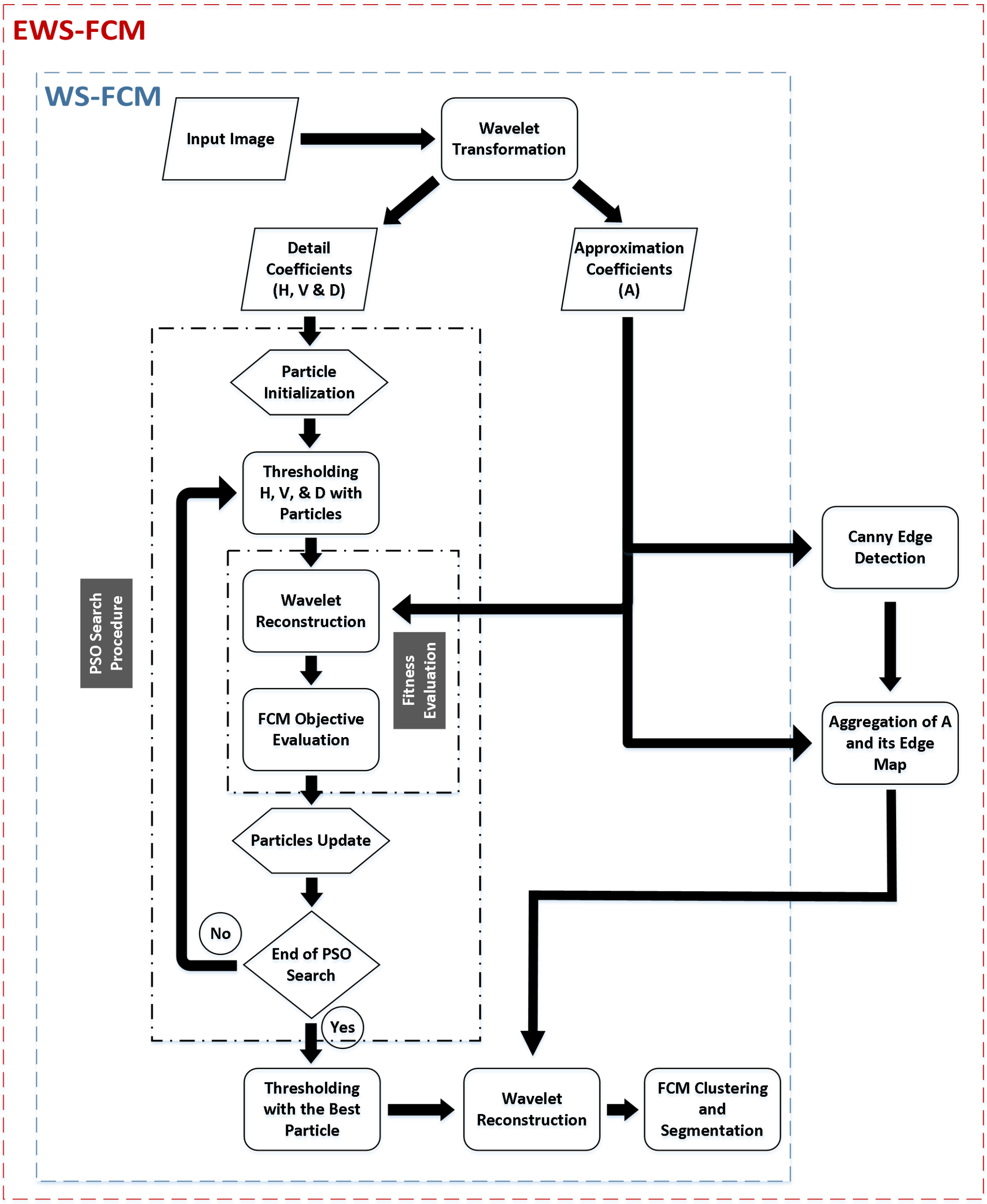


Figure 3.3: Block diagram of the proposed WS-FCM and EWS-FCM algorithms.

3.3.1 Wavelet Transformation

Threshold selection in wavelet-based thresholding plays an important role in denoising an image. Improper thresholding of wavelet coefficients would result in an undesirable image. The threshold value has to be large enough to attenuate the effect of noise, and at the same time it has to be small enough to preserve the details of the image. To demonstrate the importance of proper thresholding on segmentation, Fig. 3.4 shows an example in which FCM-based segmentation is carried out on a sample image that has been under-thresholded, over-thresholded, and properly thresholded. The sample image (Fig. 3.4-a) is contaminated with Gaussian noise of 20% variance (Fig. 3.4-b). After applying 2D-DWT to the noisy image, the wavelet coefficients are thresholded, reconstructed, and the reconstructed image is segmented using FCM. Fig. 3.4(c-e) show the under-thresholded, over-thresholding, and optimally thresholded results, respectively. Under-thresholding leaves a lot of noise which causes many redundant segmented regions in the image, while over-thresholding deforms the boundary lines, and also causes redundant segmented regions around the edges. Although not demonstrated in this example, over-thresholding can also eliminate details from an image. These errors are greatly reduced in the optimally thresholded image.

Most conventional approaches apply a single threshold to all the wavelet coefficients [19]. The new research trend, however, is to have different threshold values for different subbands or scales, but this greatly increases the difficulty of identifying optimal threshold values. The number of scales for which the wavelet transform is performed also becomes very important: too few scales could keep a considerable volume of noise in the image, and too many scales increases the computational cost.

Using different wavelet filter families in the wavelet transform results in different properties [18]. Performing different experiments with different wavelet filters at different scales using various images, we realized using a “Coiflet 5” wavelet at a five-scale transformation would result in an ideal compromise between segmentation accuracy and computation.

3.3.2 PSO Representation

In our PSO-based search, each particle is a set of threshold values for thresholding of different wavelet subbands. Since noise energy spreads at different levels at different

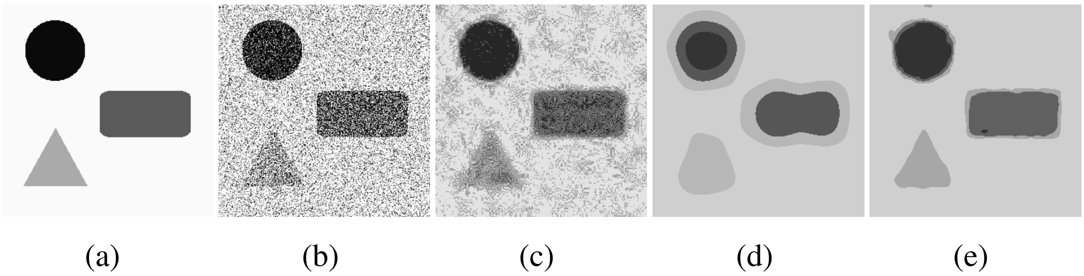


Figure 3.4: The effect of different thresholding conditions on segmentation. (a) the original image, (b) the noisy image, segmentation results of (c) under-thresholding, (d) over-thresholding, and (e) optimally-thresholding.

frequencies, thresholding a noisy image at different frequencies makes the denoising procedure more effective. To cover all the frequencies, we set a threshold value for each detail coefficient at each scale. Given that we have a five-scale transformation, and there are three detail coefficients at each scale, each particle i, P_i, consist of a 1D vector of 15 threshold values that represents a 15D search space:

$$P_i = [\theta_1, \theta_2, ..., \theta_{15}] \tag{3.9}$$

θ_1, θ_2, and θ_3 are threshold values allocated to H, V, and D at scale one. The next three threshold values (θ_4, θ_5, and θ_6) are for the detail coefficients at level two, and so on.

To make the search more efficient, we set minimum and maximum limits for the threshold values:

$$0 < \theta_i < \theta_U \tag{3.10}$$

θ_U is the **Universal** threshold, provided by the Visu Shrink method [47]:

$$\theta_U = \sigma\sqrt{2\ln(n)} \tag{3.11}$$

where σ is the standard deviation of noise, and n the image size. Since we make no assumption about the volume of noise, the σ value will be obtained using a robust median estimator as proposed by [47]:

$$\sigma = \frac{median\big[|w_{ij} : i, j \in HH_1|\big]}{0.6745} \tag{3.12}$$

where w_{ij} are noisy coefficients of subband HH_1.

The initial solutions for particles are selected from the $[0, \theta_U]$ interval, and in each iteration the members of particles outside this interval will be replaced by $Xpbest$ positions.

3.3.3 Wavelet-based Shrinkage

Thresholding is performed on wavelet coefficients to suppress the noise by removing or attenuating the small coefficients that are more likely to be corrupted by noise, and keeping or modifying the large coefficients that carry the image information. Two classic thresholding functions are hard and soft thresholding [47]. The hard thresholding is also called wavelet thresholding, and simply zeros coefficients values smaller than the threshold.

$$Y = \begin{cases} w, & \text{if } |w| > \theta \\ 0, & \text{if } |w| \leq \theta \end{cases} \tag{3.13}$$

Soft thresholding is also called wavelet shrinkage, because it not only zeros coefficients values smaller than the threshold but also "shrinks" coefficients values larger than the threshold as shown in Fig. 3.5.

$$Y = \begin{cases} sign(w)(|w| - \theta), & \text{if } |w| > \theta \\ 0, & \text{if } |w| \leq \theta \end{cases} \tag{3.14}$$

We use the soft thresholding (wavelet shrinkage) function in our method. PSO-provided threshold values in form of particles go through this function to produce thresholded coefficients. The approximation coefficients (A subband) are not usually thresholded as they represent low-frequency parts of an image so they are less affected by noise, and typically contain very important components of an image. Thresholding this subband is likely to decrease the segmentation accuracy.

Fitness Evaluation

Performance evaluation of particles is required in each iteration to guarantee the best solution at the end. Thresholded coefficients from the current iteration along with the

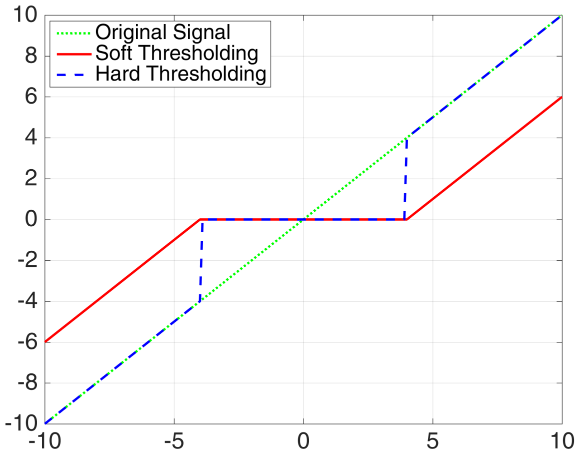


Figure 3.5: The hard and soft thresholding functions.

approximation coefficients are used to reconstruct the image via a reverse wavelet transform. The intensity resultant picture is then used to calculate the FCM objective function as in Eq. 3.15, where datapoints are the intensity values allocated to each pixel. This fitness evaluation targets the finer segmentation results directly while targeting denoising indirectly.

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} d^{2}(x_i, v_j) \tag{3.15}$$

3.3.4 Edge Enhancement

The research trend in wavelet-based image denoising leaves the approximation coefficients intact, because it is often considered that they have no part in improving the denoising performance. We introduce a scheme that makes good use of approximation coefficients to preserve more edge information in a noisy image, and results in more accurate segmentation results. Instead of thresholding, we use edge detection techniques to produce an edge-enhanced version of the approximation to improve the final segmentation results.

The edge detector used in this study is the Canny edge detector [29]. We determine the standard deviation of the Gaussian filter in the detector according to the noise level using Eq. 3.12. Threshold values in the detector are determined heuristically depending on the input data. Canny suggests that the ratio of the high to low threshold should be two or three. If A is the approximation subband, and the edge map produced by applying the Canny edge detector to A is A_e, the new enhanced version of A, A_f, is obtained as follows:

$$A_f = a \times A + (1 - a) \times A_e \tag{3.16}$$

in which a is a constant to be determined experimentally according to the image intensity and region complexity. Eq. 3.16 performs a weighted aggregation of approximation coefficients and its edge map in order to emphasize edge information in the final intensity image.

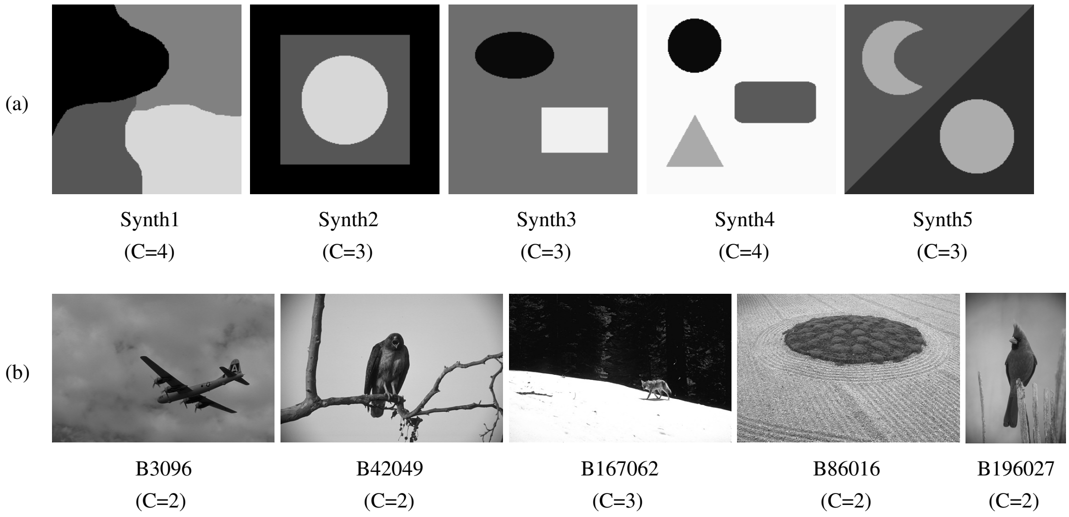
3.3.5 Segmentation

WS-FCM and EWS-FCM use the best set of threshold values found by the PSO search to threshold the details coefficients, and the approximation coefficients are edge enhanced in EWS-FCM. All the details and approximation coefficients are then used to reconstruct the image. For wavelet reconstruction a reverse wavelet is performed using the modified A (or intact A), H, V, and D. The result is a denoised image according to FCM clustering criteria. A final FCM clustering using the intensity information from the denoised image is done in order to cluster the pixels according to their intensity value. The mean intensity value of the pixels inside each cluster determines the gray value of the cluster. Therefore, segmented regions are pixels inside each cluster being represented with this mean value.

3.4 Experiment Design

3.4.1 Datasets and Evaluation

We use a synthetic and a real dataset to evaluate the performance of the proposed methods on images with different properties. The synthetic dataset, SynthDB, consists of images in which the regions are geometrically simple, easy to recognize, and are composed of a single intensity value. Running experiments on noisy versions of images from this dataset provides an understanding of how the proposed methods perform on segmentation of noisy regions when dealing with simple images. We have five images in this dataset named $Synth1, Synth2, ...,$ and $Synth5$. The number of clusters while performing FCM-based clustering is determined by the number of coherent regions in each region. The real dataset, BerkDB, is selected from the Berkeley dataset [88] which has been specifically created for segmentation and boundary detection purposes. Here the images are not so simple: the regions are not simple, they are not easy to recognize, and each region is composed of several intensities. Experiments on this dataset reveals how accurate and consistent the proposed method is when encountering real-world images. We have five images in this dataset named $B3096, B42049, B167062, B86016$, and $B196027$. The number of clusters is again determined according to the number of coherent regions in each image. The ground truth is either selected from one of provided groundtruths in the Berkeley dataset (the one with the minimum number of segments), or is formed from one of the groundtruths. For the latter, all the segments belonging to a coherent region are merged together as one segment corresponding to a cluster in FCM results. Fig. 3.6 shows all the images from the two datasets, along with the number of clusters.


(a)

Synth1 (C=4) Synth2 (C=3) Synth3 (C=3) Synth4 (C=4) Synth5 (C=3)

(b)

B3096 (C=2) B42049 (C=2) B167062 (C=3) B86016 (C=2) B196027 (C=2)

Figure 3.6: Original test images from the SynthDB, row (a), and BerkDB, row (b).

To generate the noisy images, we apply two types of common noises: Gaussian and Salt & Pepper. Each image is corrupted with Gaussian or Salt & Pepper noise of different volumes to examine the performance of our method under different severities of noise, and to analyze the effect of noise variation. The variance/density of the noise level for Gaussian/Salt & Pepper noise ranges from 10% to 80% respectively. For the rest of this chapter, "*image*" refers to one of the original images in Fig 3.6, and "*instance*" refers to a sample noisy image. Given that we have five images in each dataset, two types of noise, and eight levels of noise for each type, there are a total of 160 instances, half of which are created from SynthDB (40 Gaussian instances, and 40 Sal & Pepper instances), and the other half created from the BerkDB dataset (40 Gaussian instances, and 40 Salt & Pepper instances).

For qualitative evaluation, we have adopted the common Segmentation Accuracy, SA, metric [6]:

$$SA = \sum_{i=1}^{C} \frac{A_i \cap S_i}{\sum_{j=1}^{C} S_j} \tag{3.17}$$

in which A_i represents the set of segmented pixels belonging to the i^{th} cluster in the segmented image and, S_i is the set of pixels belonging to the i^{th} cluster in the groundtruth image.

We have selected seven other state-of-the-art algorithms to compare our method with: FCM_S1 and FCM_S2 [38], EnFCM [118], FGFCM, FGFCM_S1 and FGFCM_S2 [25], and FLICM [74]. WS-FCM and EWS-FCM are also compared to each other in order to observe the effect of edge enhancement in segmentation accuracy.

3.4.2 Parameter Design

All the parameters related to FCM, PSO, wavelet transform, the edge enhancement process and their values used in this chapter are listed in Table 3.1. The parameter design has been applied based on empirical search through experiments.

3.4.3 Statistical Significance Test

To analyze the non-deterministic behavior of PSO in our algorithms, a pair-wise statistical significance test is performed. Our algorithm is run 30 times independently on each image, and the results in the form of the SA values are compared with the results of other methods using the test. We select the Wilcoxon test with a significance level

Table 3.1: Parameter settings of the proposed method.

Parameter	**Value/Type**
Wavelet filter	Coiflets family
Scale number	5
Termination threshold for FCM	0.001
Maximum number of iterations for FCM	100
weighting exponent (m)	2
Population size	20
PSO iterations	50
c_1 and c_2 in PSO	1
Inertia weight (ω) in PSO	Linearly decreasing from 1 to 0.5
a (in SynthDB)	0.9
a (in BerkDB)	0.7

of 0.05 for this comparison. For more information about this test please refer to [135]. If the p-value (the probability of observing a test statistic as or more extreme than the observed value under the null hypothesis) is greater than the significance level, the pairwise comparison is not considered significantly different. Otherwise, one method is significantly better than the other. Tables that are providing p values also use **+**, **-** and **=** signs to show that one method is significantly better, worse, or not different respectively for an instance.

3.5 Results and Discussion

This section presents the experiments to analyze the functionality of the proposed methods and also to compare them to other methods both quantitatively and qualitatively. In the beginning we show how the proposed PSO approach functions in terms of feature enhancement and segmentation accuracy. The rest of the experiments are presented in separate sub-sections for different datasets and different types of noise. Each sub-section considers two different aspects of the analysis. The first part presents a comprehensive quantitative evaluation and the second part presents some qualitative evaluations.

To have a valid comparison in the quantitative part, both SA and p values are considered. In this part, we first compare the proposed WS-FCM and EWS-FCM individually with the comparison methods on each instance in a pair-wise manner, and then we present the pair-wise comparison of WS-FCM and EWS-FCM. Throughout the pair-wise comparison, a better performance is determined according to the SA value unless the p value from the Wilcoxon test for that pair is bigger than the significance level (0.05). Next, in another evaluation we rank the performance of all the algorithms on each instance. Based on the number of the best and second-best performances for each algorithm, the overall best, second-best, and third best performers are determined. The quantitative analysis in each sub-section concludes with an evaluation of all the algorithms performances on noise level variation on each sample image. This evaluation investigates how the variance and the mean of SA metric changes when the noise level variance ranges between 10% to 80%. In the qualitative part we show segmented results of some sample instances to provide a visual comparison between the algorithms.

3.5.1 Depiction of Feature Enhancement

To show how intensity feature is enhanced during the coarse of adaptive thresholding, histograms of the original, the noisy and the thresholded images are provided in Fig. 3.7. Fig. 3.7(a) shows the histogram of image $B167062$ in which three main regions are recognizable. The two big regions (forest and ground) in the image are presented by pixels belonging to one of the two spikes, and the third region which is the smallest region (the animal) is presented with the fewer number of pixels ranging mostly on the right side of the left spike. Fig. 3.7(b) shows the histogram of the instance $\sigma^2 = 70\%$ in which the pixels are divided into three big chunks. The first two chunks are pixels having either completely white or black intensities, and the third one is composed of pixels occupying all the gray level values between black and white by a relatively similar cardinality. This histogram cannot provide useful information for the number of coherent regions existing in the image as the diversity of intensities are huge and a lot of pixels have turned black or white due to the severe effects of high-level Gaussian noise. Fig. 3.7(c) however, shows that the adaptive thresholding turns the histogram into something similar to the original histogram. The two spikes again present the two big regions, and the smaller third region is ranged between the two spikes in an interval much smaller than the noisy image.

Fig. 3.8 is also provided to show the transition of fitness and segmentation accuracy over the PSO iterations. Fig. 3.8(a) shows how the fitness value decreases and Fig. 3.8(b) shows how SA improves and stabilizes.

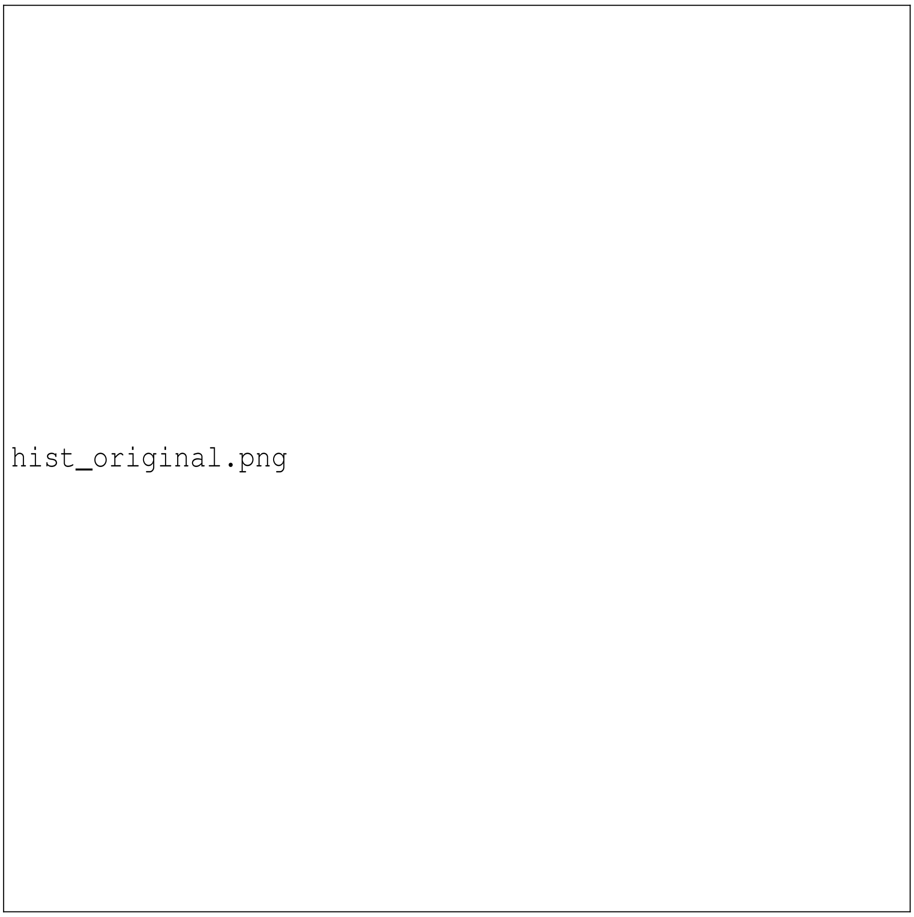
3.5.2 SynthDB

Gaussian Noise (Quantitative Analysis)

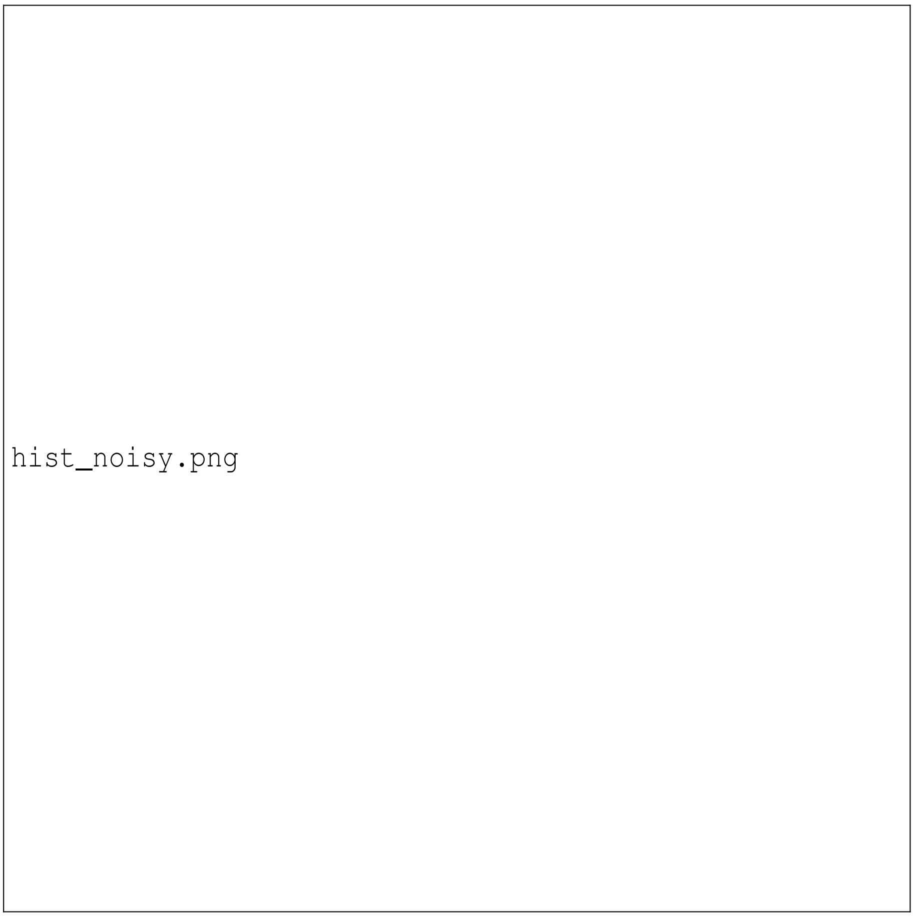
Table 3.2 shows SA values on all SynthDB Gaussian instances for WS-FCM, EWS-FCM and all the comparison methods. For the pair-wise comparison of the proposed WS-FCM with other methods, Table 3.3 provides the p values from the Wilcoxon test. These are the values from the pair-wise significance test of WS-FCM with all the comparison methods over all the 40 instances. There is one instance in Table 3.2 where the p value is bigger than the significance level. This instance belongs to the comparison of WS-FCM and FLICM on $Synth4$, $\sigma^2 = 80\%$. Therefore, except for this instance the performance difference between WS-FCM and the other algorithms is significant, and is determined by the SA values provided in Table 3.2.

An analysis of the results from Tables 3.2 and 3.3 shows that our WS-FCM is significantly better than FCM_S1, FCM_S2, EnFCM, FGFCM, FGFCM_S1, and FGFCM_S2 in all 40 instances. In comparison to FLICM, our WS-FCM still possesses 36 (out of 40) significantly better performances. There are only three instances that FLICM has significantly better performances than WS-FCM, and they are all low-level noisy instances, $Synth2$, $Synth3$, and $Synth4$ all with $\sigma^2 = 10\%$. There is also one instance ($Synth4$, $\sigma^2 = 80\%$) that WS-FCM is not significantly different from FLICM.

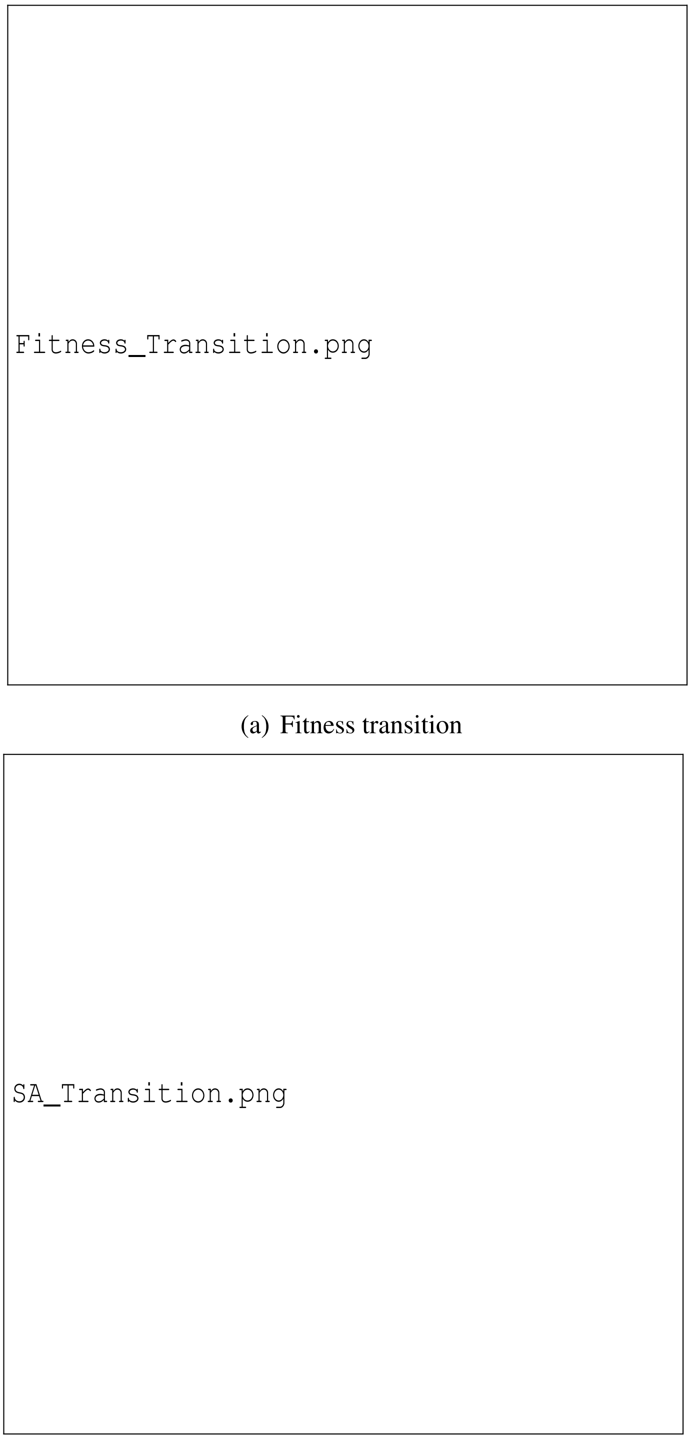
Table 3.4 which provides the p values from the Wilcoxon test between EWS-FCM with all the comparison methods, shows that EWS-FCM always performs significantly better or worse compared to other methods on SynthDB Gaussian instances. The results show that the proposed EWS-FCM outperforms FCM_S1, FCM_S2, EnFCM, FGFCM_S1, and FGFCM_S2 in all the instances. In comparison to FGFCM, EWS-FCM has a significantly better performance in all but one of the 40 instances. (FGFCM performs better for the instance $Synth2$, $\sigma^2 = 10\%$). In comparison to FLICM, EWS-FCM is significantly better in all instances except three low-level noise instances: $Synth2$, $Synth3$, and $Synth4$ all with $\sigma^2 = 10\%$.



(a) Histogram of the noiseless image.



(b) Histogram of the noisy image.



(a) Fitness transition

(b) SA transition

Figure 3.8: Transition of fitness and SA over the iterations.

To compare WS-FCM and EWS-FCM with each other, another Wilcoxon test is conducted on all the instances using SA values from these two methods. The p values resulted from this test are provided in Table 3.5. The bold numbers in this table indicates the instances where the methods are not performing significantly different (the p value is greater than the significance level). **+** indicates significantly better performance of EWS-FCM, **-** indicates better performance of WS-FCM, and **=** indicates not significantly different performances of the two algorithms. There are an overall 10 instances that WS-FCM and EWS-FCM are performing not significantly different. For the rest of the 30 instances, EWS-FCM performs significantly better in 26 instances against four better performances of WS-FCM.

Standard deviations of the two methods in Table 3.2 show that in 26 (out of 40) instances WS-FCM has a bigger standard deviation over 30 runs compared to EWS-FCM. This shows better consistency of EWS-FCM performance on each instance. There is this interesting instance ($Synth4$, $\sigma^2 = 80\%$) for which the standard deviation of WS-FCM is bigger than the significance level. This has caused WS-FCM to perform not significantly different from FLICM, but having this stabilized, EWS-FCM performs better than FLICM.

To determine the overall best performers for each instance, we sort the segmentation accuracy of all the algorithms for that instance. Here, we mention the first-three best performers. EWS-FCM is the overall best performer by performing the best in 26 (out of 40) instances and the second-best in three further instances. FLICM is the overall second-best performer by performing the best in three instances and none second-best. WS-FCM is the overall third-best performer being the best performer in one instance and the second-best in 27 further instances. There are 10 further instances that the best performance is shared between WS-FCM and EWS-FCM as they perform not significantly different.

To compare the performance of all methods over the noise level variation on all the images in the SynthDB, two more charts are provided in Fig. 3.9. Fig. 3.9(a) shows the variance of SA over the eight noisy instances for each image in the dataset. The red and magenta bars belong to the performances of the proposed WS-FCM and EWS-FCM algorithms respectively. This figure shows that the smallest SA variances always belongs to WS-FCM and EWS-FCM. Sometimes it is so small that it is almost negligible compared to the SA variances of other methods. Also, Fig. 3.9(b) shows the mean of SA

Table 3.2: SA values for the SynthDB Gaussian instances. The bold numbers indicate the best overall performance for each instance where the difference is significant.

		Algorithm								
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**EWS-FCM**
Synth1	10%	88.98	89.86	88.98	93.12	94.33	93.01	95.90	97.06± 0.03	**97.24±0.03**
	20%	79.41	82.85	78.95	86.22	89.66	87.22	88.04	95.94± 0.05	**96.09±0.02**
	30%	71.62	78.14	71.44	79.86	85.99	81.37	58.15	95.46± 0.09	**95.58±0.05**
	40%	67.16	74.94	67.10	75.31	82.15	77.10	44.95	94.28± 0.10	94.30±0.06
	50%	62.77	71.79	63.34	71.38	78.93	72.83	43.12	94.07± 0.07	**94.09±0.07**
	60%	59.78	70.42	60.64	67.94	76.36	72.67	29.27	94.17± 0.10	94.19±0.12
	70%	57.25	68.61	58.12	64.82	73.33	70.26	44.69	93.19± 0.14	**93.30±0.12**
	80%	55.29	66.27	56.10	62.87	69.89	67.67	30.49	92.70± 0.04	92.68±0.06
Synth2	10%	97.85	98.41	97.89	99.11	99.02	98.73	**99.18**	99.12± 0.01	99.07±0.01
	20%	92.26	95.05	92.42	96.51	97.85	96.98	97.35	98.67± 0.03	**98.70±0.03**
	30%	87.25	91.49	87.72	93.58	96.51	94.87	96.28	98.36± 0.03	**98.39±0.00**
	40%	81.72	89.21	82.85	89.11	93.49	92.02	74.52	98.13± 0.01	**98.19±0.01**
	50%	76.60	86.52	78.22	85.39	91.32	89.71	82.32	97.64± 0.05	**97.66±0.02**
	60%	74.44	85.53	76.40	83.65	90.05	88.70	74.51	97.38± 0.01	**97.42±0.01**
	70%	71.04	83.49	73.47	80.16	86.40	86.54	73.24	**97.68± 0.03**	97.65±0.01
	80%	68.52	81.58	71.02	77.76	85.56	85.59	73.89	97.18± 0.02	**97.25±0.04**
Synth3	10%	81.10	93.30	96.04	97.68	97.25	97.15	**99.65**	99.61± 0.00	99.61±0.00
	20%	67.89	76.50	69.33	88.37	95.48	94.37	99.19	99.33± 0.01	99.33±0.00
	30%	52.89	71.85	64.92	76.71	90.76	88.78	98.63	98.92± 0.04	**98.93±0.01**
	40%	45.92	67.07	60.81	72.89	86.83	84.82	65.52	98.73± 0.02	**98.78±0.02**
	50%	41.31	62.38	57.00	69.78	80.65	78.62	55.60	98.54± 0.06	**98.64±0.06**
	60%	38.72	61.55	54.03	68.78	80.23	77.18	54.36	98.56± 0.02	**98.59±0.02**
	70%	36.99	59.21	51.58	63.26	78.07	72.38	51.58	97.79± 0.03	97.81±0.03
	80%	34.49	58.54	49.69	62.40	75.42	70.59	45.22	97.96± 0.08	**98.02±0.01**
Synth4	10%	72.59	77.48	85.28	93.80	92.25	93.70	**98.16**	96.49± 0.01	96.82±0.01
	20%	54.42	67.71	72.10	72.65	84.55	82.02	92.41	95.16± 0.08	**95.55±0.02**
	30%	51.00	60.31	66.59	65.40	65.40	75.39	82.16	93.93± 0.04	**94.38±0.05**
	40%	49.36	58.30	61.76	62.01	60.43	76.61	81.68	93.21± 0.03	**93.66±0.03**
	50%	48.69	55.94	53.88	57.90	57.28	72.07	81.86	91.89± 0.04	**92.10±0.04**
	60%	48.82	54.17	50.62	56.55	56.25	70.21	88.39	90.13± 0.32	**90.67±0.22**
	70%	48.04	54.01	48.08	51.60	54.57	68.73	88.55	90.99± 0.15	**91.17±0.21**
	80%	47.76	52.40	47.51	47.01	50.01	68.26	85.70	81.70±10.08	87.87±0.35
Synth5	10%	85.62	86.90	85.54	90.31	93.15	90.68	57.80	96.84± 0.09	**96.92±0.01**
	20%	75.55	79.95	76.33	82.99	88.01	84.38	58.73	95.06± 0.07	**95.27±0.06**
	30%	67.21	74.05	69.03	76.47	82.39	78.03	58.25	93.96± 0.11	93.93±0.10
	40%	62.39	70.96	64.81	72.43	78.87	74.89	59.96	93.09± 0.06	**93.21±0.04**
	50%	57.20	66.68	60.11	67.50	73.84	70.19	58.19	90.54± 0.13	90.51±0.16
	60%	54.73	65.21	57.89	65.19	71.62	68.06	39.66	90.98± 0.15	91.00±0.13
	70%	52.90	63.78	56.20	63.36	69.29	66.56	58.90	89.99± 0.06	**90.18±0.06**
	80%	51.73	62.83	55.17	62.18	68.06	65.67	39.33	88.04± 0.12	88.04±0.13

Table 3.3: Wilcoxon provided p values resulted from the pair-wise comparison of WS-FCM with other methods for the SynthDB Gaussian instances. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively shows where WS-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm						
Img.	**Vol.**	**FCM_S1**	**FCMS2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**
Synth1	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	50%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)
	60%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth2	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.5e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	70%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	80%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)
Synth3	10%	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (-)
	20%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	40%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)
	50%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)
	60%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
Synth4	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00011 (+)	**0.98** (=)
Synth5	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)

Table 3.4: Wilcoxon provided p values resulted from the pair-wise comparison of EWS-FCM with other methods for the SynthDB Gaussian instances. The bold numbers indicate p values greater than the significance level. +, - and = respectively shows where EWS-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm						
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**
Synth1	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth2	10%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth3	10%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)
	20%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)
	30%	1.1e-06 (+)	1.1e-06 (+)	1.1e-06 (+)	1.1e-06 (+)	1.1e-06 (+)	1.1e-06 (+)	1.1e-06 (+)
	40%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
Synth4	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth5	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)

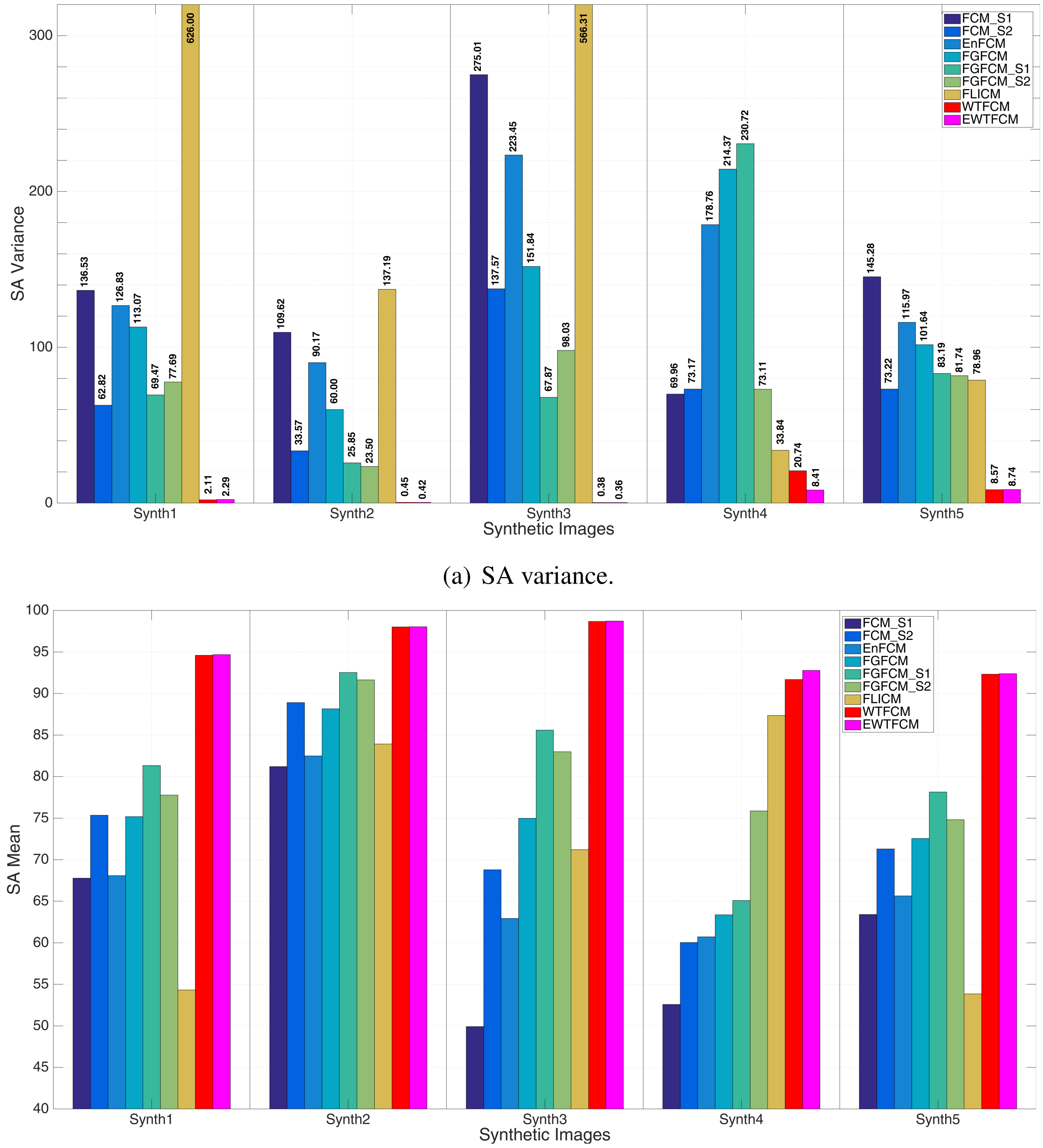
Table 3.5: Wilcoxon provided *p* values resulted from the pair-wise comparison of WS-FCM with EWS-FCM for the SynthDB Gaussian instances. The bold numbers indicate *p* values greater than the significance level. **+**, **-** and **=** respectively shows where EWS-FCM performs significantly better, significantly worse or not significantly different compared to WS-FCM.

	Noise Level							
Img.	10%	20%	30%	40%	50%	60%	70%	80%
Synth1	1.7e-06 (+)	1.7e-06 (+)	2.6e-05 (+)	**0.48** (=)	0.048 (+)	**0.91** (=)	0.0036 (+)	**0.063** (=)
Synth2	1.7e-06 (-)	0.00049 (+)	1.7e-06 (+)	1.7e-06 (+)	0.024 (+)	1.7e-06 (+)	0.002 (-)	5.2e-05 (+)
Synth3	0.022 (+)	**0.15** (=)	0.014 (+)	1.7e-06 (+)	4.8e-05 (+)	5.6e-06 (+)	**0.11** (=)	1.7e-06 (+)
Synth4	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	5.7e-06 (+)	0.0024 (+)	**0.5** (=)
Synth5	1.7e-06 (+)	1.7e-06 (+)	**0.27** (=)	1.9e-06 (+)	**0.63** (=)	**0.7** (=)	1.7e-06 (+)	**0.64** (=)

values over different noise volumes. The two highest mean performances always belong to WS-FCM and EWS-FCM, and EWS-FCM is always slightly better than WS-FCM. Overall, the smallest variances and the highest mean performances indicates the superior consistency of the proposed WS-FCM and EWS-FCM compared to other methods on the SynthDB Gaussian instances.

Gaussian Noise (Qualitative Analysis)

Figure 3.10 provides a sample Gaussian instance for each image in the SynthDB, and the segmentation results for all the methods. The instances are $Synth1,\sigma^2 = 80\%$, $Synth2,\sigma^2 = 50\%$, $Synth3,\sigma^2 = 60\%$, $Synth4,\sigma^2 = 40\%$, and $Synth5,\sigma^2 = 30\%$. Visual comparison clearly shows the better performance of WS-FCM and EWS-FCM compared to other methods. The only methods that can segment all the existing coherent regions in these instances are WS-FCM and EWS-FCM. Other methods produce lots of small redundant regions due to the volume of noise. FLICM can segment some of the main regions, however, one general problem with FLICM, as depicted in Fig. 3.10, is that it completely loses one or two clusters during the segmentation procedure. This means that FLICM cannot comply with the required number of regions determined in advance. For example, for instances $Synth1,\sigma^2 = 80\%$ and $Synth5,\sigma^2 = 30\%$ the predefined number of regions are four and three respectively, but the number of provided regions are two in both cases. Although EWS-FCM produces more redundant regions (see $Synth5,\sigma^2 = 30\%$ for an example) or bigger redundant regions (see $Synth3,\sigma^2 = 60\%$ as an example), having it perform better around the edges leads to mostly better perfor-



(a) SA variance.

(b) SA mean.

Figure 3.9: Demonstration of different algorithms performance on noise level variation on the SynthDB Gaussian instances.

mance of EWS-FCM compared to WS-FCM.

Salt & Pepper Noise (Quantitative Analysis)

Table 3.6 provides the SA values for all the SynthDB Salt & Pepper instances resulted from all the algorithms. Also, Wilcoxon test results from the pair-wise comparison of WS-FCM and EWS-FCM with other methods are provided in Tables 3.7 and 3.8 respectively. There is one p value bigger than the significance level in Table 3.7. This value belongs to the instance $S2, density = 80\%$ when compared with FGFCM_S2. Also, there are two p values in Table 3.8 bigger than the significance level belonging to the instances $S2, density = 70\%$ and $S2, density = 80\%$ when both compared with FGFCM_S2. Other than these instances, WS-FCM and EWS-FCM are always performing significantly different than other methods.

A pair-wise comparison of SA values from Table 3.6 shows that WS-FCM is always performing better than FCM_S1, EnFCM, and FGFCM_S1. For the other methods, WS-FCM performs better in 39, 36, and 27 instances (out of 40) than FCM_S2, FGFCM, and FLICM respectively. There is only FGFCM_S2 that performs mostly better than our proposed WS-FCM by performing better in 30 instances. Our WS-FCM holds nine better performances, and there is one instance that WS-FCM and FGFCM_S2 perform not significantly different.

The same analysis for EWS-FCM shows that our proposed EWS-FCM is performing significantly better than FCM_S1, EnFCM, and FGFCM_S1 in all the instances. Also, EWS-FCM performs better in 39, 38, and 27 instances (out of 40) when compared to FCM_S2, FGFCM, and FLICM respectively. Here again FGFCM_S2 is performing overall better by holding 30 against eight better performances of EWS-FCM. There are two instances that EWS-FCM and FGFCM_S2 perform not significantly different.

For the comparison of WS-FCM and EWS-FCM on SynthDB Salt & Pepper instances, Wilcoxon test results are provided in Table 3.9. There are two instances that WS-FCM and EWS-FCM performs not significantly different: $Synth1, density = 60\%$ and $Synth3, density = 70\%$. Overall, there are 24 instances that EWS-FCM performs better, 14 instances that WS-FCM performs better, and two instances that they do, not significantly different. Table 3.9 shows that for the high levels of noise, $density = 70\%$ and 80%, it is WS-FCM that performs mostly better, and for others EWS-FCM performs

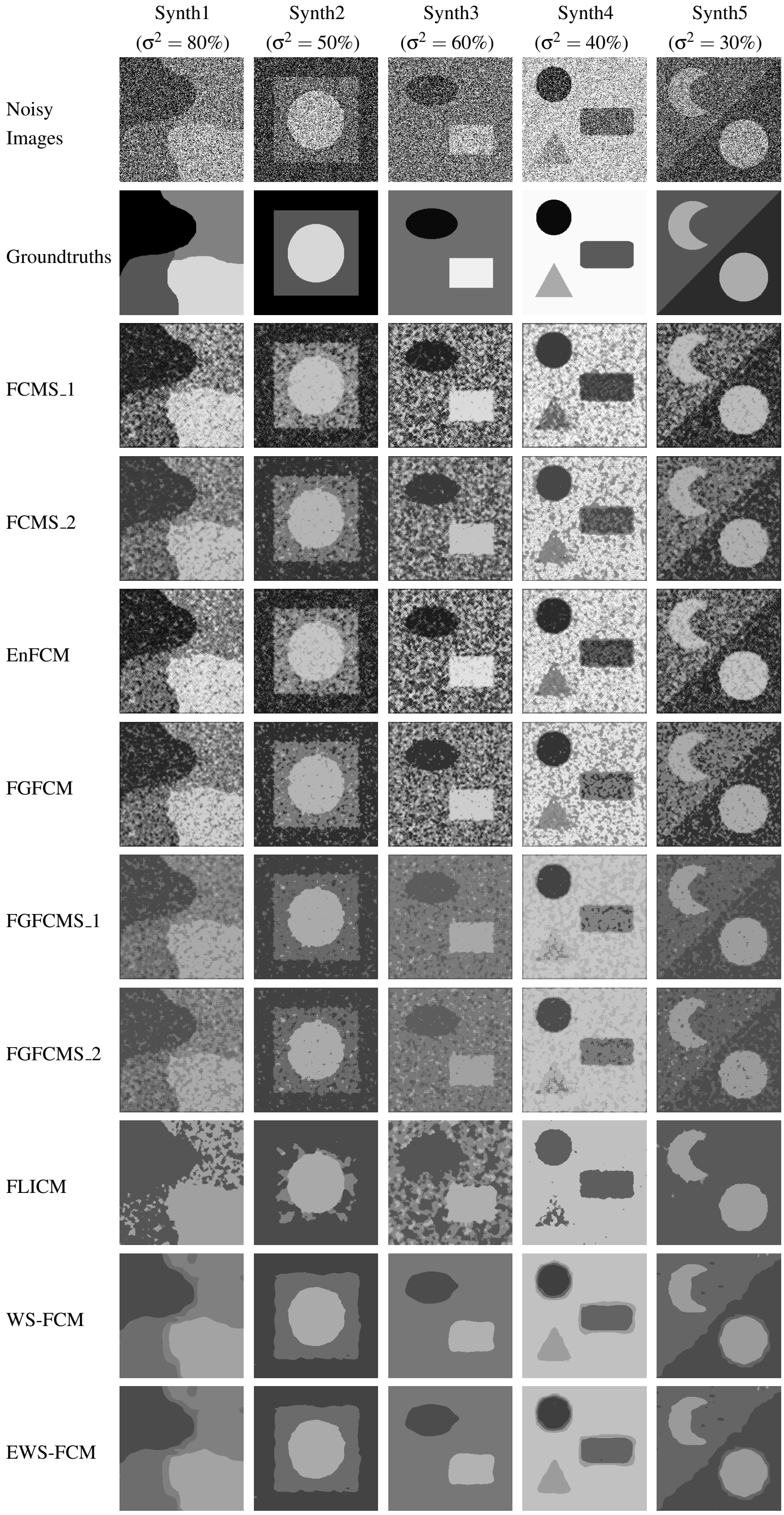


Figure 3.10: Qualitative comparison of the proposed WS-FCM and EWS-FCM with FCM_S1, FCM_S2, EnFCM, FGFCM_S1, FGFCM_S2, FGFCM, and FLICM on some Gaussian instances of SynthDB. Noise variance for Synth1, Synth2, Synth3, Synth4, and Synth5 instances is 80%, 50%, 60%, 40%, and 30% respectively.

mostly better. In this Table, **+** indicates significantly better performance of EWS-FCM, **-** indicates better performance of WS-FCM, and **=** indicates not significantly different performances of the two algorithms.

Considering the best performance on each instance, the overall four-best performers are as follows. FGFCM_S2 is the best performer being the best performer in 23 instances, and second-best performer in five more instances. FLICM is the second-best performer acting the best on seven instances and second-best on six more instances. The third-best performer in this regard is WS-FCM, by being the best performer on four instances, and second-best on seven more instances. EWS-FCM is the fourth-best performer being the best and second-best performer in three and 16 instances respectively.

For the performance evaluation of all methods on Salt & Pepper noise level variation, Fig. 3.11(a) and 3.11(b) are provided for SA variance and SA mean respectively. Fig. 3.11(a) shows that for three out of five images in the SynthDB known as $Synth1$, $Synth2$, and $Synth3$, the proposed WS-FCM and EWS-FCM have the smallest variance when the noise volume changes. For $Synth4$ and $Synth5$ the FLICM and FGFCM_S2 have the two smallest SA variance. Also, Fig. 3.11(b) shows that for $Synth1$ and $Synth3$, WS-FCM and EWS-FCM have the highest mean SA, for $Synth2$ and $Synth5$ WS-FCM/EWS-FCM have the second/third-biggest mean SA after FGFCM_S2, and for $Synth4$, the two biggest mean SA values belong to FGFCM_S2 and FLICM, and WS-FCM and EWS-FCM as third and fourth biggest mean performance. Unlike the Gaussian noise, here not always EWS-FCM has a bigger mean SA than WS-FCM.

Salt & Pepper Noise (Qualitative Analysis)

Figure 3.12 provides a Salt & Pepper instance for each image in the SynthDB and the segmentation results from all the methods. The instances are $Synth1, density = 10\%$, $Synth2, density = 60\%$, $Synth3, density = 80\%$, $Synth4, density = 50\%$, and $Synth5, density = 70\%$. Visual comparison shows that WS-FCM and EWS-FCM are the only methods that can segment the coherent regions based on the predefined number of clusters without producing any redundant regions. FGFCM_S2 mostly produces the most accurate results by preserving the details, however, the number of produced redundant regions are much more in this method compared to WS-FCM and EWS-FCM. On the other hand, WS-FCM and EWS-FCM over-smooth the boundaries that causes these methods not to

Table 3.6: SA values for the SynthDB Salt & Pepper instances. The bold number indicates the best performance for each instance. The bold numbers indicate the best performance for each instance.

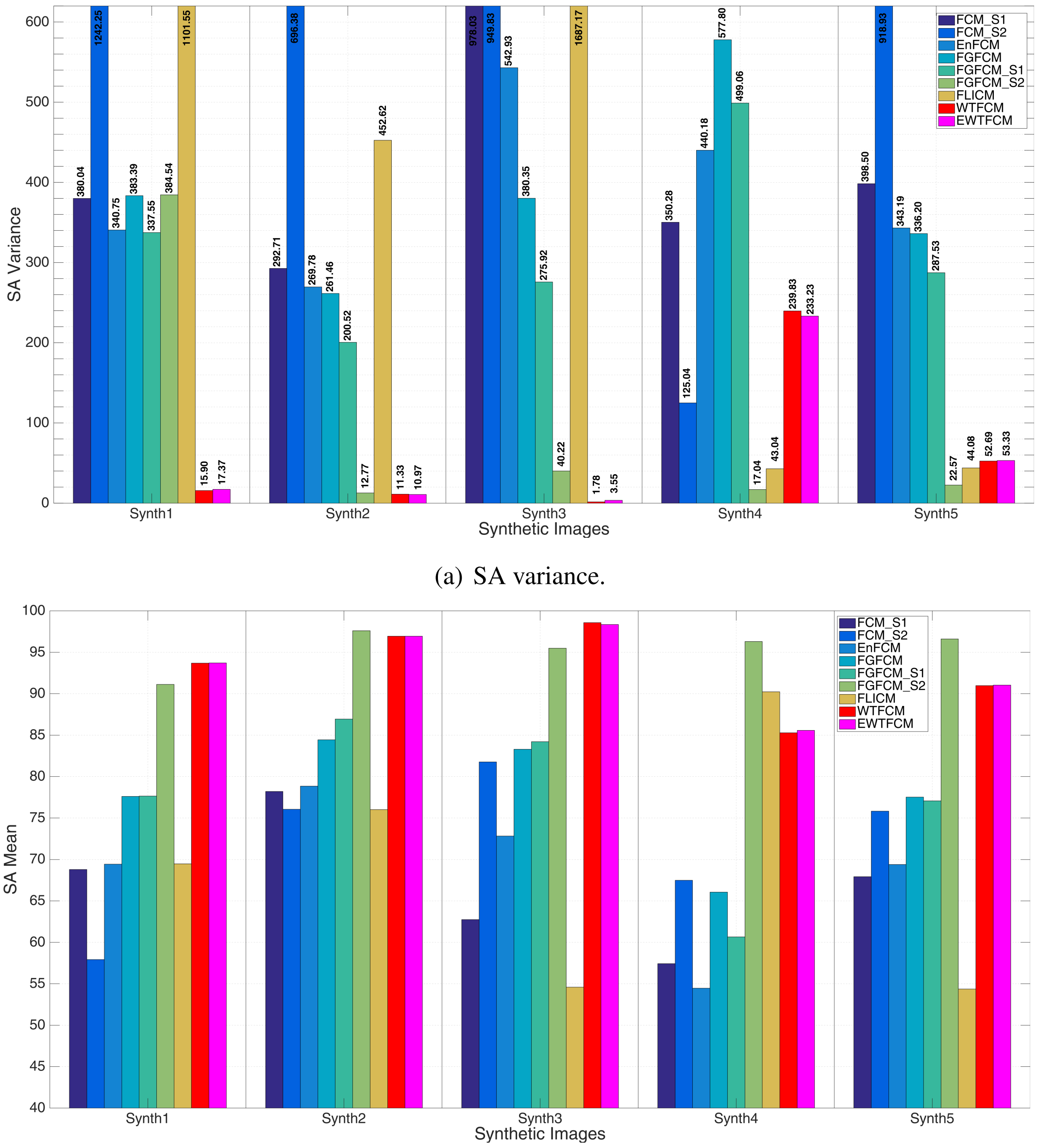
		Algorithm								
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**EWS-FCM**
Synth1	10%	93.15	95.82	92.91	97.88	96.20	99.73	**99.78**	97.84±0.08	98.02±0.08
	20%	88.46	91.77	88.34	96.59	95.15	99.33	**99.75**	96.96±0.11	97.14±0.12
	30%	82.48	87.64	82.11	92.70	91.11	98.90	**99.72**	96.03±0.00	96.15±0.00
	40%	75.06	83.55	75.05	86.93	87.29	98.39	**99.26**	95.29±0.13	95.52±0.14
	50%	66.65	35.01	66.97	77.47	78.06	**98.05**	52.76	93.91±0.22	93.99±0.20
	60%	57.75	32.80	58.39	67.09	68.06	**97.15**	25.69	92.63±0.47	92.62±0.57
	70%	47.88	30.50	50.54	57.48	59.76	**94.74**	45.18	91.38±0.04	91.08±0.05
	80%	38.96	6.29	41.16	44.71	45.61	42.75	33.59	**85.39±0.31**	85.11±0.30
Synth2	10%	97.44	97.24	97.44	99.86	99.32	99.85	**99.90**	99.62±0.00	99.60±0.00
	20%	94.35	94.33	94.34	99.01	98.74	**99.56**	99.41	99.12±0.00	99.09±0.01
	30%	90.90	91.58	90.94	96.70	97.43	**99.46**	98.83	98.68±0.01	98.66±0.01
	40%	85.79	88.81	85.95	92.67	95.67	**99.15**	70.76	98.25±0.04	98.27±0.05
	50%	78.14	85.83	78.73	86.40	90.13	**99.00**	70.12	97.76±0.05	97.77±0.06
	60%	69.22	82.78	70.20	77.80	80.92	**98.49**	53.68	96.54±0.03	96.47±0.03
	70%	58.86	35.21	61.04	67.15	72.02	96.01	69.91	**96.43±0.57**	96.32±0.65
	80%	50.92	32.70	52.12	55.89	61.33	89.28	45.49	89.09±0.85	89.25±0.99
Synth3	10%	99.08	99.80	99.33	99.85	98.23	99.80	**99.95**	99.79±0.01	99.77±0.01
	20%	96.95	99.21	97.74	99.33	97.41	99.21	16.69	99.51±0.01	**99.51±0.01**
	30%	89.54	98.53	93.93	98.59	96.61	98.79	91.71	99.46±0.01	**99.50±0.02**
	40%	73.43	97.88	80.64	96.72	94.95	98.05	16.42	**99.23±0.02**	99.22±0.02
	50%	53.37	97.46	65.51	88.37	88.63	97.69	91.70	**99.05±0.02**	99.02±0.02
	60%	39.66	72.38	56.65	70.94	77.76	96.69	16.43	98.32±0.03	**98.33±0.03**
	70%	30.06	10.11	48.69	62.54	65.62	92.92	88.23	97.34±0.02	97.34±0.02
	80%	19.92	78.75	40.07	50.01	54.47	80.70	15.65	**95.92±0.52**	94.12±0.83
Synth4	10%	89.78	81.70	89.32	94.95	92.89	**99.73**	99.67	96.88±0.01	97.09±0.01
	20%	85.07	76.93	84.64	91.93	90.36	99.25	**99.55**	95.64±0.04	95.99±0.03
	30%	52.68	72.84	52.73	84.02	69.47	**98.61**	89.34	94.34±0.04	94.67±0.04
	40%	47.27	73.15	46.24	74.31	60.53	**98.00**	89.28	92.74±0.03	93.24±0.03
	50%	45.63	67.57	47.71	63.41	53.18	**97.19**	86.22	87.12±0.10	87.37±0.10
	60%	47.41	62.75	43.91	46.60	46.22	**96.26**	88.95	86.59±0.04	86.77±0.20
	70%	47.58	57.44	38.65	40.22	40.79	**94.34**	88.89	79.20±0.15	78.48±0.17
	80%	44.05	47.49	32.60	32.95	31.76	**87.03**	80.03	49.82±2.33	51.01±2.44
Synth5	10%	94.39	97.70	94.34	97.83	95.89	**99.74**	57.89	97.81±0.02	97.89±0.02
	20%	88.85	95.36	89.09	95.61	93.65	**99.39**	57.61	96.73±0.08	96.84±0.08
	30%	81.75	93.24	82.02	91.86	89.78	**98.95**	57.43	95.82±0.09	95.85±0.09
	40%	72.91	91.03	73.27	85.28	83.82	**98.69**	38.23	93.88±0.12	93.97±0.09
	50%	63.24	88.60	64.82	75.57	75.54	**98.04**	57.08	91.02±0.19	91.06±0.17
	60%	56.38	86.63	58.56	68.02	69.76	**97.49**	56.97	90.32±0.02	90.54±0.01
	70%	46.99	26.92	50.54	58.28	59.75	**95.20**	55.62	86.64±0.05	86.61±0.04
	80%	38.91	27.19	42.49	47.80	48.42	**85.39**	54.13	75.56±0.32	75.51±0.37

Table 3.7: Wilcoxon provided p values resulted from the pair-wise comparison of WS-FCM with other methods for the SynthDB Salt & Pepper instances. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively show where WS-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm						
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**
Synth1	10%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (-)	1.5e-06 (+)	1.5e-06 (-)	1.5e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth2	10%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	20%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (-)	1.5e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.6e-05 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.34** (=)	1.7e-06 (+)
Synth3	10%	1.1e-06 (+)	1.1e-06 (-)	1.1e-06 (+)	1.1e-06 (-)	1.1e-06 (+)	1.1e-06 (-)	1.1e-06 (-)
	20%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	40%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	50%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth4	10%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	80%	2.6e-06 (+)	0.00077 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
Synth5	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.0002 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)

Table 3.8: Wilcoxon provided p values resulted from the pair-wise comparison of EWS-FCM with other methods for the SynthDB Salt & Pepper instances. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively show where EWS-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm						
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**
Synth1	10%	1e-06 (+)	1e-06 (+)	1e-06 (+)	2.2e-05 (+)	1e-06 (+)	1e-06 (-)	1e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (-)	1.5e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth2	10%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (-)	1.5e-06 (+)	1.5e-06 (-)	1.5e-06 (-)
	20%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (-)	1.5e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.17** (=)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.38** (=)	1.7e-06 (+)
Synth3	10%	1.1e-06 (+)	1.1e-06 (-)	1.1e-06 (+)	1.1e-06 (-)	1.1e-06 (+)	1.1e-06 (-)	1.1e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)
	50%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	60%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth4	10%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	20%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	80%	1.9e-06 (+)	4.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
Synth5	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)



(a) SA variance.

(b) SA mean.

Figure 3.11: Demonstration of different algorithms performance on noise variation (SynthDB Salt & Pepper instances).

Table 3.9: Wilcoxon provided *p* values resulted from the pair-wise comparison of WS-FCM with EWS-FCM for the SynthDB Salt & Pepper instances. The bold numbers indicate *p* values greater than the significance level. **+**, **-** and **=** respectively show where EWS-FCM performs significantly better, significantly worse or not significantly different than WS-FCM.

	Noise Level							
Img.	10%	20%	30%	40%	50%	60%	70%	80%
Synth1	1.5e-06 (+)	1.7e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.057** (=)	1.7e-06 (-)	1.7e-06 (-)
Synth2	1.4e-06 (-)	1.5e-06 (-)	1e-05 (-)	0.00035 (+)	0.00076 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (+)
Synth3	9.7e-07 (-)	0.00071 (+)	1.7e-06 (+)	1.6e-06 (-)	1.5e-06 (-)	0.0031 (+)	**0.066** (=)	1.7e-06 (-)
Synth4	1.6e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00036 (+)	1.7e-06 (-)	1.7e-06 (+)
Synth5	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.2e-06 (+)	1.7e-06 (+)	2.5e-06 (-)	0.00018 (-)

performer as good as the Gaussian instances. FLICM can segment the coherent regions on some of the instances, but not according to the required number of clusters. This causes some of the segmented regions to have the same gray values as the intensity representative of the regions.

3.5.3 BerkDB

Gaussian Noise (Quantitative Analysis)

The results of Wilcoxon test from the pair-wise comparison of WS-FCM and EWS-FCM with other methods are presented in Tables 3.11 and 3.12 respectively. Table 3.11 reveals that there is not any case that the *p* value is bigger than the significance level. Therefore, WS-FCM always performs significantly different compared to other comparison methods. To determine which one is the case, we have a look at SA values provided in Table 3.10. The analysis of results from this table shows that WS-FCM always performs significantly better than FCM_S1, FCM_S2, EnFCM, and FGFCM in all the 40 instances. It also performs significantly better in 39, 38, and 28 instances compared to FGFCM_S1, FGFCM_S2, and FLICM. Therefore, WS-FCM is always or mostly better on BerkDB Gaussian instances.

The performance of WS-FCM becomes even better in EWS-FCM. Table 3.12 shows that there is one single *p* value bigger than the significance level. This is for the instance $B42049, \sigma^2 = 10\%$ when EWS-FCM is compared with FLICM. Other than is instance, EWS-FCM always performs significantly different than other methods. Analysis of SA

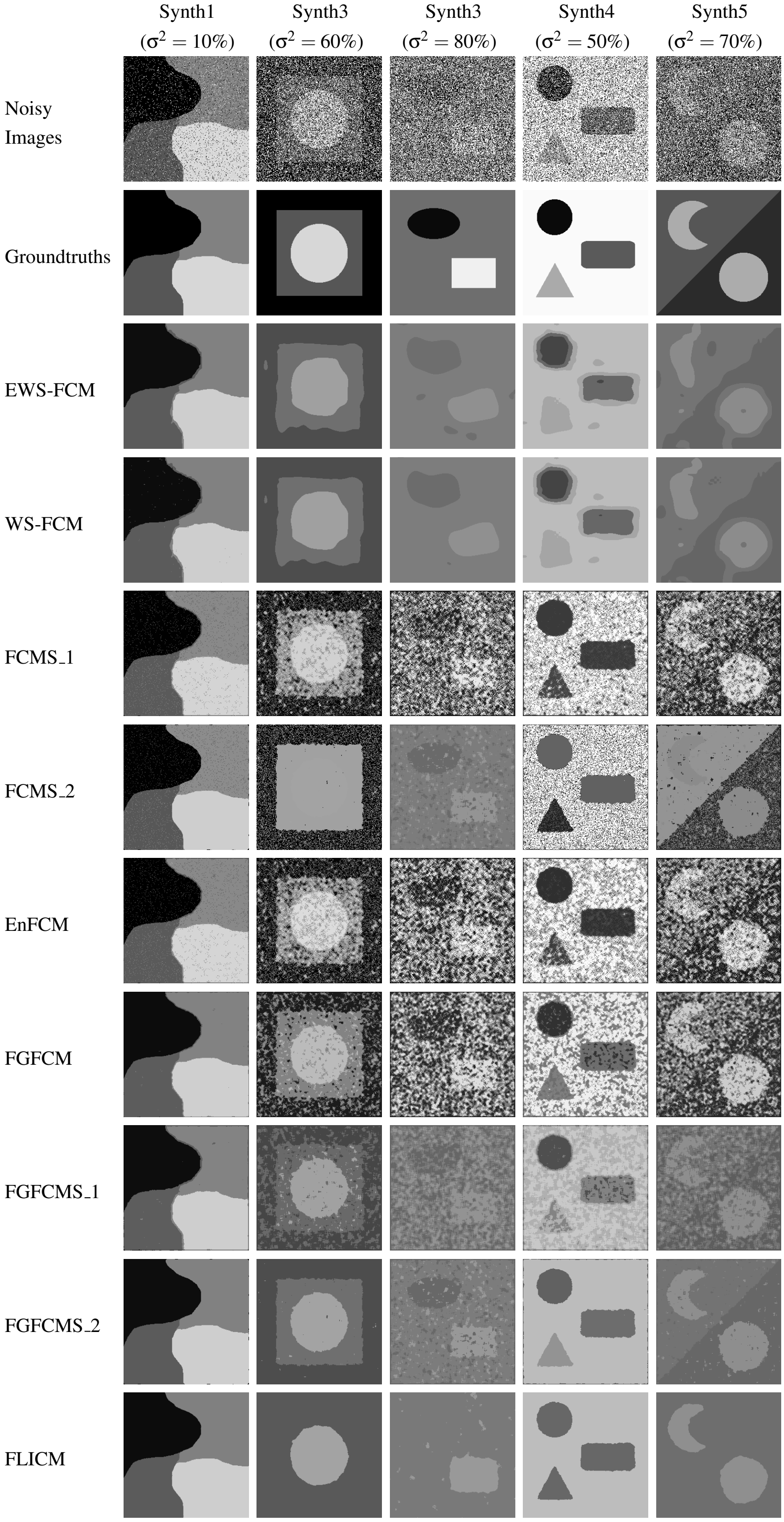


Figure 3.12: Qualitative comparison of the proposed WS-FCM and EWS-FCM with FCM_S1, FCM_S2, EnFCM, FGFCM_S1, FGFCM_S2, FGFCM, and FLICM on some SynthDB Salt & Pepper instances. Noise densities for Synth1, Synth2, Synth3, Synth4, and Synth5 instances are 10%, 60%, 80%, 50%, and 70% respectively.

values in Table 3.10 shows that EWS-FCM always outperforms FCM_S1, FCM_S2, EnFCM, FGFCM, and FGFCM_S1 in all the 40 instances. It is also significantly better than FGFCM_S2 and FLICM on 39 and 30 instances respectively. Overall, the number of significantly better performances of other methods in a pair-wise manner has dropped from 15 in WS-FCM to 11 in EWS-FCM.

Wilcoxon test results on the comparison of WS-FCM and EWS-FCM is provided in Table 3.13. There is no p value bigger than the significance level in this table. The Table shows that EWS-FCM is significantly better than WS-FCM in 25 instances, and WS-FCM is significantly better than EWS-FCM in 15 instances. This concludes that EWS-FCM again performs mostly better than WS-FCM on BerkDB Gaussian instances.

Considering the best and second-best performance on each instance, the overall first three best performers are determined by another set of analysis. EWS-FCM is the overall best performer by being the best and second-best performer in 15, and 22 instances respectively. WS-FCM is the overall second-best performer, by possessing 14 best and second-best performances, and FLICM is the overall third-best performer having 10 best, and one second-best performance(s).

The performances of all methods on noise variation are compared in Fig. 3.13. Considering the variance of SA metric (Fig. 3.13(a)), if we overlook the unusual behavior of FLICM on image $B3096$ with a very low mean SA, WS-FCM is always possessing the smallest SA variation except for image $B167062$ where FLICM has the lowest SA variance with a slight difference to WS-FCM and EWS-FCM. EWS-FCM has the second-lowest SA variance on images $B42049$, $B167062$, $B86016$, and $B196027$. For the mean SA value (3.13(b)), the proposed WS-FCM and EWS-FCM always have the highest mean performance except for image $B167062$ in which FLICM possess the highest mean performance. On three images WS-FCM has a slightly better mean performance than EWS-FCM, and for the two rest, EWS-FCM performs slightly better.

Gaussian Noise (Qualitative Analysis)

Figure 3.14 provides a sample Gaussian instance for each image in the BerkDB and the corresponding segmentation results of all the methods. The instances are $B3096, \sigma^2 = 40\%$, $B42049, \sigma^2 = 50\%$, $B86016, \sigma^2 = 60\%$, $B167062, \sigma^2 = 80\%$, and $B196027, \sigma^2 = 30\%$. This figure shows that the only method that produces coherent segmented com-

Table 3.10: SA values for the BerkDB Gaussian instances. The bold number indicates the best performance for each instance. The bold numbers indicate the best overall performance for each instance where the difference is significant.

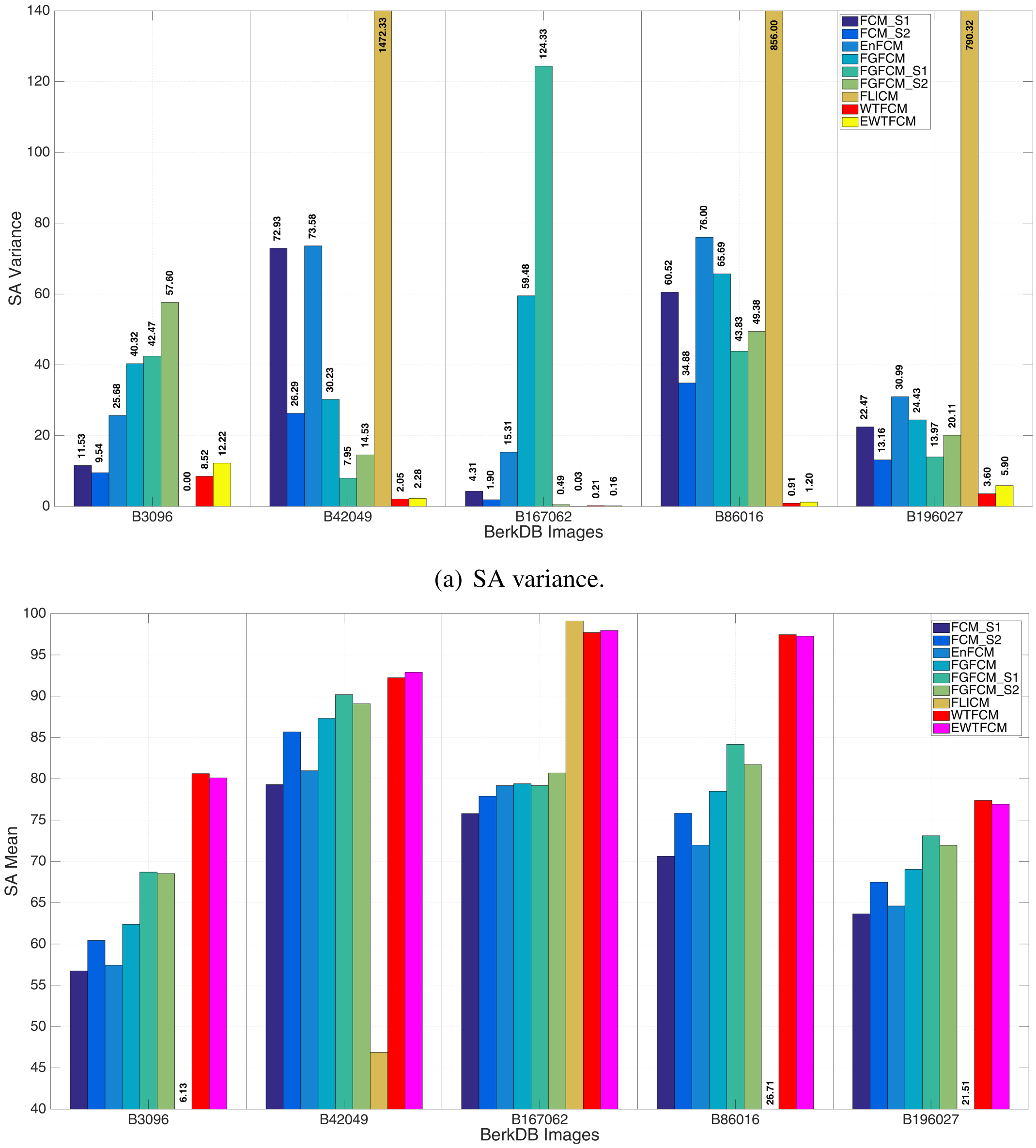
		Algorithm								
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**EWS-FCM**
B3096	10%	64.18	67.18	68.78	76.26	82.71	84.41	6.13	83.82±0.10	**84.55±0.11**
	20%	58.73	62.27	59.99	65.92	72.80	74.09	6.13	**79.88±0.01**	78.14±0.02
	30%	57.00	60.56	57.54	62.73	69.95	69.36	6.13	**76.67±0.03**	74.45±1.73
	40%	55.98	59.85	56.24	61.52	67.04	67.70	6.13	82.67±0.03	**82.77±0.02**
	50%	54.96	58.69	54.72	59.88	65.35	63.74	6.13	**84.33±0.01**	83.98±0.00
	60%	54.72	58.55	54.61	58.14	64.50	63.61	6.13	76.72±0.04	**77.45±0.03**
	70%	54.45	58.53	54.12	57.79	64.24	63.45	6.13	**80.44±0.07**	80.29±0.16
	80%	53.85	57.73	53.35	56.81	63.13	61.76	6.13	**80.48±0.04**	79.31±0.05
B42049	10%	93.07	93.43	93.46	94.12	93.62	93.80	95.21	94.65±0.02	95.22±0.02
	20%	89.00	91.02	90.77	92.86	92.70	92.47	94.16	93.35±0.09	**94.23±0.07**
	30%	83.74	88.65	86.63	91.37	92.05	91.43	90.16	92.75±0.08	**93.29±1.24**
	40%	78.72	86.03	81.48	88.65	91.12	90.18	19.09	92.27±0.04	**93.13±0.04**
	50%	75.59	84.42	77.67	86.91	90.21	89.21	19.09	91.97±1.37	**92.76±1.32**
	60%	73.31	82.63	74.97	84.17	88.95	87.88	19.09	91.17±0.11	**91.86±0.00**
	70%	71.37	80.71	72.17	81.81	87.78	85.65	19.09	91.93±0.14	**92.54±0.20**
	80%	69.59	78.55	70.66	78.60	85.10	82.15	19.09	89.84±0.23	**90.18±1.41**
B167062	10%	79.13	79.77	85.49	98.00	97.27	81.84	**99.06**	98.47±0.01	98.63±0.00
	20%	77.40	79.05	82.38	79.87	96.99	80.22	**99.15**	98.18±0.00	98.35±0.01
	30%	76.64	78.56	81.45	77.91	75.23	80.17	**99.20**	97.84±0.01	98.05±0.00
	40%	76.34	78.54	79.67	77.16	74.34	80.76	**99.37**	97.62±0.01	97.90±0.02
	50%	75.61	77.82	78.59	77.16	73.22	80.61	**99.19**	97.60±0.00	97.86±0.00
	60%	74.67	77.25	76.91	76.09	72.84	80.73	**99.21**	97.56±0.01	97.86±0.00
	70%	74.00	76.78	75.50	74.68	72.52	81.54	**98.94**	97.21±0.01	97.51±0.02
	80%	72.48	75.46	73.47	74.45	70.99	79.74	**98.85**	97.13±0.00	97.43±0.00
B86016	10%	86.19	87.21	89.17	92.69	94.10	93.23	**99.12**	98.47±0.02	98.29±0.02
	20%	77.12	81.05	79.43	86.77	91.36	89.22	16.36	**97.81±0.01**	97.72±0.01
	30%	72.44	77.88	74.32	82.22	87.90	85.65	16.36	**97.43±0.02**	97.13±0.03
	40%	69.06	74.50	70.22	77.56	84.35	81.36	16.36	**98.31±0.04**	98.14±0.06
	50%	67.19	73.21	68.18	74.54	82.43	78.67	16.36	**97.78±0.01**	97.53±2.83
	60%	65.87	72.26	66.18	73.57	80.89	76.95	16.36	97.81±0.00	**98.08±0.01**
	70%	64.06	70.43	64.69	70.91	76.82	74.56	16.36	**96.12±0.15**	95.83±3.05
	80%	63.14	70.11	63.54	69.68	75.53	74.11	16.36	**95.89±0.04**	95.38±0.02
B196027	10%	73.29	74.95	76.09	78.74	79.61	80.24	**91.09**	79.15±0.25	79.62±0.37
	20%	67.30	70.06	68.85	72.99	76.45	75.71	11.57	76.84±0.03	**76.98±0.03**
	30%	64.64	68.19	65.46	70.20	74.40	73.88	11.57	77.78±0.02	**78.45±0.01**
	40%	63.04	66.91	63.80	68.99	73.69	71.80	11.57	**78.54±0.41**	76.34±0.15
	50%	61.68	66.14	61.97	67.30	72.35	70.24	11.57	80.08±0.16	**80.49±0.13**
	60%	60.57	65.28	60.76	65.29	70.37	68.88	11.57	**76.52±0.12**	73.77±0.15
	70%	59.69	64.07	60.36	64.67	68.43	67.26	11.57	74.19±0.06	**74.50±0.07**
	80%	59.12	64.22	59.52	64.17	69.59	67.50	11.57	**75.95±0.04**	75.24±0.04

Table 3.11: Wilcoxon provided p values resulted from the pair-wise comparison of WS-FCM with other methods for the BerkDB Gaussian instances. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively show where WS-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm						
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**
B3096	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B42049	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	3.1e-05 (+)	3.1e-05 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B167062	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	50%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	80%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (-)
B86016	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B196027	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.2e-05 (+)	1.9e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)

Table 3.12: Wilcoxon provided p values resulted from the pair-wise comparison of EWS-FCM with other methods for the BerkDB Gaussian instances. The bold numbers indicate p values greater than the significance level. +, - and = respectively show where EWS-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm						
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**
B3096	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	7.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.9e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B42049	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.068** (=)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)
	30%	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	3.1e-05 (+)	3.1e-05 (+)	3.1e-05 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	3.1e-05 (+)	2.1e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B167062	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	80%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (-)
B86016	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B196027	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.015 (+)	4.7e-06 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)



(a) SA variance.

(b) SA mean.

Figure 3.13: Demonstration of different algorithms performance on noise variation (BerkDB Gaussian instances).

Table 3.13: Wilcoxon provided p values resulted from the pair-wise comparison of WS-FCM with EWS-FCM for the BerkDB Gaussian instances. The bold numbers indicate p values greater than the significance level. +, - and = respectively show where EWS-FCM performs significantly better, significantly worse or not significantly different than WS-FCM.

	Noise Level							
Img.	10%	20%	30%	40%	50%	60%	70%	80%
B3096	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00018 (+)	1.7e-06 (+)
B42049	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	1.7e-06 (+)	3.1e-05 (+)	1.7e-06 (+)	1.7e-06 (+)	4.9e-05 (+)
B167062	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.6e-06 (+)
B86016	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	1.7e-06 (+)	3.1e-05 (+)	1.7e-06 (+)
B196027	3.5e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)

ponents are WS-FCM, EWS-FCM, and FLICM (only in one instance for the latter). Other comparison algorithms except FLICM suffer from over segmentation. FLICM, except for the $B167062$ instance on which it shows very good segmentation accuracy, completely misses the regions. It is clear from the $B42049$ and $B86016$ instances that EWS-FCM performs better on preserving the boundaries.

Salt & Pepper Noise (Quantitative Analysis)

SA value from all the methods are provided in Table 3.14. Wilcoxon test results using these values are provided in Tables 3.15 and 3.16 for the comparison of WS-FCM and EWS-FCM to other methods respectively. There is no p value bigger than the significance level in Table 3.15, and there is only one p value bigger than the significance level in Table 3.16. This value belongs to comparison of EWS-FCM and FGFCM_S2 on instance $196027, density = 70\%$. Therefore, other than this instance, for the rest of the comparisons we refer to SA measure provided in Table 3.14 to determine a significantly better performance. This table shows that WS-FCM performs significantly better in 38 (out of 40) compared to FCM_S1, 30 compared to FCM_S2, 38 compared to EnFCM, 31 compared to FGFCM, and 34 compared to FGFCM_S1. These are the cases that the proposed WS-FCM performs mostly better than other comparison algorithms. As for comparison to FGFCM_S2, WS-FCM performs better in nine instances against 31 better performances of FGFCM_S2. Also, when compared to FLICM, with only four difference, FLICM performs mostly better: WS-FCM on 18 instances and FLICM on 22 instances.

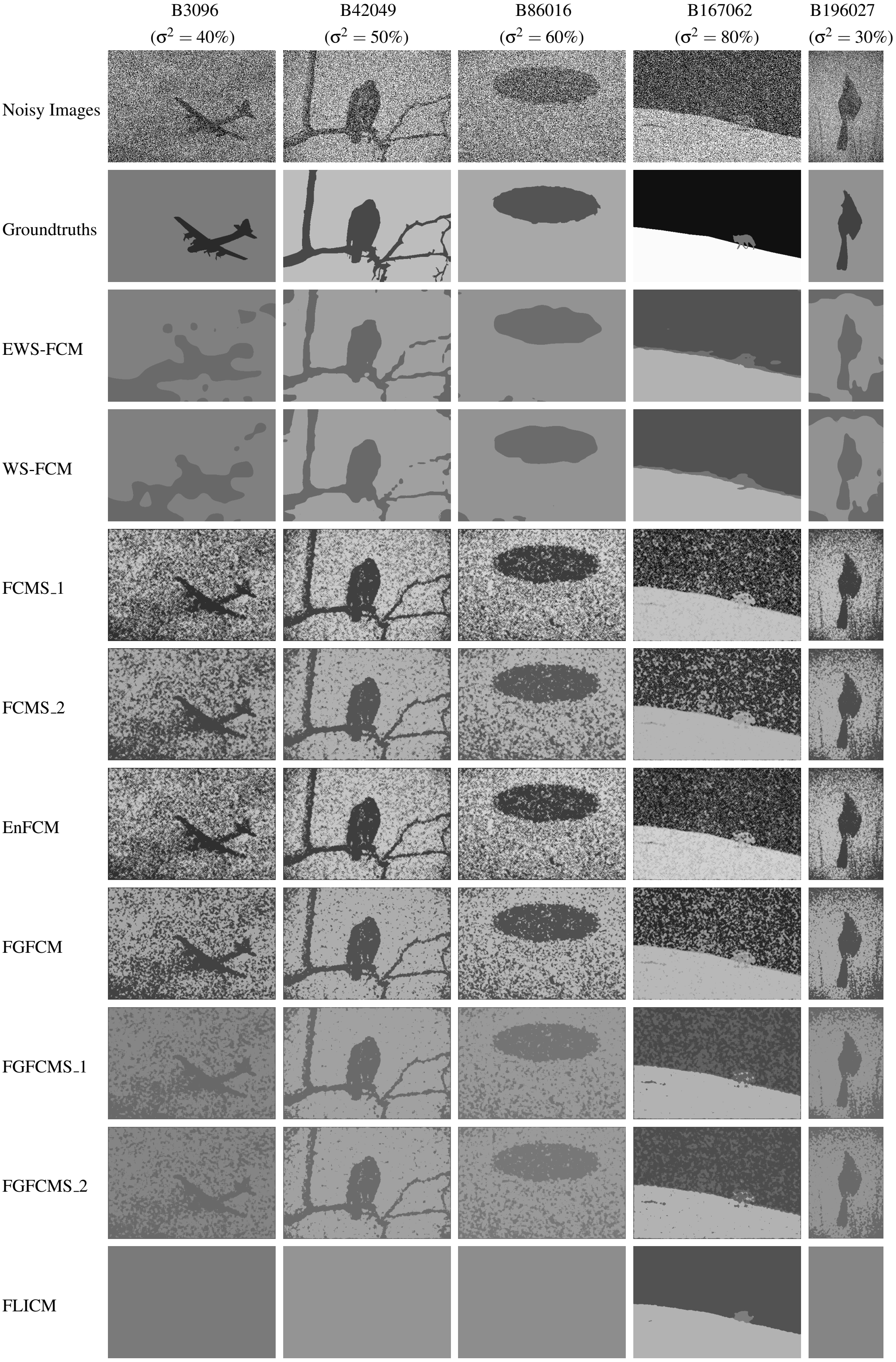


Figure 3.14: Qualitative comparison of the proposed WS-FCM and EWS-FCM with FCM_S1, FCM_S2, EnFCM, FGFCM_S1, FGFCM_S2, FGFCM, and FLICM on some of the BerkDB Gaussian instances. B3096, B42049, B86016, B167062, and B196027 are corrupted with Gaussian noise of variance 40%, 50%, 60%, 80%, and 30% respectively.

In comparison to WS-FCM performance, EWS-FCM performs generally better in the pair-wise comparison to other methods. It is a better performer on 38 instances when compared to FCM_S1, on 32 ones when compared to FCM_S2, on 38 ones when compared to EnFCM, on 35 when compared to FGFCM, and on 35 ones when compared to FGFCM_S1. This clearly shows that the number of better performances has improved compared to WS-FCM. In comparison to FGFCM_S2, the latter performs better on 33 instances against 6 of better performances of EWS-FCM, and one instance of not doing significantly different. Also, in comparison to FLICM, the latter performs better on on 21 instances against 20 of our EWS-FCM. Therefore, with only one difference, FLICM mostly performs better than the proposed EWS-FCM.

For the comparison of WS-FCM and EWS-FCM, p values from the Wilcoxon test are provided in table 3.17. There are two instances that the p value is bigger than the significance level. For the rest, one method significantly outperforms the other. WS-FCM performs better on 20 instances, EWS-FCM on 18 instances, and there are two instances that they do not significantly different. Therefore, with only two better performances, WS-FCM outperforms EWS-FCM in the number of better performances.

For the comparison of performance of all methods on noise level variation, Fig. 3.15 is provided. Fig. 3.15(a) shows SA variance and Fig. 3.15(b) shows SA mean on all the instances of each image. Fig. 3.15(a) indicates that the proposed method has the lowest SA variance on images $B42049$ and $B86016$, and FGFCM_S2 has the lowest SA variance for images $B3096$, $B167062$, and $B196027$. The second-lowest SA variance belongs to EWS-FCM, FLICM, WS-FCM, FLICM, and WS-FCM for $B3096$, $B42049$, $B167062$, $B86016$ and $B196027$ respectively. Also, Fig. 3.15(b) indicates that the proposed WS-FCM and EWS-FCM have the two-highest SA mean on image $B86016$, and for the rest of the images FGFCM_S2 has the highest mean of SA. For these images, one of the WS-FCM and EWS-FCM always possesses the second-highest SA mean after FGFCM_S2.

Salt & Pepper Noise (Qualitative Analysis)

Figure 3.16 provides a sample Salt & Pepper instance for each image in the BerkDB dataset and the corresponding segmentation results of all the methods. The instances are $B3096, density = 70\%, B42049, density = 20\%, B86016, density = 50\%, B167062, density =$

Table 3.14: SA values for the BerkDB Salt & Pepper instances. The bold number indicates the best performance for each instance. The bold numbers indicate the best overall performance for each instance where the difference is significant.

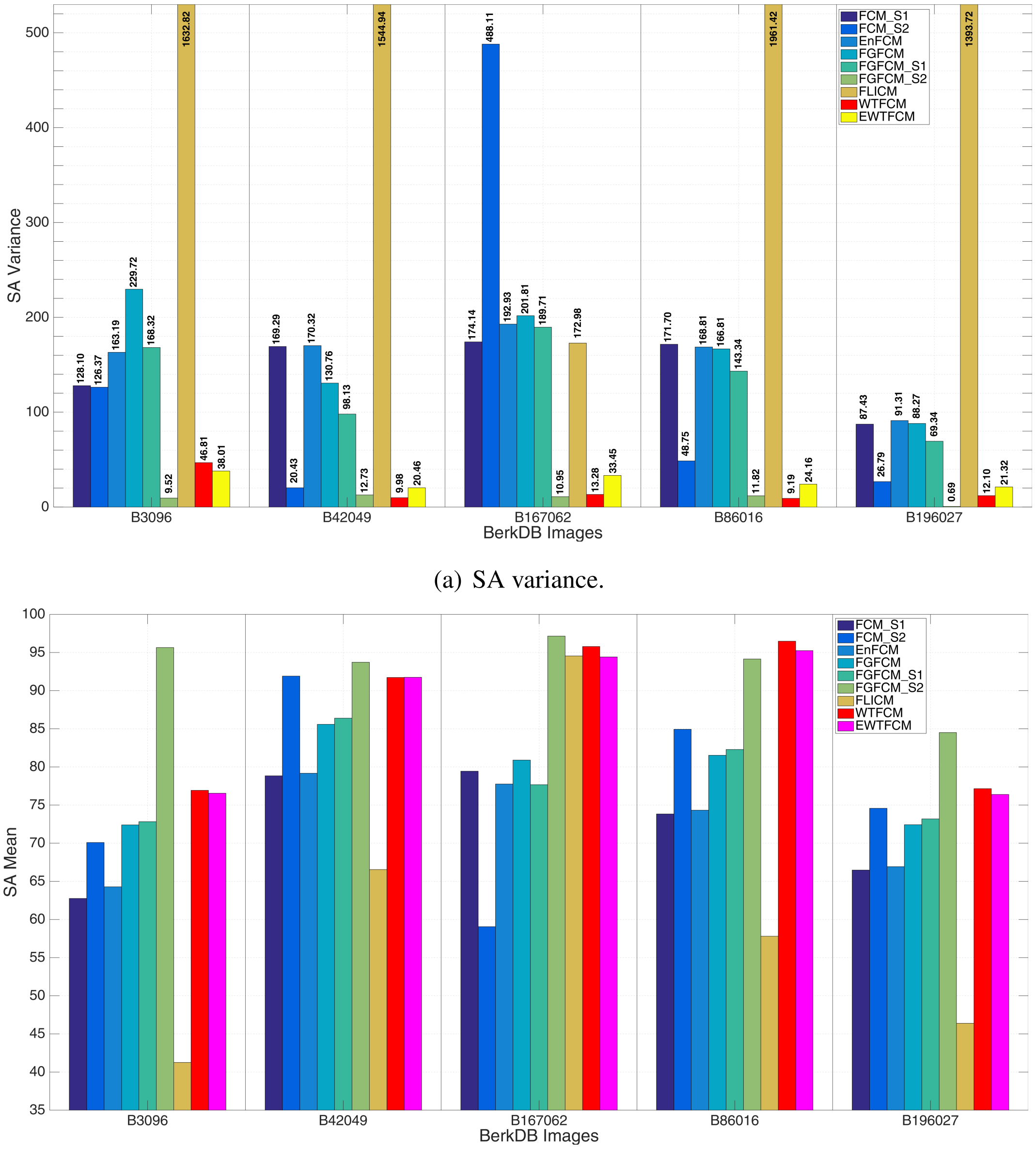
		Algorithm								
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**EWS-FCM**
B3096	10%	83.45	89.60	86.71	93.34	90.98	**97.25**	95.03	81.95±0.01	81.45±0.09
	20%	74.16	80.24	77.20	90.05	88.09	**97.34**	87.23	82.93±0.04	80.63±2.63
	30%	66.93	75.03	70.38	83.25	81.97	**97.38**	79.76	79.05±0.02	83.38±0.04
	40%	60.75	70.66	63.30	73.53	72.76	**97.11**	43.58	81.38±0.02	78.09±2.28
	50%	57.68	66.83	57.67	67.23	68.75	**96.81**	6.13	81.91±0.03	79.73±3.73
	60%	54.84	63.19	55.12	60.86	63.83	**96.29**	6.13	72.81±0.06	73.87±0.15
	70%	52.69	59.10	52.91	56.79	58.76	**94.64**	6.13	71.57±0.03	67.21±1.40
	80%	51.48	56.05	50.97	54.09	57.44	**88.35**	6.13	63.83±0.83	68.08±2.78
B42049	10%	93.90	95.83	94.06	95.08	93.73	**96.23**	95.88	95.11±0.03	95.51±0.01
	20%	92.42	95.18	92.91	94.65	93.49	**96.00**	95.64	94.47±0.01	94.98±0.01
	30%	88.88	94.46	89.45	94.04	93.05	**95.62**	95.51	93.77±0.01	94.71±0.01
	40%	83.34	93.80	83.57	92.36	92.13	**95.26**	94.55	93.14±0.04	93.68±0.02
	50%	77.10	92.90	77.53	88.72	90.09	**94.72**	93.53	91.57±0.12	92.06±0.19
	60%	71.30	91.88	71.35	82.40	86.02	**94.28**	19.09	91.08±1.59	91.72±0.11
	70%	64.79	89.14	65.18	72.86	75.58	**92.23**	19.09	89.01±0.77	89.71±0.90
	80%	58.95	82.00	59.24	64.53	67.04	85.47	19.09	**85.66**±0.19	81.66±0.18
B167062	10%	96.65	95.93	94.36	98.12	97.26	99.11	**99.18**	97.94±1.55	96.87±7.27
	20%	92.89	92.84	91.30	97.78	97.19	99.00	**99.31**	98.17±0.00	98.31±0.00
	30%	88.39	52.83	87.52	87.86	79.29	98.75	**99.40**	97.96±0.03	98.14±0.05
	40%	83.34	49.84	83.04	82.92	76.18	98.76	**99.45**	97.69±0.01	98.06±0.00
	50%	77.58	46.92	76.93	80.15	74.63	98.26	**99.33**	97.51±0.00	97.51±0.00
	60%	71.52	43.91	70.44	75.58	71.86	97.97	**98.98**	96.91±0.00	95.34±3.75
	70%	65.04	41.37	63.10	68.11	67.59	96.08	98.82	91.48±0.73	88.62±9.65
	80%	60.13	48.91	55.41	56.68	57.33	**89.33**	62.02	88.61±0.32	82.51±5.76
B86016	10%	92.51	94.40	92.91	95.61	94.90	96.58	**99.26**	98.26±0.06	98.52±0.05
	20%	88.30	92.24	88.72	94.81	94.35	96.70	**99.22**	98.60±0.00	98.12±0.00
	30%	82.26	89.54	82.56	92.10	91.75	96.03	**99.20**	98.65±0.02	98.44±0.03
	40%	75.87	86.68	76.18	87.03	88.32	95.99	**99.19**	97.80±0.02	98.38±0.00
	50%	69.71	83.47	70.34	80.10	81.47	95.40	16.36	96.62±3.47	**97.35±0.02**
	60%	64.70	80.65	65.67	73.07	74.99	94.18	16.36	**97.41±0.05**	95.37±0.05
	70%	60.44	77.69	61.11	67.09	68.06	91.55	16.36	**94.84±0.06**	90.80±0.05
	80%	56.84	74.88	57.22	62.33	64.50	86.73	16.36	**89.65±0.25**	84.95±0.50
B196027	10%	80.06	81.62	80.43	82.39	81.87	**83.69**	83.26	79.66±0.10	78.75±0.11
	20%	76.60	79.91	77.62	81.87	81.42	**84.19**	83.17	79.73±0.07	80.14±0.11
	30%	71.76	77.62	72.31	79.57	78.86	**84.45**	83.92	78.44±0.03	81.69±0.12
	40%	68.34	75.80	68.85	76.62	76.98	**84.89**	74.45	78.50±0.02	77.54±0.01
	50%	64.04	73.81	64.53	72.40	74.09	**84.90**	11.57	78.35±0.02	79.19±0.04
	60%	60.21	71.79	60.58	67.16	69.14	**85.56**	11.57	78.67±0.14	72.10±0.28
	70%	56.68	69.42	56.93	61.58	63.43	**85.30**	11.57	74.28±1.35	73.29±1.28
	80%	54.12	66.59	54.13	57.75	59.65	**83.10**	11.57	69.65±0.99	68.37±0.11

Table 3.15: Wilcoxon provided p values resulted from the pair-wise comparison of WS-FCM with other methods for the BerkDB Salt & Pepper instances. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively show where WS-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm						
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**
B3096	10%	1.6e-06 (-)	1.6e-06 (-)	1.6e-06 (-)	1.6e-06 (-)	1.6e-06 (-)	1.6e-06 (-)	1.6e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
B42049	10%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	5.2e-05 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	0.00036 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B167062	10%	0.0027 (+)	3.9e-05 (+)	3e-06 (+)	0.0027 (-)	0.0027 (+)	1.6e-06 (-)	1.6e-06 (-)
	20%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	50%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	60%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
B86016	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (-)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	50%	1.6e-06 (+)	1.8e-06 (+)	1.6e-06 (+)	1.8e-06 (+)	1.8e-06 (+)	3e-05 (+)	1.6e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B196027	10%	1.9e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)

Table 3.16: Wilcoxon provided p values resulted from the pair-wise comparison of EWS-FCM with other methods for the BerkDB Salt & Pepper instances. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively show where EWS-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm						
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**
B3096	10%	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	20%	3.1e-05 (+)	3.1e-05 (+)	3.1e-05 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	3.1e-05 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	2.4e-06 (+)	1.7e-06 (+)	2.4e-06 (+)	2.4e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
B42049	10%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	60%	1.7e-06 (+)	2.6e-05 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	0.00036 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	4.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
B167062	10%	0.0027 (+)	0.00043 (+)	0.00038 (+)	0.0027 (-)	0.0027 (-)	1.7e-06 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.1** (=)	1.7e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	7.7e-06 (-)	1.7e-06 (+)
B86016	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	40%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
B196027	10%	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	3.1e-05 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	60%	1.7e-06 (+)	3.1e-05 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)



(a) SA variance.

(b) SA mean.

Figure 3.15: Demonstration of different algorithms performance on noise variation (BerkDB Salt & Pepper instances).

Table 3.17: Wilcoxon provided p values resulted from the pair-wise comparison of WS-FCM with EWS-FCM for the BerkDB Salt & Pepper instances. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively show where EWS-FCM performs significantly better, significantly worse or not significantly different than WS-FCM.

	Noise Level							
Img.	10%	20%	30%	40%	50%	60%	70%	80%
B3096	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	3.2e-06 (+)
B42049	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (-)
B167062	0.0027 (-)	1.7e-06 (+)	1.7e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	**0.16** (=)	**0.1** (=)	0.00017 (-)
B86016	1.7e-06 (+)	1.7e-06 (-)	1.9e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)
B196027	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	2.8e-05 (-)	3.1e-05 (-)

80%, and $B196027, density = 10\%$. Overall, the proposed WS-FCM and EWS-FCM perform better than other methods in terms of producing less redundant segments. However, they suffer from over-smoothing the image and eliminating the details. FGFCM_S2 mostly, and FLICM sometimes, produce more accurate results because of preserving more details and no smoothing around the edges. Other methods produce many segmented regions mostly, which affects detection of the region of interest.

3.6 Chapter Summary

Two new noisy image segmentation algorithms were proposed in this chapter. The first algorithm, WS-FCM, utilizes adaptive thresholding of wavelet coefficients, in which the optimal values of thresholds were determined using PSO. FCM was used as a fitness metric in the PSO search, and also as the final segmentation algorithm. The second algorithm, EWS-FCM, adds edge enhancement applied to the low-frequency coefficients of the coarsest scale. Unlike the other FCM-based noisy image segmentation methods, the proposed algorithms look at the problem from feature enhancement/manipulation point of view. The distinctive properties of the new method are the three following:

- It shows much better than other methods performance on severely noisy images particularly with Gaussian noise.
- It does not need parameter-tuning for different noise levels and types.
- It produces very stable results even when noise volume has a large variation.

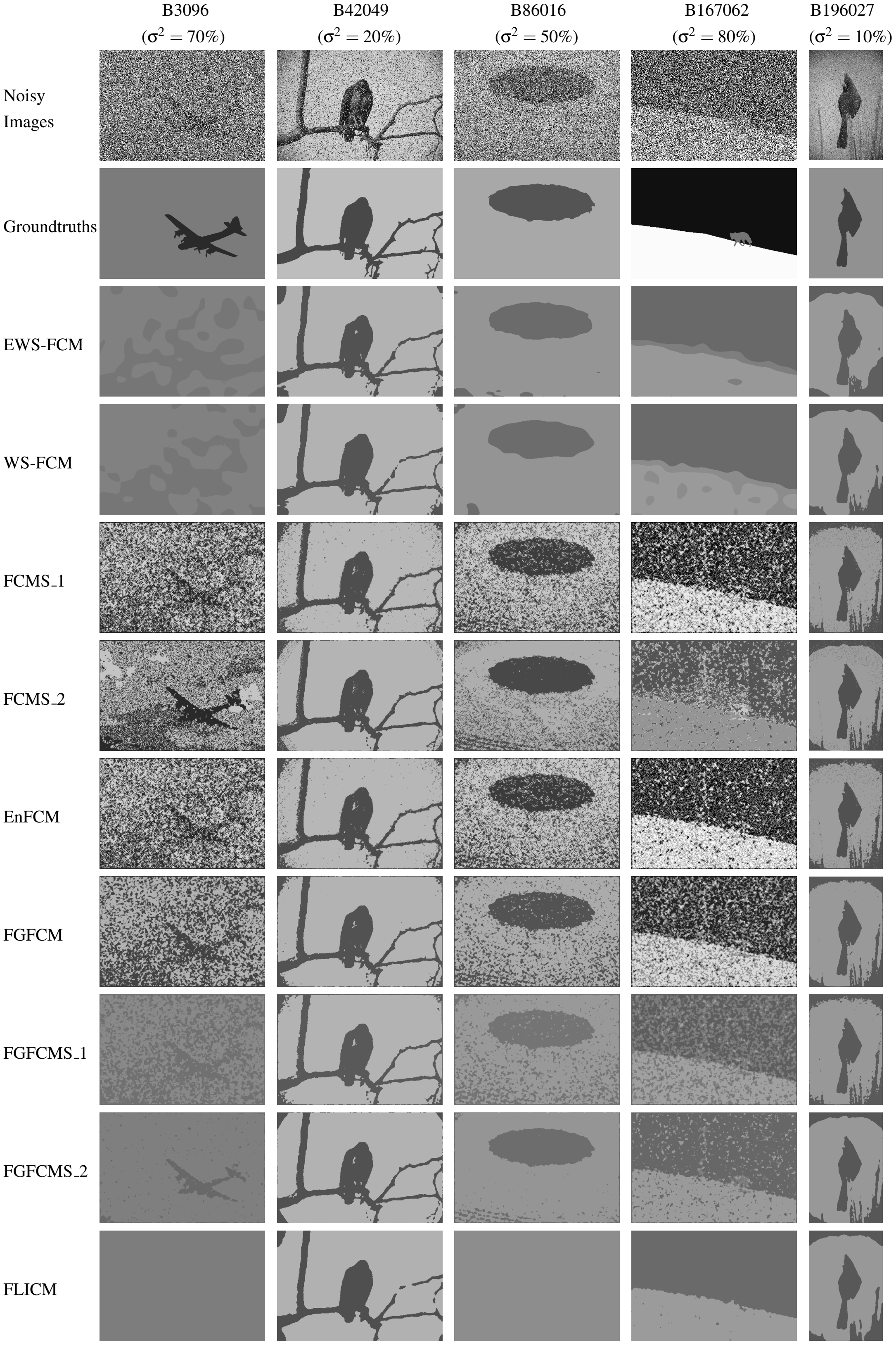


Figure 3.16: Qualitative comparison of the proposed WS-FCM and EWS-FCM with FCM_S1, FCM_S2, EnFCM, FGFCM_S1, FGFCM_S2, FGFCM, and FLICM on some BerkDB Salt & Pepper instances. Noise density for B3096, B42049, B86016, B167062, and B196027 instances is 70%, 20%, 50%, 80%, and 10% respectively.

The proposed WS-FCM and EWS-FCM mostly perform significantly better than other state-of-the-art methods on Gaussian noisy images with EWS-FCM being slightly better than WS-FCM. WS-FCM segmentation results show fewer redundant segments caused by noise compared to EWS-FCM, but the better performance of EWS-FCM around the edges results in the greater accuracy of WS-FCM. In the case of Salt & Pepper noise, the results are not as promising, and over-smoothing becomes a negative point. The proposed methods still outperform the majority of state-of-the-art-methods. The only method that outperforms ours overall is FGFCM_S2.

WS-FCM and EWS-FCM perform well in extracting coherent regions and eliminating noise, but they have limitations in preserving details such as edges and small regions. EWS-FCM is less problematic in this regard, but its performance is only slightly higher than WS-FCM. In the next two chapters, we discuss how to address this issue by utilizing other tools or techniques.

Chapter 4

Wavelet Shrinkage and Anisotropic Diffusion for Severely Noisy Image Segmentation

4.1 Introduction

Severely noisy image segmentation has seldom been investigated in the literature. With a high volume of noise, preserving the details while performing the segmentation can be extremely challenging. In the previous chapter we introduced an FCM-based image segmentation method for gray-level images called WS-FCM utilizing adaptive thresholding of wavelet coefficients with PSO. Although the method performs successfully in some aspects, one of its drawbacks is elimination of some of the important details. This happens due to the over-smoothing properties of the proposed WS-FCM, and is specifically observable in details related to region boundaries. This chapter addresses this issue by applying another detail-preserving denoising method, and then fusing both results in the wavelet domain. This has positive qualitative and quantitative effects on segmentation results.

4.1.1 Chapter Goals

This chapter proposes a novel combination of two image denoising techniques in order to explore severely noisy image segmentation based on FCM clustering performance.

More specifically, we address how different denoising methods can be combined in order to produce accurate and consistent results. The results in this chapter are based on the algorithm introduced in the previous chapter named WS-FCM. The chapter goals can be summarized as follows:

- Exploring the abilities of wavelet and spatial domain denoising algorithms in order to benefit from the strengths of denoising algorithms in both domains in an FCM-based noisy image segmentation scheme.

- Exploring possible ways to address the over-smoothing drawbacks of our wavelet coefficients shrinkage approach, WS-FCM, using other edge-preserving denoising algorithms for FCM-based severely noisy image segmentation.

- Exploring the abilities of existing edge-preserving denoising algorithms and selecting one that has a proper performance in severely noisy images and a predictable performance with noise level variation.

- Finding a proper domain and scheme to fuse the denoising results of WS-FCM and the selected edge-preserving denoising algorithm.

- Comparing the performance of the proposed method with several FCM-modified algorithms, and other FCM-based feature enhancing segmentation algorithms on severely noisy image segmentation.

4.1.2 Chapter Organization

The remainder of this chapter is as follows. Section 4.2 is dedicated to the methodology of the proposed approach in this chapter. Section 4.3 introduces the foundation of the new detail-preserving severely noisy image segmentation approach. Section 4.4 provides the datasets, comparison algorithms, and evaluation metrics. Section 4.5 is devoted to a comprehensive analysis of the results and discussions, and Section 4.6 concludes this chapter.

4.2 Methodology

This section briefly describes the techniques we have utilized in the introduction of the new proposed approach. The specific representation of the techniques in our approach will be discussed in the next section.

4.2.1 Wavelets

Detailed introduction to wavelets and image denoising based on thresholding of wavelet coefficients was given in subsection 3.2.1 in the previous chapter (page 33).

4.2.2 Anisotropic Diffusion (AD)

Methods of noise removal were known a long time before the Anisotropic Diffusion case was first claimed in 1990. Most denoising methods typically apply a blurring/smoothing filter as a kernel to the image, as blurring hopefully results in a smooth, less noisy image. Such an image, with low noise levels, is a much better starting point for any image analysis process. Blurring removes both significant details and artifact of noise. With severe noise, the blurring has to be stronger and therefore it will remove some of the significant details from the image. That is why there has been an everlasting search for an effective noise removal method that is able to denoise and clarify the details in an image as much as possible.

AD is an algorithm introduced by Prona and Malik [101] aiming at reducing the noise in an image while preserving the details such as edges and lines that are important for properly interpreting an image. It is an extension to the concept of *Scale-space* [136], which is a technique for multi-scale signal processing. It provides a framework to handle an image structure at different scales by representing the image as a family of smoothed images from the finest resolution to the coarsest. The scale-space parameter, t, controls the smoothing property of the kernels. The most popular Scale-space is the linear Gaussian scale-space in which a Gaussian-based kernel with variance of t is applied at each scale. Assume that $I(x,y,t)$ are a family of images obtained from the convolution of an original image, $I_0(x,y)$ with a Gaussian kernel, $G(x,y;t)$ of variance t:

$$I(x,y,t) = I_0(x,y) * G(x,y;t) \tag{4.1}$$

in which larger values of t corresponds to images at coarser resolutions. One could show that one-parameter family of derived images from Eq. 4.1, is equivalent to the solution to the isotropic heat conduction (or isotropic diffusion) equation [66, 73]:

$$I_t = div(c\nabla) = c\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{4.2}$$

where c is the diffusion coefficient. div, ∇, and Δ are the divergence, gradient, and Laplacian operators. Like heat generated in a cold surrounding that tends to flux from hotter regions to colder regions, the gradient of brightness could represents the margins of temperature excellently. As time advances on t, each potent molecule (a "hot" unit) spreads in the direction of its gradient vector. In an image, it is not heat, but brightness level. So, an image could be generalized to be a surface, where bright spots are "hot" and dark spots are "cold".

One problem with the standard Scale-space approach is that the true location of edges and junctions which carry a lot of spatial information, may not directly exist at a coarse scale or are destroyed. This spatial distortion is due to the fact that Gaussian smoothing does not respect the natural boundaries of regions. Proma and Malik proposed the following criteria for a new multi-scale paradigm to describe image semantically meaningful [101]:

1. **Causality**: a proper multi-scale representation should not produce any spurious details moving from finer to coarser scales.

2. **Immediate localization**: region boundaries at each scale should be sharp and occur at the same spatial position that semantically meaningful boundaries exist at that resolution.

3. **Piecewise smoothing**: intra-region smoothing should have priority over inter-region smoothing at all scales.

The anisotropic diffusion equation as below, unlike the isotropic equation, considers the diffusion coefficient, c, not to be a constant:

$$I_t = div\big(c(x,y,t)\nabla I\big) = c(x,y,t)\Delta I + \nabla c.\nabla I \tag{4.3}$$

This equation becomes the isotropic heat equation as in 4.1 if $c(x,y,t)$ is a constant. While the linear (isotropic) approach, Eq. 4.1, treats every pixel with the exact same convolution, the non-linear (anisotropic) approach in Eq. 4.3 treats a pixel with varying intensity, depending on its neighborhood qualities. If we could compute a current best estimate of the location of boundaries at a certain scale, $E(x,y,t)$, then smoothing within regions could be promoted in preference to smoothing across the boundaries. This will allow smoothing to happen independently in each region with no interactions between regions. Generally, if (x,y) is a part of an edge, little smoothing should be applied, and if it is not part of an edge, full smoothing should be applied. We can achieve outstanding results by setting the boundary estimation, E, to be the gradient of the intensity function and the conduction parameter, $c(x,y,t)$ to be a function of the magnitude of E:

$$E(x,y,t) = \nabla I(x,y,t) \tag{4.4}$$

$$c(x,y,t) = g\Big(\|\nabla I(x,y,t)\|\Big) \tag{4.5}$$

where $g(.)$ has to be a nonnegative monotonically decreasing function with $g(0) = 1$. $c(x,y,t)$ is large when (x,y) is not a part of an edge, and $c(x,y,t)$ is small when (x,y) is a part of an edge. This guarantees that the diffusion process mainly happens inside the regions without affecting the region boundaries where the magnitude of E is large. The authors in [101] introduce the two following functions for $g(.)$, though other functions could also be proposed that would provide perceptually similar results:

$$\begin{aligned} g(\|\nabla I\|) &= e^{(-(\|\nabla I\|/K)^2)} \\ g(\|\nabla I\|) &= \frac{1}{1+\left(\frac{\|\nabla I\|}{K}\right)}^{2} \end{aligned} \tag{4.6}$$

where the parameter K could be fixed at some constant value. The first equation advantages high-contrast edges over low-contrast ones, and the second equation advantages wide regions over smaller ones. Throughout this chapter the second equation has been applied as the $g(.)$ function.

4.2.3 Connected Component Analysis

Extracting and labeling of various disjoint and connected components in an image is crucial in many automated image analysis applications. Connected Component Analysis (CCA), also known as connected component labeling, is a heuristic technique used in computer vision to detect connected region in binary images. Connected components labeling scans an image and groups its pixels into components based on pixel connectivity, i.e. all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labeled with a gray-level or a color (color labeling) according to the component it was assigned to. To get good results it is critical to obtain a good binary image which separates the objects from the background, whereas CCA is straightforward part of the automated analysis process.

Connected component labeling works on binary or gray-level images and different measures of connectivity are possible. To apply binary CCA to gray level images, we need to modify the algorithm in some ways. In this approach, we propose a scheme to convert segmented gray level images into several binary images. There are different binary CCA algorithms to find the connected components in a binary image; this research uses the one based on the following basic steps:

1. Search for the next unlabeled pixel, p.

2. Use a flood-fill algorithm to label all the pixels in the connected component containing p.

3. Repeat steps 1 and 2 until all the pixels are labeled.

4.3 Fusion of Wavelet Shrinkage and Anisotropic Diffusion

The proposed method in this chapter fuses the two denoising algorithms of adaptive wavelet shrinkage in the frequency domain (proposed in the previous chapter with the same parameter settings) and anisotropic diffusion in the spatial domain for severely noisy image segmentation. We call the proposed method Aggregation of Wavelet coefficients shrinkage and Anisotropic diffusion for FCM-based noisy image segmentation (AWA-FCM). The reason for the fusion is to benefit from both techniques to introduce a noise suppressing algorithm with more detail- preserving properties.

Fig. 4.1 shows a detailed block diagram of the proposed method. The figure shows four major processes named A, B, C and D. Block A is the feature enhancing part of WS-FCM in which adaptive wavelet shrinkage using PSO is applied for image denoising. WS-FCM is adaptive in a sense that it comes up with unique threshold values for each image based on the noise volume, type, and intensity properties. Block B is the AD denoising process. Having an image denoised using this method at a proper scale could preserve the details better than WS-FCM while not removing the noise as effectively. An improper scale number can result in either to much noise remaining in an image or over-smoothing. Therefore, this block includes also a scale number estimation step according to the level of noise. Block C is where the denoising results of blocks A and B are fused together. The key issue in combining the algorithms is how to combine their results in a way that it preserves the denoising properties of WS-FCM, but enhances the detail information with the edge-preserving properties of AD. Block D performs final image reconstruction and segmentation on the feature-enhanced image.

4.3.1 Scale Number Estimation in AD

As mentioned before, the scale number in AD plays an important role in denoising. It acts like an iteration number which has to be larger when the level of noise is high. Since the noise volume varies in our application, one important task is the estimation of the scale number. Based on extensive empirical experiments on different noisy images, we discovered that if the scale number changes with the estimated standard deviation of noise according to the curve provided in Fig. 4.2, the denoised outcome image generally has enough detail information while not having too much noise remaining in the image. The curve equation is:

$$f(\sigma) = 270.9\sigma^2 - 45.13\sigma + 1.564 \tag{4.7}$$

in which $f(\sigma)$ is the scale number, and σ is the standard deviation of noise estimated in the WS-FCM approach according to the following equation:

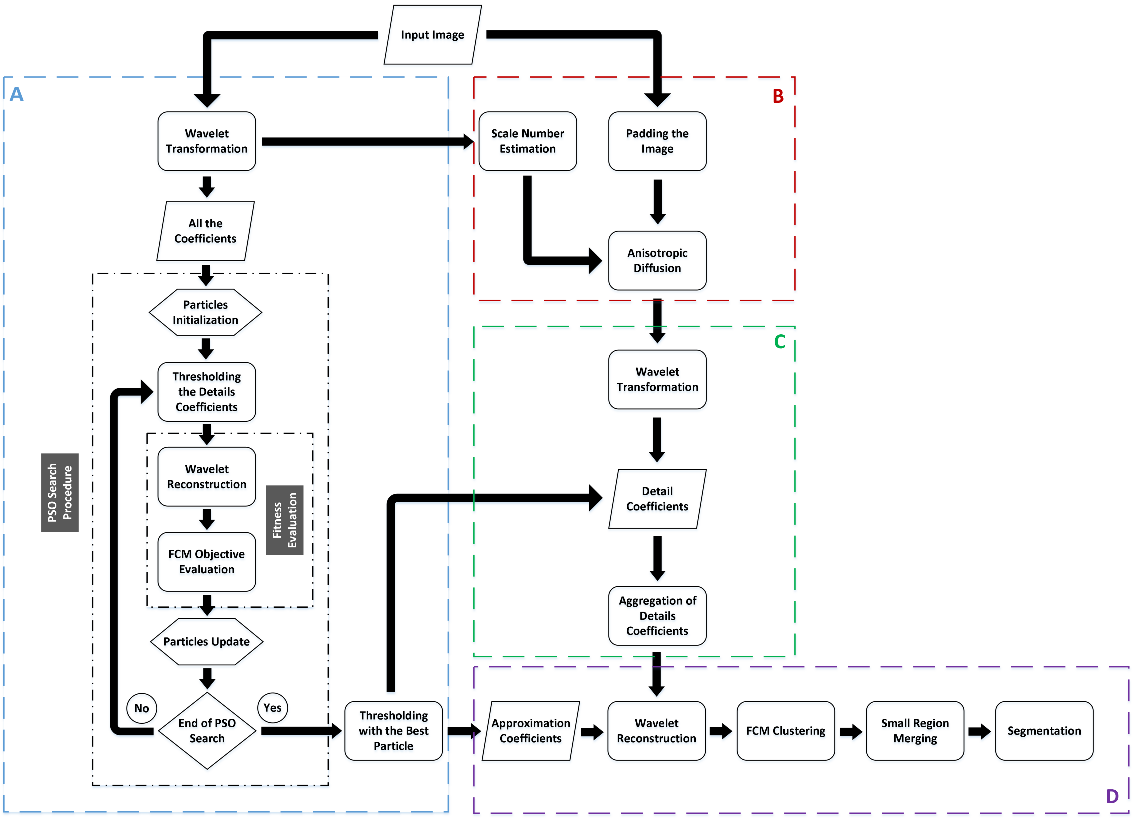


Figure 4.1: Block diagram of the proposed AWA-FCM algorithms.

$$\sigma = \frac{median\left[\left|w_{mn} : i, j \in HH_1\right|\right]}{0.6745} \tag{4.8}$$

where w_{mn} are noisy coefficients of the finest high-frequency subband, HH_1, as proposed by [47]. σ is calculated in the wavelet transformation step of WS-FCM, and is used throughout the method wherever needed.

The gradient-based implementation of diffusion in AD has a tendency to strengthen all the edges and borders when the scale number increases, but the error in the misplacement of the edges also generally increases due to the existence of noise. The higher the noise level, the more serious the problem. Increasing the scale number also increases redundant borders around the real image borders. To prevent this, we pad the original image with a 10-pixel-wide margin. This margin is set to repeat border pixels of the image.

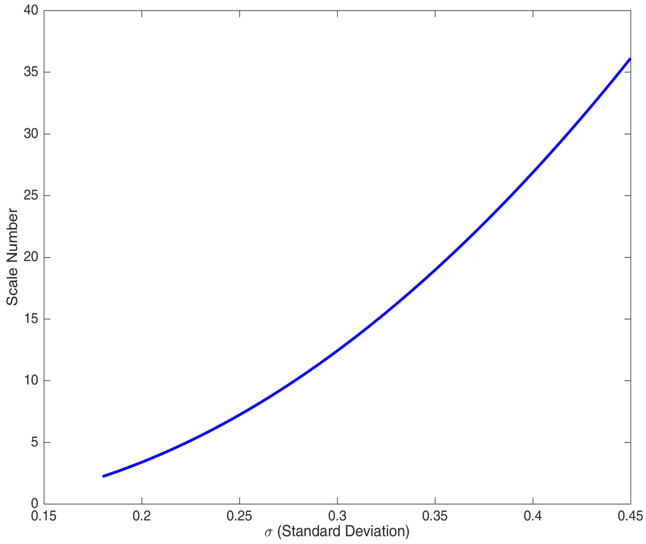


Figure 4.2: A curve that estimates AD scale numbers from estimated noise standard deviation (σ).

4.3.2 Fusion in Wavelet Domain

So far, we have two denoised images obtained from the denoising process of WS-FCM and Anisotropic Diffusion based on the proposed representation. Now we have to find a way to fuse the results in a way that takes the best out of each result. Our goal is to add back some of the image details that are removed from the image during the wavelet shrinkage. This means that coefficients representing boundaries that were removed in the shrinkage process, need to be brought back. Given that the denoised image from AD has preserved the edges and is reasonably denoised, converting it to wavelet domain provides coefficients that are mostly related to real edge information. However, the conversion also provides some coefficients that are related to fake edge information caused by noise. Therefore, aggregation of detail coefficients from the two outputs in the wavelet domain could add back some of the lost details, but would also add some noise. Since we have a five-scale wavelet transformation for WS-FCM, the output image of AD also has to be converted in a five-scale manner. We separate the detail coefficients (H, V, and D) from the denoising results of WS-FCM and AD, and aggregate them according to the following equations:

$$\begin{aligned} H_f^i &= \frac{H^i_{WS-FCM} + b \times H^i_{AD}}{1+b} \\ V_f^i &= \frac{V^i_{WS-FCM} + b \times V^i_{AD}}{1+b} \\ D_f^i &= \frac{D^i_{WS-FCM} + b \times D^i_{AD}}{1+b} \qquad i = 1, 2, ..., 5 \end{aligned} \tag{4.9}$$

in which H^i_{WS-FCM}, V^i_{WS-FCM}, and D^i_{WS-FCM} are the detail coefficients at scale i from the feature enhancing process of WS-FCM, and H^i_{AD}, V^i_{AD}, and D^i_{AD} are that of the AD feature-enhanced image, transformed to wavelet domain. The approximation coefficients do not take part in the aggregation process. b is a constant representing the weighting given to AD coefficients, and is set as 2σ based on extensive quantitative experiments on different noisy images. σ is the noise standard deviation estimated from equation 4.8. $b = 2\sigma$ indicates that higher levels of noise require higher contribution of AD results. For higher volumes of noise, more coefficients tend to lose their original values. Larger coefficients are shrunk by the soft-thresholding function in WS-FCM. This causes more severe over-smoothing of edges. Therefore, details are more likely to vanish in higher levels of noise. Adjusting these large coefficients to their original values adds back the edge information. In the aggregation step, this is being taken care of by an increasing contribution of wavelet coefficients from the AD denoised image for higher volumes of noise.

However, in extremely high noise volumes, wavelet coefficients from the AD denoised image would contain too many artifacts which would be hard to eliminate. Therefore, the contribution of AD need to be scaled down for very high volume of noise. Again, based on extensive quantitative experiments, we discovered that when the estimated σ is bigger than 0.4, b has to decrease, and also it has to increase at a lower pace when σ increases. So the overall formula to determine the contribution of AD in the fusion scheme is:

$$b = \begin{cases} 2\sigma, & \sigma < 0.4 \\ \sigma, & \sigma \geq 0.4 \end{cases} \tag{4.10}$$

Next, the approximation coefficients are added to the aggregated detail coefficients, and a wavelet reconstruction is performed to obtain the intensity image. This image has more details into it due to the aggregation. Segmentation of this image will show some very small spurious regions created among big homogeneous regions. In the final post-processing step, these small regions are merged into the bigger ones in their background as explained in the next section.

4.3.3 Elimination of Small Spurious Regions

While adding edge information in the aggregation process, some spurious edge information might be introduced to the image, caused by noise rather than real edges. Thanks to the good performance of WS-FCM on eliminating small regions and producing compact segments, it is easy to get rid of some of these spurious regions since they typically appear at spatial positions that have big compact regions around them. For this, we use the concept of Connected Component Analysis (CCA) to remove them. Essentially, we search for small connected regions that are surrounded by larger background regions.

Overall, the elimination process includes the following steps:

1. Cluster the reconstructed image using FCM.
2. Find the maximum membership for each pixel to determine the cluster each pixel belongs to.
3. Label each pixel with a number based on its cluster to differentiate pixels belonging to different clusters.
4. For each label in the image: find the connected components of pixels with that label.
5. Select the small components (based on the number of pixels inside them), and then find the bounding rectangle surrounding those small regions.
6. For each bounding rectangle: look for the labels of the pixels inside the bounding rectangle that belong to other labels, and then find the label with maximum cardinality.
7. For each bounding rectangle: change each pixel's label number inside the small connected component to the label found in the previous step.

To find connected components, an 8-connectivity analysis is performed. Small regions are defined as regions with the number of pixels inside them less than 40 based on experiments.

4.3.4 Segmentation

The final label matrix from the previous step and the original noisy image are required in order to obtain a segmented image. Intensity values of all the pixels in the noisy image that have the same label number in the label matrix are averaged, and the result is assigned to all those pixels. This way, all the segments having the same label number will be represented with a unique gray-level value. If we assume I_n, L, and I_s are equal-sized matrices of the noisy image, the labeled matrix, and the final segmented image respectively, we can write:

$$I_s(x,y) = mean\Big\{I_n(u,v)|L(u,v) = L(x,y)\Big\} \tag{4.11}$$

4.4 Experiment Design

4.4.1 Datasets and Evaluation

We use a synthetic and a real dataset to evaluate the performance of the proposed method on images with different properties. The synthetic dataset, SynthDB, consists of images in which the regions are geometrically simple, easy to recognize, and are composed of a single intensity value. Running experiments on noisy versions of images from this dataset provides an understanding of how the proposed method performs on segmentation of noisy regions when dealing with simple images. We have five images in this dataset named $Synth1, Synth2, ...,$ and $Synth5$. The number of clusters while performing FCM-based clustering is determined by the number of homogeneous regions in each image.

The real dataset, BerkDB, is selected from the Berkeley dataset [88] which has been specifically created for segmentation and boundary detection purposes. Here the images are not so simple: the regions are not geometrically simple, they are not easy to recognize, and each region is composed of several intensities. Experiments on this dataset reveals how accurate and consistent the proposed method is when encountering real-world images. We have 11 images in this dataset named $B3096, B42049, B167062, B86016,$, $B196027$, $B24063$, $B253036$, $B147091$, $B108073$, and $B135069$. The number of clusters is again determined according to the number of homogeneous and coherent regions in each image. The groundtruth is either selected from one of the provided groundtruths in the Berkeley dataset (the one with the minimum number of segments), or is formed from one of the groundtruths. For the latter, all the segments belonging to a coherent region are merged together as segments that correspond to a cluster in FCM results. Fig. 4.3 provides all the images from the two datasets along with the preferred number of clusters.

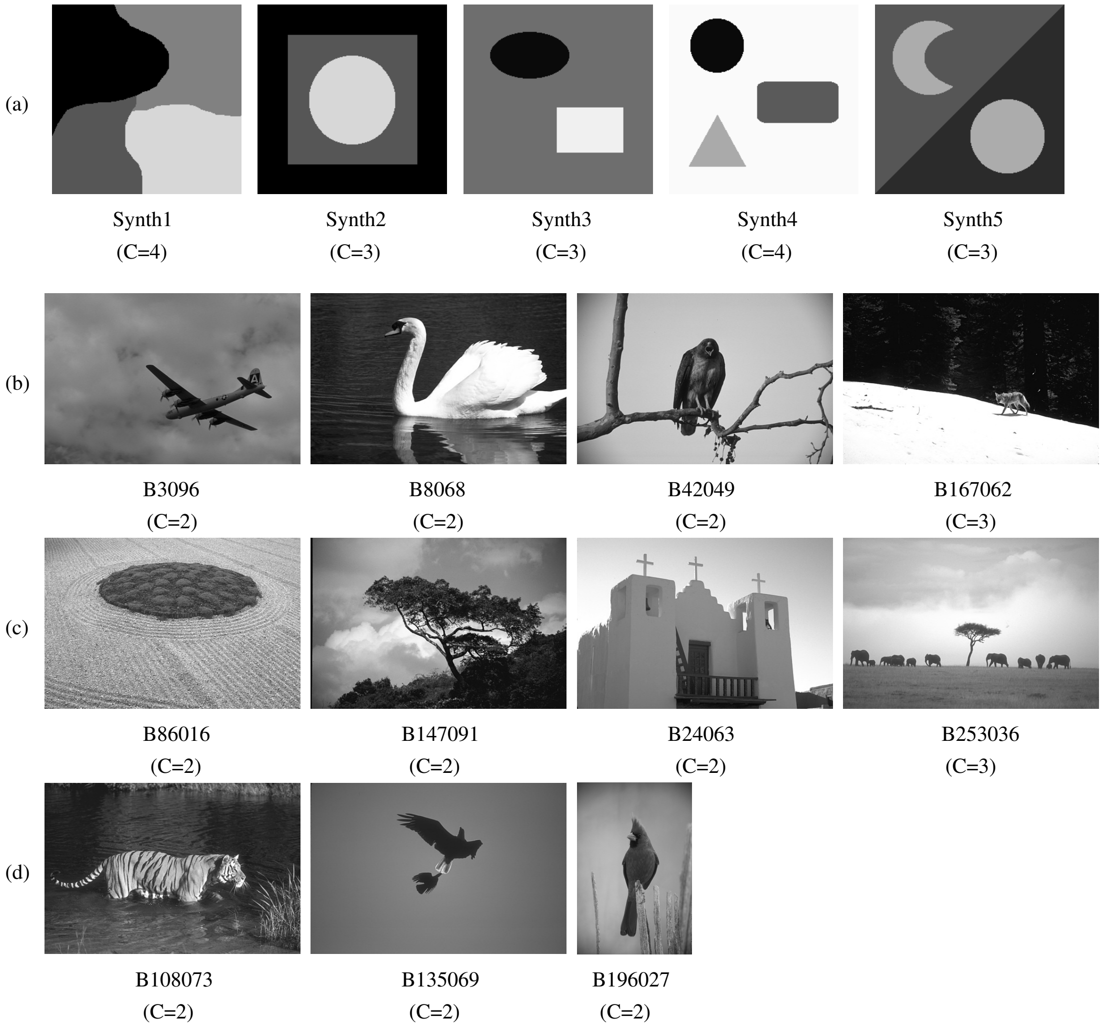
To generate the noisy images, we apply the common Gaussian noises. Each image is corrupted with Gaussian noise of different volumes to examine the performance of our method under different severities of noise, and to analyze the effect of noise variation. The variance of the noise level ranges from 10% to 80%. For the rest of this chapter "image" refers to one of the original images in Fig. 4.3, and "instance" refers to a sample noisy image generated from either datasets. Given that we have five and 11 images in SynthDB and BerkDB datasets respectively, and eight different levels of noise for each image, there are a total of 128 instances, 40 of which are created from SynthDB and the other 88 instances are created from the BerkDB dataset.

For qualitative evaluation we have adopted the common Segmentation Accuracy, SA metric [6], which was explained in Chapter 3 (page 45).

We have selected nine other state-of-the-are algorithms to compare our method with. Seven of these methods are FCM-modified algorithms named FCM_S1 and FCM_S2 [38], EnFCM [118], FGFCM, FGFCM_S1 and FGFCM_S2 [25], and FLICM [74]. The two rest are WS-FCM and AD-FCM. WS-FCM is the PSO-based approach for adaptive wavelet thresholding proposed in the previous chapter and AD-FCM is the Anisotropic Diffusion based on the proposed representation applied for feature enhancement to FCM-based image segmentation.

4.4.2 Parameter Design

All the parameters and settings related to FCM, PSO, wavelet transform, AD, the fusion process and CCA, and their values/settings used in this chapter are listed in Table 4.1



Synth1 (C=4) Synth2 (C=3) Synth3 (C=3) Synth4 (C=4) Synth5 (C=3)

B3096 (C=2) B8068 (C=2) B42049 (C=2) B167062 (C=3)

B86016 (C=2) B147091 (C=2) B24063 (C=2) B253036 (C=3)

B108073 (C=2) B135069 (C=2) B196027 (C=2)

Figure 4.3: Original test images from the SynthDB, row (a), and BerkDB, rows (b),(c), and (d), with the number of required clusters in the segmented image.

based on empirical search through experiments.

Table 4.1: Parameter settings of the proposed method.

Parameter/Setting	**Value/Type**
Wavelet filter	Coiflets family
Scale number	5
Termination threshold for FCM	0.001
Maximum number of iterations for FCM	100
Weighting exponent (m)	2
Padding margin size (in pixels)	10
Small regions size for merging (in pixels)	40
$f(\sigma)$ (AD scale number estimating function)	$270.9\sigma^2 - 45.13\sigma + 1.564$ (σ is estimated according to Eq. 4.8)
Image padding size (in pixels)	10
b (the aggregation parameter)	σ or 2σ (σ is estimated according to Eq. 4.8)
Spurious region size (in pixels)	40
CCA connectivity type	8-connectivity

4.4.3 Statistical Significance Test

To analyze the non-deterministic behavior of PSO in our algorithms a pair-wise statistical significance test is performed. Our algorithm is run 30 times independently on each image, and the results in the form of the SA values are compared with the results of other methods using the test. We select the Wilcoxon test with a significance level of 0.05 for this comparison. For more information about this test please refer to [135]. If the p-value (the probability of observing a test statistic as or more extreme than the observed value under the null hypothesis) is greater than the significance level, the pair-wise comparison is not considered significantly different. Otherwise, one method is significantly better than the other. Tables that are providing p values also use **+**, **-** and **=** signs to show that one method is significantly better, worse, or not different respectively for an instance.

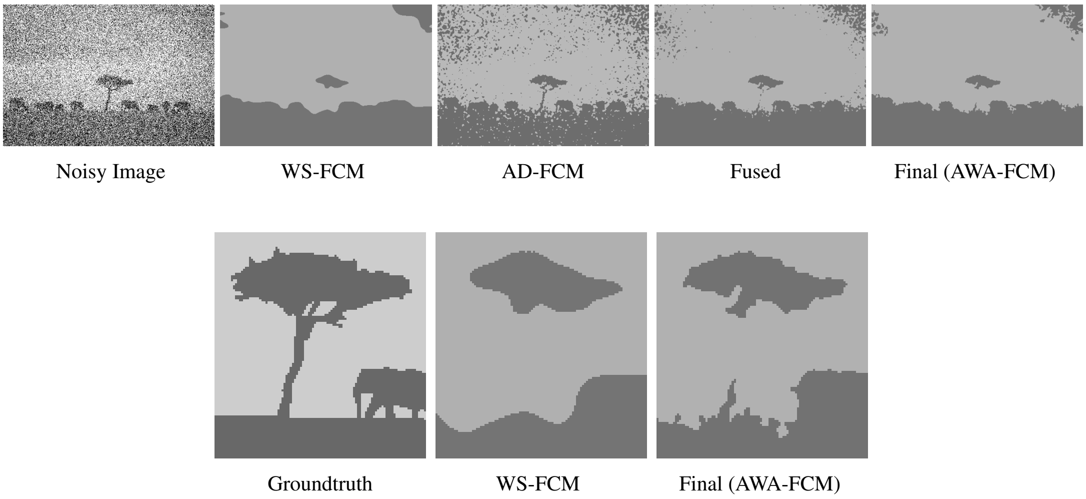
4.5 Results and Discussions

This section analyzes all the experiments in comparison to other methods both quantitatively and qualitatively. To have a proper comparison in the quantitative part, both SA and p values are considered for an instance. The comparison is made in separate sub-sections for different datasets. Each sub-section considers two different aspects of the analyses. The first part, presents a comprehensive quantitative evaluation and the second part presents some sample qualitative evaluations. In the quantitative evaluation part, we compare the results of the proposed AWA-FCM with the comparison methods on each instance in a pair-wise manner. Throughout the pair-wise comparison, a better performance is determined according to SA value unless the p value for that pair is bigger than the significance level (0.05). Next, another evaluation is done by ranking the performance of all the algorithms on each instance. Based on the number of the best and second-best performances for each algorithm, the overall best, second-best, and third best performers are determined. The quantitative analysis in each sub-section concludes with a performance evaluation on noise variation on each sample image. This evaluation investigates how SA metric changes in terms of variance and mean when the noise level ranges between 10% to 80%.

4.5.1 A Visual Example of the Segmentation Results

For further clarity, we provide a visual showcase of what takes place at different steps of AWA-FCM . Fig. 4.4 provides the segmentation outputs of these steps. Segmentation results belong to WS-FCM, AD-FCM, fusion of WS-FCM and AD-FCM before the postprocessing step, and fusion followed by the postprocessing. Considering the $B253036, \sigma^2 = 30$ instance (Fig. 4.4-c), this example shows how the details existing in the AD-FCM segmentation result (Fig. 4.4-c) is carried to to the WS-FCM segmentation results (Fig. 4.4-b) in the fusion of both methods before the postprocessing (Fig. 4.4-d), and how the postprocessing step refines the results of the fusion step by removing very small segments (Fig. 4.4-e). For a better observation, zoomed shots from an area around the tree in the image is provided in the second raw of Fig. 4.4. Fig. 4.4-f shows the shot from the groundtruth image, Fig. 4.4-g is the shot from the WS-FCM segmentation result, and Fig. 4.4-h is the shot from AWA-FCM segmentation result.

Another aspect of the fusion process is that it also removes a great deal of the artifacts causing the redundant regions. This makes the final postprocessing step more effective as removing the small regions from AD-FCM results could be much more challenging, and could compromise accuracy.



Noisy Image WS-FCM AD-FCM Fused Final (AWA-FCM)

Groundtruth WS-FCM Final (AWA-FCM)

Figure 4.4: A visual example of implementation of different steps in the proposed AWA-FCM. First row: the instance $B253036, \sigma^2 = 30$ along with the segmentation results of WS-FCM, AD-FCM, the fusion process before the postprocessing, and final results (AWA-FCM). Second row: shots from similar locations of the groundtruth, and segmentation results of WS-FCM and AWA-FCM.

4.5.2 SynthDB (Quantitative Analysis)

Table 4.2 provides the SA values for all the methods, and also Table 4.3 provides the results of the statistical significance test. Overall, there is only one instance for which the p value is greater than the significance level. This instance belongs to the pair-wise comparison of AWA-FCM and WS-FCM for the instance $Synth4, \sigma^2 = 80\%$. Other than this single instance, for the rest of the instances, the results from Table 4.2 indicates that the proposed AWA-FCM performs significantly better than all other methods in a pair-wise comparison in almost all the instances. More clearly, AWA-FCM performs better than FCM_S1, FCM_S2, EnFCM, FGFCM, FGFCM_S1 and FGFCM_S2, WS-FCM, and AD-FCM in all the 40 instances. In comparison to FLICM, AWA-FCM performs significantly better in all but one instance.

Ranking the performances of different algorithms on a single instance, an analysis of SA and p values from Tables 4.2 and 4.3 shows that AWA-FCM, FLICM, WS-FCM, AD-FCM are holding the highest number of best performances respectively. AWA-FCM as the overall best performer, is the best performer in 38 instances, and second-best performer in another instance. The second overall best performer is FLICM being the best and second-best performer in one instance. The third overall performer is WS-FCM being none as the best performer, and the second-best performer on 33 instances. The fourth overall best performer is AD-FCM having none best performances, and four second-best performances.

For the next experiments, we analyze the effect of noise variation on SA values. For this, SA variance and mean of each method on each image when then noise level takes eight different values is depicted in the graphs in Fig. 4.5. As the graph in Fig. 4.5(a) shows, the proposed AWA-FCM plotted in red bar, possesses four out of five lowest SA variances on the images $S1, S2, S3$, and $S5$, and it holds the second-lowest SA variance for the image $S4$ where AD-FCM has the lowest SA variance. AD-FCM has the third-lowest variances on four other instances. WS-FCM has four second-lowest SA variance on four images, and the third-lowest variance in an another image. Overall, AWA-FCM, WS-FCM, and AD-FCM are the methods with the least SA variance over the noise variation.

The other graph in Fig. 4.5 shows the SA mean of each method for each image when the noise level increases (Fig. 4.5(b)). This graph clearly shows that AWA-FCM plotted in red bar, has the highest mean performance on all five images. Form this point of view, WS-FCM is always the second-best performer, and AD-FCM is the third-best performer.

SynthDB (Qualitative Analysis)

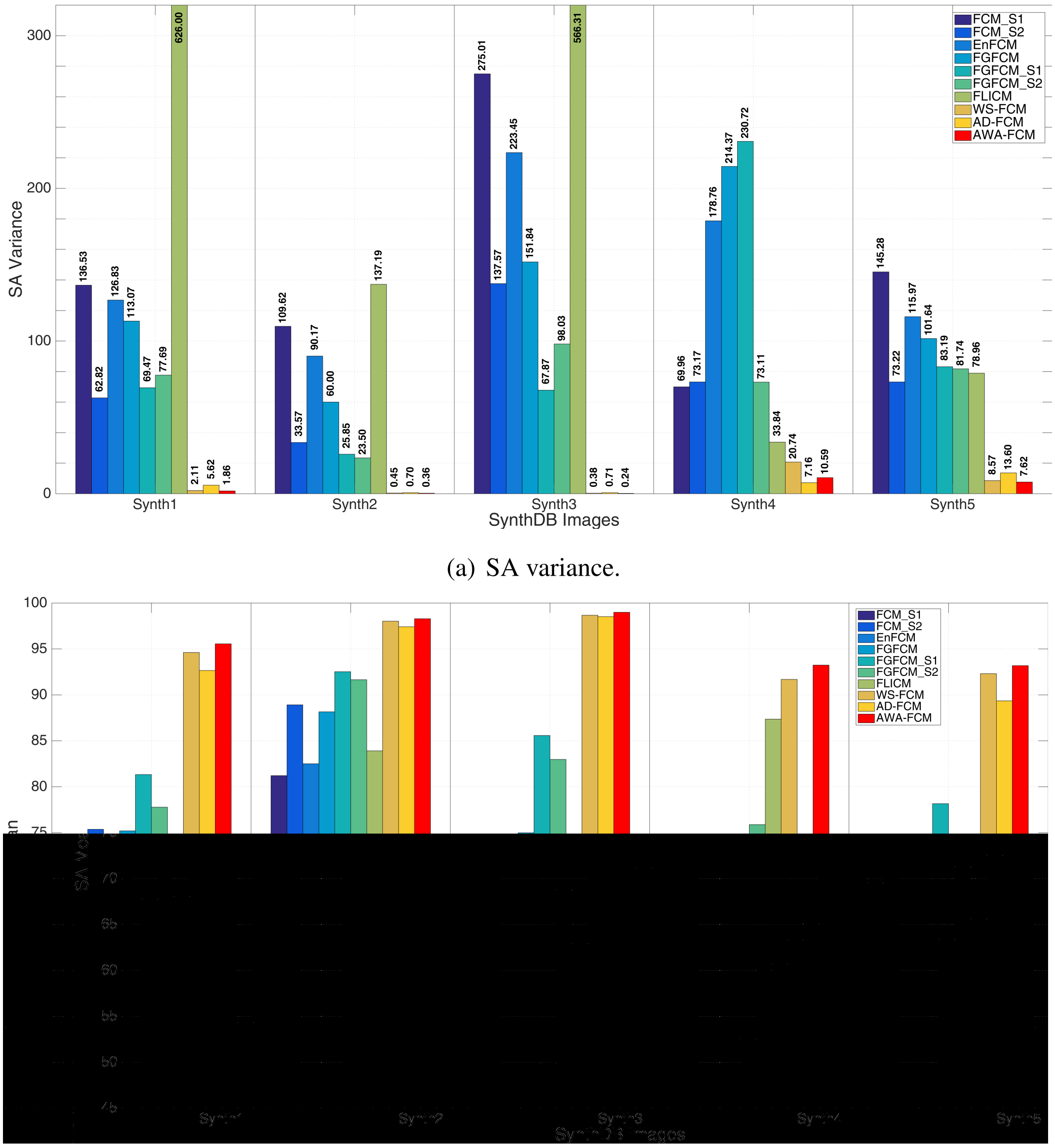
Figure 4.6 provides segmentation results of all the methods on some samples form the SynthDB. An instances (out of eight) has been selected from the noisy instances for each image. These instances for S1, S2, S3, S4, and S5 are $\sigma^2 = 70\%, 80\%, 50\%, 60\%$, and 20% respectively. One could see that the segmentation results of FCM_S1, FCM_S2, EnFCM, FGFCM, FGFCM_S1 and FGFCM_S2 are problematic, for over-segmentation is preventing the method from segmenting the main regions in the images compactly. FLICM could sometimes perform compact region segmentation on images S2, S3 and

Table 4.2: SA values for the SynthDB instances. The bold numbers indicate the best overall performance for each instance where the difference is significant.

		Algorithm									
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**AD-FCM**	**AWA-FCM**
Synth1	10%	88.98	89.86	88.98	93.12	94.33	93.01	95.90	97.06± 0.03	95.58±0.00	**97.74±0.03**
	20%	79.41	82.85	78.95	86.22	89.66	87.22	88.04	95.94± 0.05	94.79±0.00	**96.80±0.02**
	30%	71.62	78.14	71.44	79.86	85.99	81.37	58.15	95.46± 0.09	93.64±0.00	**96.14±0.05**
	40%	67.16	74.94	67.10	75.31	82.15	77.10	44.95	94.28± 0.10	92.92±0.00	**95.53±0.05**
	50%	62.77	71.79	63.34	71.38	78.93	72.83	43.12	94.07± 0.07	92.69±0.00	**95.07±0.01**
	60%	59.78	70.42	60.64	67.94	76.36	72.67	29.27	94.17± 0.10	92.99±0.00	**95.56±0.03**
	70%	57.25	68.61	58.12	64.82	73.33	70.26	44.69	93.19± 0.14	90.55±0.00	**94.11±0.07**
	80%	55.29	66.27	56.10	62.87	69.89	67.67	30.49	92.70± 0.04	88.11±0.00	**93.56±0.03**
Synth2	10%	97.85	98.41	97.89	99.11	99.02	98.73	99.18	99.12± 0.01	98.41±0.00	**99.29±0.00**
	20%	92.26	95.05	92.42	96.51	97.85	96.98	97.35	98.67± 0.03	98.46±0.00	**98.85±0.00**
	30%	87.25	91.49	87.72	93.58	96.51	94.87	96.28	98.36± 0.03	98.18±0.00	**98.59±0.01**
	40%	81.72	89.21	82.85	89.11	93.49	92.02	74.52	98.13± 0.01	97.43±0.00	**98.30±0.01**
	50%	76.60	86.52	78.22	85.39	91.32	89.71	82.32	97.64± 0.05	97.07±0.00	**97.97±0.02**
	60%	74.44	85.53	76.40	83.65	90.05	88.70	74.51	97.38± 0.01	96.81±0.00	**97.80±0.01**
	70%	71.04	83.49	73.47	80.16	86.40	86.54	73.24	97.68± 0.03	96.67±0.00	**98.02±0.01**
	80%	68.52	81.58	71.02	77.76	85.56	85.59	73.89	97.18± 0.02	96.32±0.00	**97.44±0.01**
Synth3	10%	81.10	93.30	96.04	97.68	97.25	97.15	99.65	99.61± 0.00	99.69±0.00	**99.72±0.00**
	20%	67.89	76.50	69.33	88.37	95.48	94.37	99.19	99.33± 0.01	99.35±0.00	**99.46±0.01**
	30%	52.89	71.85	64.92	76.71	90.76	88.78	98.63	98.92± 0.04	99.00±0.00	**99.16±0.01**
	40%	45.92	67.07	60.81	72.89	86.83	84.82	65.52	98.73± 0.02	98.93±0.00	**99.10±0.00**
	50%	41.31	62.38	57.00	69.78	80.65	78.62	55.60	98.54± 0.06	98.26±0.00	**98.96±0.02**
	60%	38.72	61.55	54.03	68.78	80.23	77.18	54.36	98.56± 0.02	97.91±0.00	**98.83±0.01**
	70%	36.99	59.21	51.58	63.26	78.07	72.38	51.58	97.79± 0.03	97.71±0.00	**98.23±0.02**
	80%	34.49	58.54	49.69	62.40	75.42	70.59	45.22	97.96± 0.08	97.34±0.00	**98.46±0.01**
Synth4	10%	72.59	77.48	85.28	93.80	92.25	93.70	**98.16**	96.49± 0.01	71.96±0.01	97.11±0.01
	20%	54.42	67.71	72.10	72.65	84.55	82.02	92.41	95.16± 0.08	73.13±0.02	**95.94±0.05**
	30%	51.00	60.31	66.59	65.40	65.40	75.39	82.16	93.93± 0.04	71.68±0.03	**94.93±0.04**
	40%	49.36	58.30	61.76	62.01	60.43	76.61	81.68	93.21± 0.03	70.59±0.02	**94.26±0.02**
	50%	48.69	55.94	53.88	57.90	57.28	72.07	81.86	91.89± 0.04	65.81±0.03	**93.23±0.02**
	60%	48.82	54.17	50.62	56.55	56.25	70.21	88.39	90.13± 0.32	68.59±0.03	**92.02±0.12**
	70%	48.04	54.01	48.08	51.60	54.57	68.73	88.55	90.99± 0.15	67.64±0.02	**91.88±0.05**
	80%	47.76	52.40	47.51	47.01	50.01	68.26	85.70	81.70±10.08	66.85±0.01	86.56±4.63
Synth5	10%	85.62	86.90	85.54	90.31	93.15	90.68	57.80	96.84± 0.09	94.55±0.00	**97.38±0.07**
	20%	75.55	79.95	76.33	82.99	88.01	84.38	58.73	95.06± 0.07	93.57±0.00	**95.76±0.01**
	30%	67.21	74.05	69.03	76.47	82.39	78.03	58.25	93.96± 0.11	91.92±0.00	**95.19±0.03**
	40%	62.39	70.96	64.81	72.43	78.87	74.89	59.96	93.09± 0.06	89.98±0.00	**93.43±0.04**
	50%	57.20	66.68	60.11	67.50	73.84	70.19	58.19	90.54± 0.13	87.49±0.00	**91.85±0.09**
	60%	54.73	65.21	57.89	65.19	71.62	68.06	39.66	90.98± 0.15	86.56±0.00	**91.80±0.07**
	70%	52.90	63.78	56.20	63.36	69.29	66.56	58.90	89.99± 0.06	85.84±0.00	**90.85±0.02**
	80%	51.73	62.83	55.17	62.18	68.06	65.67	39.33	88.04± 0.12	84.91±0.00	**89.20±0.05**

Table 4.3: Wilconxon provided p values resulted from the pair-wise comparison of AWA-FCM with other methods for the SynthDB instances. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively shows where AWA-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm								
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**AD-FCM**
Synth1	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	9.8e-07 (+)	9.8e-07 (+)	9.8e-07 (+)	9.8e-07 (+)	9.8e-07 (+)	9.8e-07 (+)	9.8e-07 (+)	1.7e-06 (+)	1.2e-06 (+)
	40%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.7e-06 (+)	1.6e-06 (+)
	50%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	60%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.4e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
Synth2	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.6e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.6e-06 (+)
	50%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.6e-06 (+)
	60%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
	70%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.6e-06 (+)
	80%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.6e-06 (+)	1.5e-06 (+)
Synth3	10%	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.2e-06 (+)	1.4e-06 (+)	1.2e-06 (+)
	20%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.1e-06 (+)	1.5e-06 (+)
	30%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)
	40%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.7e-06 (+)	1.3e-06 (+)
	50%	6.9e-07 (+)	6.9e-07 (+)	6.9e-07 (+)	6.9e-07 (+)	6.9e-07 (+)	6.9e-07 (+)	6.9e-07 (+)	1.2e-06 (+)	6.9e-07 (+)
	60%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)
	70%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.6e-06 (+)
	80%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)
Synth4	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.6e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	3.1e-05 (+)	0.67 (=)	1.9e-06 (+)
Synth5	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.6e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)



(a) SA variance.

(b) SA mean.

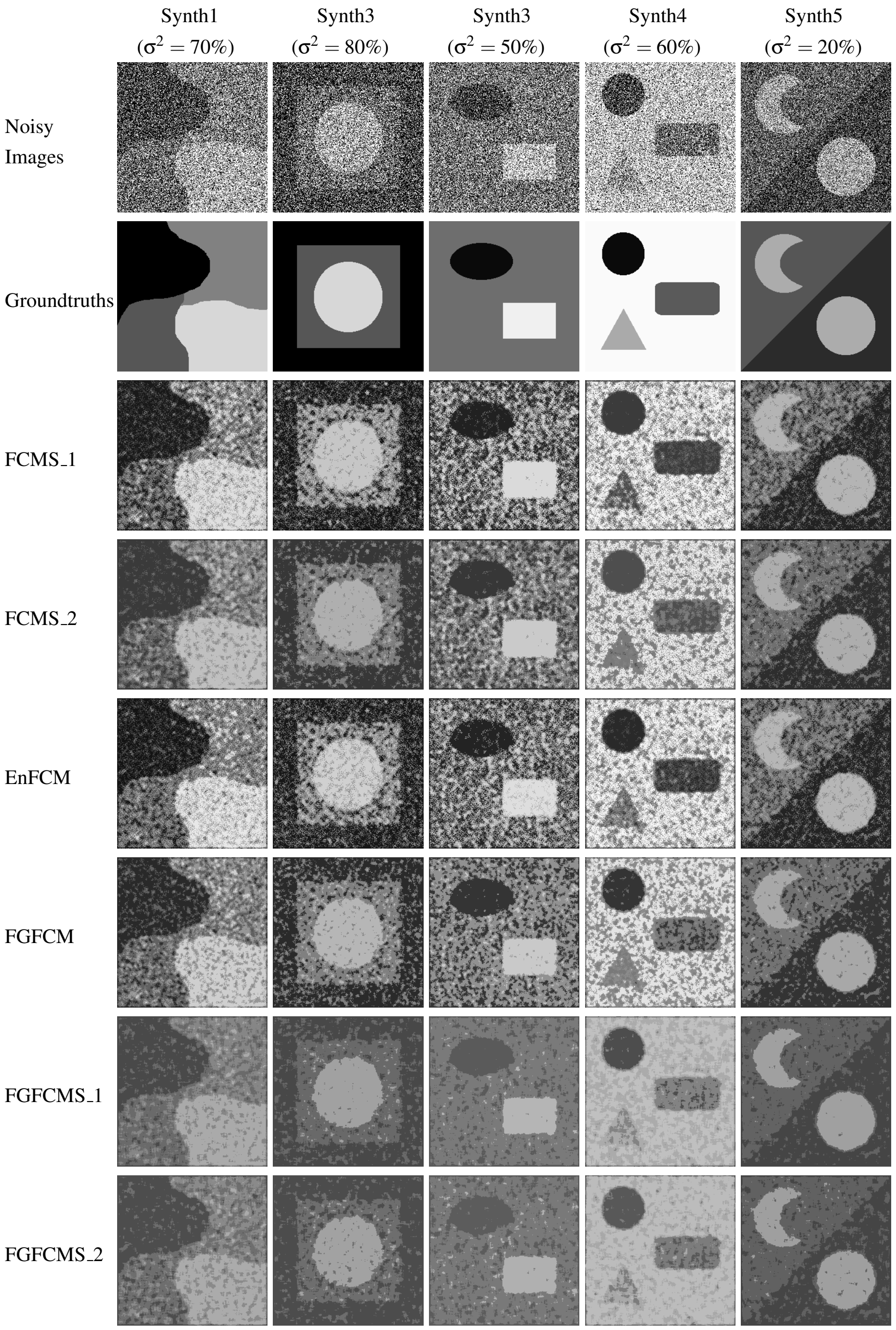
Figure 4.5: Demonstration of different algorithms performances on noise level variation (the SynthDB instances).

S5 but it usually misses one or more regions according to the groundtruth. The result of AD-FCM still suffers from a less-serious over-segmentation issue compared to the previous methods. WS-FCM and AWA-FCM has the most compact segmentation results, and the results look pretty similar. However, having a closer look at the redundant boundaries created around the real edges in images S1 and S4 clearly shows how AWA-FCM produces less redundant boundaries. Generally, AWA-FCM partially returns back to the images some of the missing details related to edges caused by over-smoothing.

4.5.3 BerkDB (Quantitative Analysis)

SA values for the BerkDB instances resulted from each method is provided in Table 4.4, and the p values from the statistical significance test are also provided in Table 4.5. The latter shows that there are an overall three instances that the p value is greater than the significance level. Two belong to the pair-wise comparison of AWA-FCM and WS-FCM for the instances $B86016, \sigma^2 = 70\%$ and $B135069, \sigma^2 = 60\%$, and the third belongs to the comparison of AWA-FCM and AD-FCM for the instance $B196027, \sigma^2 = 40\%$. These are the instances that the proposed AWA-FCM performs not significantly different from WS-FCM and AD-FCM. For the rest of methods and instances in the pair-wise comparison, the performance difference is significant and is determined according to the SA value.

The analysis of the p and SA values indicates that AWA-FCM is always (in all 88 instances) performing better than FCMS_1, FCMS_2, and EnFCM. In comparison to FGFCM, FGFCMS_1, FGFCMS_2, FLICM, WS-FCM, and AD-FCM, the majority of better performances still belong to AWA-FCM by possessing 86, 87, 67, 73, and 75 (out of 88) of significantly better performances. There are an overall 12 instances that WS-FCM is performing better than AWA-FCM. Five of these instances belong to the noisy instances of image 86016. The reason for this is that not much detail is attributed to the boundaries of the one only “bush” region in the image. That is why emphasizing on edge details in the aggregation process only stresses on spurious details and edges which eventually results in slightly worse performance of AWA-FCM compared to WS-FCM. Therefore, one conclusion from the comparison of AWA-FCM and WS-FCM is that if fine details and edges exist around the region boundaries, AWA-FCM could produce mostly better results by addressing the issue of over-smoothed boundaries.


Synth1 ($\sigma^2 = 70\%$)
Synth3 ($\sigma^2 = 80\%$)
Synth3 ($\sigma^2 = 50\%$)
Synth4 ($\sigma^2 = 60\%$)
Synth5 ($\sigma^2 = 20\%$)
Noisy Images
Groundtruths
FCMS_1
FCMS_2
EnFCM
FGFCM
FGFCMS_1
FGFCMS_2

Continued on next page

Figure 4.6 – continued from previous page

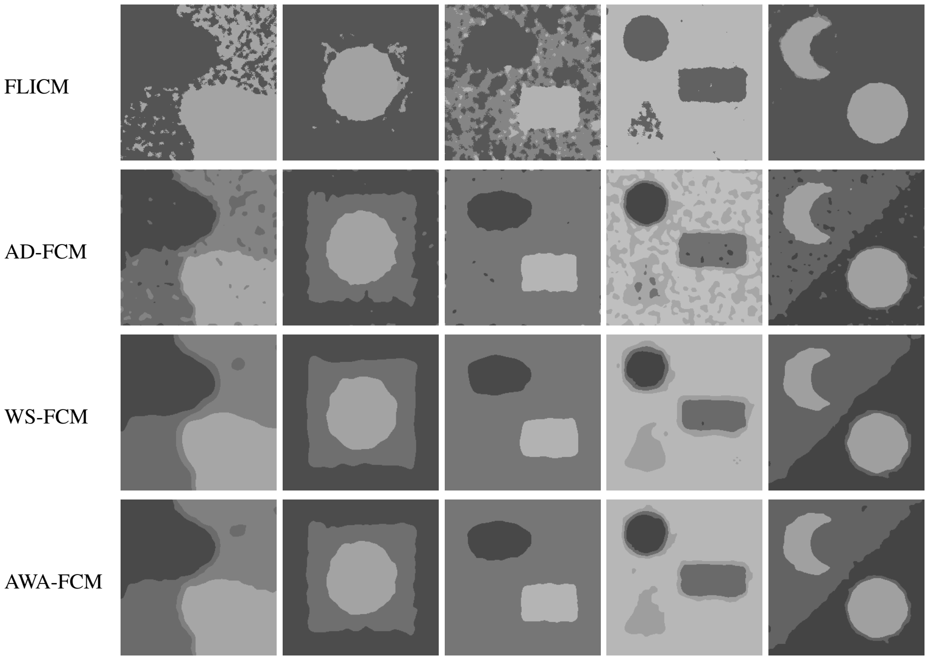


Figure 4.6: Qualitative comparison of the proposed AWA-FCM with FCM_S1, FCM_S2, EnFCM, FGFCM_S1, FGFCM_S2, FGFCM, FLICM, AD-FCM, and WS-FCM on some of the SynthDB instances. Synth1, Synth2, Synth3, Synth4, and Synth5 are corrupted with Gaussian noise with variance of 70%, 80%, 50%, 60%, and 20% respectively.

Table 4.4: SA values for the BerkDB instances. The bold numbers indicate the best overall performance for each instance where the difference is significant.

		Algorithm									
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**AD-FCM**	**AWA-FCM**
B3096	10%	63.55	66.46	68.06	74.04	80.67	83.04	6.13	80.74±0.03	81.57±0.00	**83.26±0.03**
	20%	58.93	62.44	61.03	66.89	73.65	74.45	6.13	77.97±0.02	77.53±0.00	**80.15±0.01**
	30%	57.04	60.84	57.65	63.23	70.29	70.70	6.13	79.40±0.04	76.35±0.00	**81.97±0.04**
	40%	55.57	59.27	54.87	59.98	66.78	65.59	6.13	80.59±0.07	74.65±2.24	**81.50±0.03**
	50%	55.06	58.49	54.91	58.47	64.92	62.43	6.13	78.45±0.01	71.92±0.00	**79.59±0.01**
	60%	54.65	58.38	53.80	58.65	63.42	63.23	6.13	77.48±0.05	74.81±0.00	**78.64±0.02**
	70%	54.08	57.55	53.94	57.66	61.80	61.18	6.13	73.98±0.06	67.50±0.00	**74.13±2.08**
	80%	53.71	57.39	53.10	56.40	62.73	61.20	6.13	75.57±0.03	70.09±0.00	**77.22±0.03**
B8068	10%	94.56	94.80	94.73	94.94	95.22	95.12	**95.78**	95.05±1.17	95.01±0.00	95.38±0.00
	20%	93.60	94.06	93.90	94.40	94.72	94.61	**96.45**	95.16±0.01	94.88±0.00	95.08±0.00
	30%	92.25	93.30	92.98	93.84	94.35	93.97	**96.92**	94.93±0.01	94.80±0.00	94.90±0.00
	40%	90.20	92.74	91.81	93.51	94.33	93.81	**96.96**	94.98±1.19	95.16±0.00	95.29±0.01
	50%	87.97	92.36	90.07	93.23	94.18	93.57	**96.93**	95.40±0.00	95.08±0.00	95.37±0.00

Continued on next page

Table 4.4 – continued from previous page

		Algorithm									
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**AD-FCM**	**AWA-FCM**
	60%	85.27	91.17	87.59	92.04	93.72	92.81	77.69	95.07±0.01	94.64±1.74	**95.27±0.00**
	70%	82.27	89.89	84.47	90.66	93.03	92.02	77.69	95.12±0.02	94.93±0.00	**95.28±0.00**
	80%	80.46	89.37	82.31	89.81	92.73	91.54	77.69	94.87±0.01	94.69±0.00	**94.94±0.00**
B42049	10%	92.98	93.46	93.34	94.04	93.38	93.79	95.15	94.64±0.01	**95.23±0.00**	94.89±0.01
	20%	88.83	90.87	90.70	92.70	92.79	92.65	94.02	93.40±0.01	**94.60±0.00**	94.07±0.00
	30%	84.09	89.13	86.73	91.81	92.50	92.08	94.53	93.24±0.00	**94.64±0.00**	94.24±0.00
	40%	78.97	86.15	81.55	88.72	90.90	89.92	19.09	92.36±0.01	**93.78±0.00**	93.38±0.00
	50%	75.56	84.18	77.37	86.63	90.31	89.27	19.09	92.54±0.11	93.22±1.58	**93.51±0.04**
	60%	72.97	81.88	74.13	83.44	88.77	86.74	19.09	90.64±0.09	**92.80±0.00**	91.96±0.05
	70%	71.45	81.31	72.68	82.41	87.78	86.31	19.09	91.52±0.05	**92.88±0.00**	92.29±0.03
	80%	69.66	79.04	70.59	78.54	85.29	83.61	19.09	89.30±0.05	**92.66±0.00**	91.10±0.04
B167062	10%	79.13	79.68	85.42	97.92	97.16	81.66	**99.08**	98.54±0.00	82.43±5.52	98.68±0.00
	20%	77.43	79.21	82.03	79.52	97.04	81.68	**99.19**	98.23±0.01	80.99±5.06	98.38±0.00
	30%	76.92	78.87	81.69	78.32	76.14	80.34	**99.24**	97.78±0.01	79.83±0.01	98.01±0.01
	40%	76.11	77.77	79.81	77.17	74.39	80.02	**99.23**	97.81±0.00	78.32±0.00	97.97±0.00
	50%	75.60	77.21	78.63	76.06	73.42	80.47	**99.27**	97.60±0.01	80.61±5.04	97.78±0.00
	60%	74.64	77.10	76.99	75.75	88.21	81.03	**98.80**	97.51±0.01	78.55±4.32	97.41±0.01
	70%	73.67	76.45	75.14	74.50	71.25	80.64	**98.93**	97.53±0.00	80.46±5.15	97.76±0.00
	80%	72.58	75.61	73.56	74.51	71.11	79.92	**98.93**	97.16±0.01	78.67±5.86	96.09±4.04
B86016	10%	85.75	86.62	88.58	92.21	93.73	92.71	**99.09**	98.46±0.02	96.31±0.00	98.37±0.01
	20%	76.81	80.21	79.30	86.78	90.70	88.64	16.36	**97.65±0.01**	95.98±0.00	96.85±2.68
	30%	71.68	76.25	72.80	80.45	86.48	83.51	16.36	**97.76±0.02**	95.28±0.00	97.38±0.01
	40%	69.44	75.43	70.95	78.20	84.64	82.00	16.36	96.94±2.86	95.10±0.00	**97.11±0.04**
	50%	67.24	73.56	67.63	75.02	83.35	79.75	16.36	**98.44±0.01**	96.09±0.00	98.37±0.01
	60%	65.75	72.43	66.10	73.31	80.68	78.14	16.36	**97.84±0.00**	94.87±0.00	97.63±0.00
	70%	63.92	69.97	64.21	69.92	77.25	73.41	16.36	**97.13±0.01**	93.18±0.00	96.55±3.14
	80%	63.32	70.21	63.88	69.91	76.07	74.34	16.36	94.93±3.11	92.14±0.00	**95.73±0.03**
B196027	10%	73.62	75.25	76.34	78.23	79.70	80.38	**90.42**	79.55±0.02	80.04±0.00	80.42±0.01
	20%	67.89	70.80	70.11	74.12	76.61	77.32	11.57	79.79±0.05	78.73±0.00	**80.26±0.02**
	30%	65.12	68.67	66.27	71.24	75.21	74.64	11.57	78.78±1.78	78.30±0.00	**79.87±0.02**
	40%	62.99	67.39	63.80	69.02	73.96	72.83	11.57	76.58±0.20	77.92±0.00	77.91±0.05
	50%	61.51	65.88	62.54	66.95	71.91	70.44	11.57	80.18±0.02	77.54±0.00	**80.36±0.06**
	60%	60.52	65.28	61.13	65.74	71.09	69.83	11.57	79.87±0.02	79.03±0.00	**80.28±0.02**
	70%	59.39	64.34	59.91	64.49	70.24	67.66	11.57	76.38±1.48	76.72±0.00	**77.78±0.03**
	80%	59.05	63.80	59.04	63.40	68.64	66.68	11.57	**80.29±0.14**	75.72±1.56	80.14±0.12
B24063	10%	91.41	92.76	91.52	94.15	94.77	94.91	80.52	95.93±0.01	96.34±0.00	**96.60±0.00**
	20%	85.99	88.53	85.99	89.82	92.24	91.51	56.98	95.07±0.12	95.51±0.00	**95.70±0.00**
	30%	81.95	85.50	81.77	86.70	90.36	88.46	53.42	95.74±0.00	95.66±0.00	**96.36±0.00**
	40%	79.27	83.65	78.99	84.02	87.79	86.19	58.70	94.89±0.17	94.62±0.00	**95.49±0.05**
	50%	76.42	81.39	76.11	81.86	86.08	83.39	58.70	94.06±0.06	93.99±0.00	**94.68±0.01**
	60%	74.74	80.45	74.39	80.23	84.18	82.20	58.70	93.75±0.06	93.28±0.00	**94.52±0.01**
	70%	74.40	80.34	74.09	79.77	84.20	81.61	58.70	94.51±0.01	93.71±0.00	**94.79±0.01**
	80%	71.89	78.65	71.73	77.45	81.54	79.93	58.70	93.34±0.06	92.37±0.00	**94.15±0.06**
	10%	93.15	92.63	93.34	94.88	95.07	93.79	**98.46**	96.56±0.03	96.58±0.00	96.88±0.02

Continued on next page

B253036

Table 4.4 – continued from previous page

		Algorithm									
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**AD-FCM**	**AWA-FCM**
	20%	88.51	89.38	89.08	91.98	93.61	91.67	**97.60**	95.91±0.01	96.00±0.00	96.46±0.01
	30%	84.94	87.91	85.57	90.07	91.98	90.49	29.45	94.73±0.01	95.12±0.00	**95.54±0.01**
	40%	81.94	86.23	82.61	87.70	90.58	88.66	29.45	95.77±0.04	94.95±0.00	**95.97±0.01**
	50%	78.68	84.00	78.87	84.78	88.19	86.16	29.45	93.54±0.00	93.28±0.00	**93.76±1.12**
	60%	76.79	83.23	77.18	83.17	87.17	85.84	29.45	94.70±1.35	93.59±0.00	**94.81±1.31**
	70%	75.35	82.45	75.84	81.87	86.48	84.72	29.45	94.45±0.05	93.50±0.00	**94.91±0.04**
	80%	73.22	81.06	73.54	79.86	84.57	83.11	29.45	93.96±0.02	92.87±0.00	**94.13±0.01**
B147091	10%	87.83	88.55	87.90	88.72	88.81	89.40	88.88	89.43±0.02	89.51±0.00	**89.68±0.01**
	20%	85.12	86.74	85.12	87.23	88.11	88.25	65.41	89.06±0.06	89.40±0.00	**89.56±0.05**
	30%	82.34	84.69	82.23	85.03	86.78	86.01	42.74	88.21±0.02	88.53±0.00	**88.66±0.01**
	40%	80.74	84.15	80.92	84.25	86.38	86.01	42.74	88.61±0.07	89.06±0.00	**89.06±0.03**
	50%	78.47	82.72	78.36	82.64	85.40	84.13	42.74	88.84±0.00	89.19±0.00	**89.49±0.00**
	60%	76.37	81.41	76.26	81.10	84.26	83.02	42.74	88.31±0.01	88.15±0.00	**88.65±0.00**
	70%	74.87	80.71	74.68	79.93	83.55	82.06	42.74	87.50±0.05	87.38±0.00	**87.69±0.02**
	80%	73.53	79.50	73.46	78.81	82.57	80.60	42.74	88.34±0.01	88.15±0.00	**88.78±0.02**
B108073	10%	78.11	77.06	80.78	82.86	83.68	80.08	**87.13**	82.41±0.16	83.39±1.66	83.56±0.04
	20%	72.42	73.38	75.01	78.57	79.73	76.00	**86.22**	79.63±0.10	81.47±0.00	80.89±0.11
	30%	69.52	71.56	71.44	75.79	77.13	74.20	77.45	79.97±0.19	80.87±0.00	**81.44±0.08**
	40%	67.13	70.31	69.26	72.70	74.92	73.32	77.45	79.99±0.02	**81.49±0.00**	81.01±1.22
	50%	65.76	69.00	67.36	70.59	73.98	70.91	77.45	80.21±0.01	80.77±1.37	**81.73±0.01**
	60%	64.10	67.66	65.37	68.64	70.67	69.32	77.45	76.21±0.85	**77.63±1.51**	77.60±0.06
	70%	63.37	67.62	64.60	67.77	70.51	69.26	77.45	80.05±0.14	79.56±0.00	**81.26±0.12**
	80%	63.05	67.47	64.32	67.33	69.55	69.16	77.45	81.40±1.21	80.48±0.00	**82.88±0.03**
B135069	10%	58.28	60.61	59.40	64.99	69.70	**72.74**	5.23	66.49±0.01	67.20±0.00	67.92±0.00
	20%	55.19	57.95	55.79	60.00	63.99	66.68	5.23	**68.13±0.04**	63.77±0.00	67.70±0.02
	30%	53.39	56.13	51.88	56.99	61.40	63.04	5.23	61.01±0.61	62.56±0.91	**64.68±0.01**
	40%	52.86	56.10	52.25	56.17	62.44	61.20	5.23	60.13±0.05	63.03±0.00	**63.51±0.97**
	50%	52.52	55.63	51.99	55.54	59.66	59.62	5.23	63.68±0.05	63.70±0.00	**65.63±1.19**
	60%	52.27	55.58	50.86	54.43	59.86	59.29	5.23	64.91±0.33	64.71±0.00	64.94±0.10
	70%	52.47	55.69	51.13	54.73	60.11	58.89	5.23	61.36±0.04	**64.79±0.00**	64.40±0.06
	80%	51.95	55.33	51.00	54.58	60.59	58.95	5.23	67.29±1.23	63.12±0.00	**69.43±0.04**

The analysis on the overall best performer shows that AWA-FCM is the overall best performer by holding the highest number of best performances. AWA-FCM is the best and second-best performer on 49 and 25 instances respectively. The overall second-best performer is FLICM being the best and second-best performer on 19 and two instances respectively, and the overall third-best performer is AD-FCM being the the best and second-best performer on 10 and 15 instances respectively.

To have a look at the performance of different methods over different noise levels, Fig. 4.7 is provided in which the mean and variance of SA values from all the instances

Table 4.5: Wilconxon provided *p* values resulted from the pair-wise comparison of AWA-FCM with other methods for the BerkDB instances. The bold numbers indicate *p* values greater than the significance level. **+**, **-** and **=** respectively shows where AWA-FCM performs significantly better, significantly worse or not significantly different.

		Algorithm								
Img.	Vol.	FCM_S1	FCM_S2	EnFCM	FGFCM	FGFCM_S1	FGFCM_S2	FLICM	WS-FCM	AD-FCM
B3096	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	1.9e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B8068	10%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (-)	1.6e-06 (+)	1.5e-06 (+)
	20%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.7e-06 (-)	1.6e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (-)	1.6e-06 (-)	1.2e-06 (+)
	60%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.7e-06 (+)	1.6e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	9.9e-07 (+)	9.9e-07 (+)	9.9e-07 (+)	9.9e-07 (+)	9.9e-07 (+)	9.9e-07 (+)	9.9e-07 (+)	1.7e-06 (+)	9.9e-07 (+)
B42049	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (+)	1.6e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.6e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.045 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
B167062	10%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (-)	1.6e-06 (+)	1.6e-06 (+)
	20%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (+)	1.7e-06 (+)
	40%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (-)	1.7e-06 (+)	1.6e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	0.0027 (-)	1.9e-06 (+)
B86016	10%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.7e-06 (-)	1.6e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.8e-06 (+)	3e-05 (+)	1.8e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	3e-05 (+)
	30%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.7e-06 (-)	1.6e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	1.7e-06 (+)
	50%	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.3e-06 (+)	1.6e-06 (-)	1.3e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	70%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	0.43 (=)	3e-05 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)

Continued on next page

Table 4.5 – continued from previous page

		Algorithm								
Img.	Vol.	FCM_S1	FCM_S2	EnFCM	FGFCM	FGFCM_S1	FGFCM_S2	FLICM	WS-FCM	AD-FCM
B196027	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.26 (=)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.3e-06 (-)	1.7e-06 (+)
B24063	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.5e-06 (+)	1.4e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.7e-06 (+)	1.5e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B253036	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.4e-06 (+)	1.7e-06 (+)	1.4e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	3.1e-05 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.8e-05 (+)	3.1e-05 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B147091	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00035 (+)
	50%	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.5e-06 (+)	1.7e-06 (+)	1.5e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B108073	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	3.1e-05 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	3.1e-05 (+)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.1e-06 (+)	1.7e-06 (+)	0.00036 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	10%	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (+)	1.6e-06 (-)	1.6e-06 (-)	1.6e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)

Continued on next page

Table 4.5 – continued from previous page

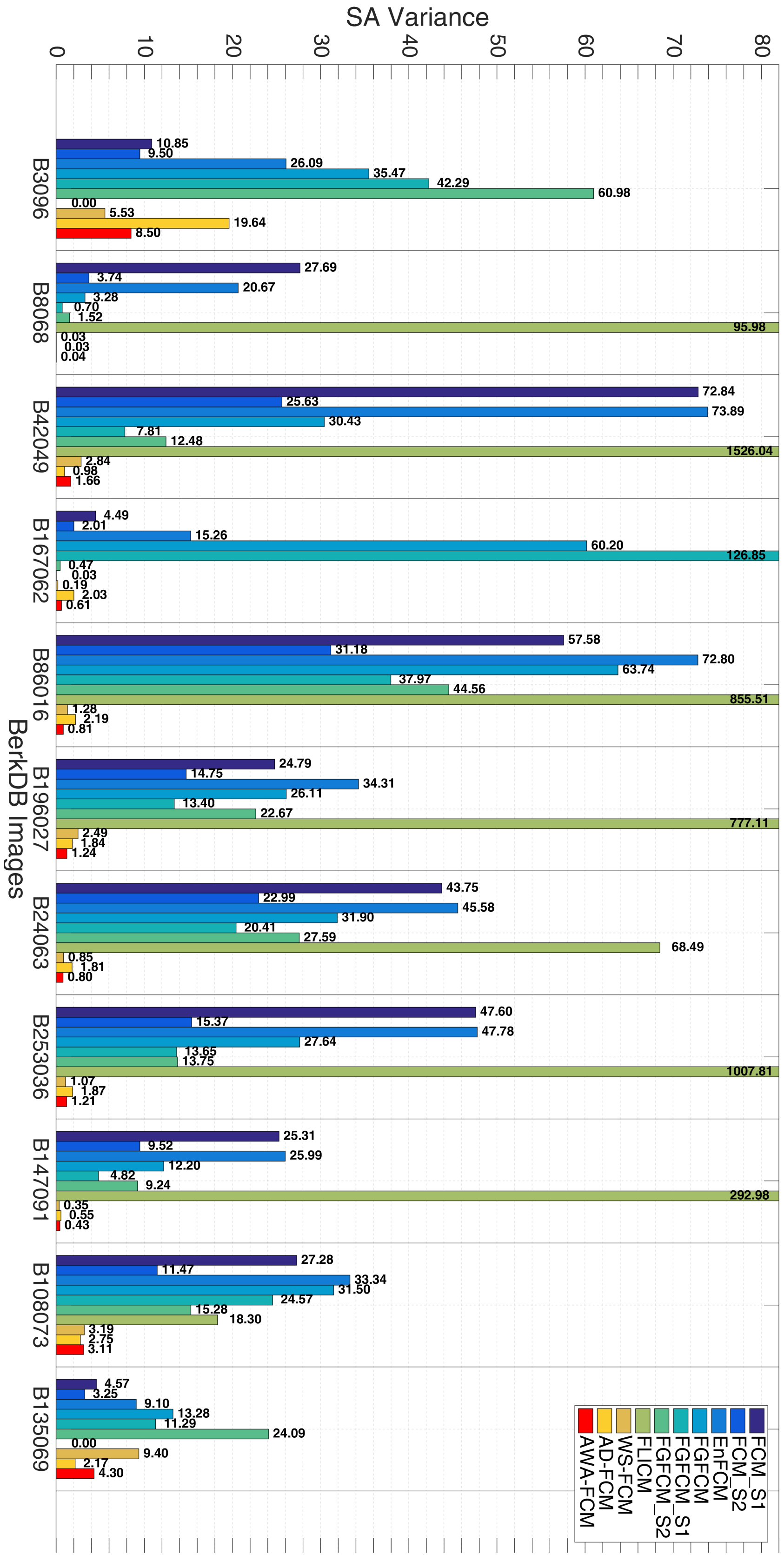
		Algorithm								
Img.	Vol.	FCM_S1	FCM_S2	EnFCM	FGFCM	FGFCM_S1	FGFCM_S2	FLICM	WS-FCM	AD-FCM
B135069	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	3.1e-05 (+)	1.7e-06 (+)	1.9e-06 (+)	3.1e-05 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	3.1e-05 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.98 (=)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)

for each image in BerkDB is plotted. The value of variance or mean has been mentioned above each bar for better comparison between close performances. In the plot related to SA variance, if the variance is too large, to have a closer look at the small variances, the whole bar is not shown, but the value has been mentioned on it. Also, for a better comparison in the mean plot, the bars related to vary small mean values are not shown, but the value has been mentioned on the location of bar. The proposed AWA-FCM is the red bar in both plots. The variance of SA performance in Fig. 4.7(a) shows that AWA-FCM is generally having small variances over noise variation. AWA-FCM is having three (out of 11) smallest, and five further second-smallest SA variances. Due to very small and unusual SA mean of FLICM for images $B3096$ and $B135069$, the zero variance of this method has not been considered in this comparison. From this perspective, WS-FCM has the highest number of smallest SA variance: four as smallest, and three more as the second-smallest. The next best performer is AD-FCM having four smallest and one second-smallest SA variance(s). Therefore, AWA-FCM is the third-best performer considering the number of lowest SA variances.

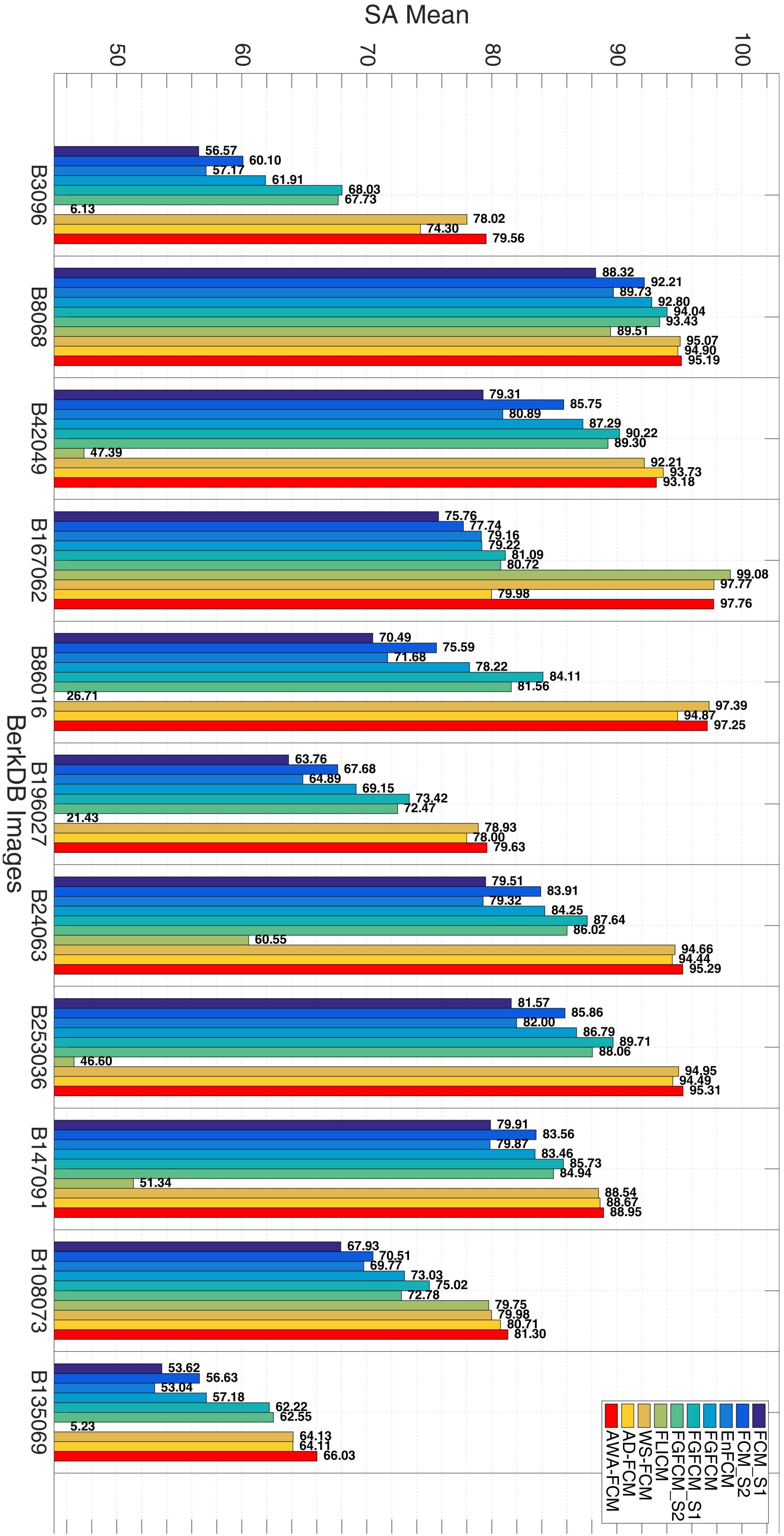
The plot related to the mean of SA performances, Fig. 4.8(b), shows that AWA-FCM is the performer with the highest mean values in the majority of the images. It has the largest mean performance on eight images and the second-largest on two more images. The next best performers from this viewpoint are WS-FCM and AD-FCM. WS-FCM has one largest and seven further second-largest mean performance(s), and AD-FCM has one largest and two further second-largest mean performance(s).

BerkDB (Qualitative Analysis)

For visual comparison in BerkDB, Fig. 4.8 is provided in which an instance from each of the 11 images in the dataset is selected, and then the segmentation results from all



(a) SA variance.



(b) SA mean.

Figure 4.7: Demonstration of different algorithms performances on noise level variation (the BerkDB instances).

10 methods are presented. Like the previous dataset, due to the volume of noise, over segmentation is a serious problem, and a lot of spurious boundaries and segmented regions exist in the results of FCM_S1, FCM_S2, EnFCM, FGFCM, FGFCM_S1 and FGFCM_S2 algorithms. FLICM can sometimes provide highly accurate segmentation results as in the instances of $B253036$ and $B167062$, and sometimes completely misses the main regions. AD-FCM performs well in retaining edge information, however it produces redundant segments which interfere with accurate region segmentation. WS-FCM although successful at homogeneous region segmentation and noise removal, it also eliminates fine details around the edges. AWA-FCM on the other hand makes a good compromise between the number of redundant regions and preserving the details. As an example, one could see how the tree body is back to the segmentation results of AWA-FCM in the $B253036$ instance compared to WS-FCM result, or how the overs-smoothed treetop region in the $B147091$ instance is having more details around its boundary. One visually negative point regarding AWA-FCM segmentation results is that it generally produces more redundant regions which could be problematic in some advanced level applications.

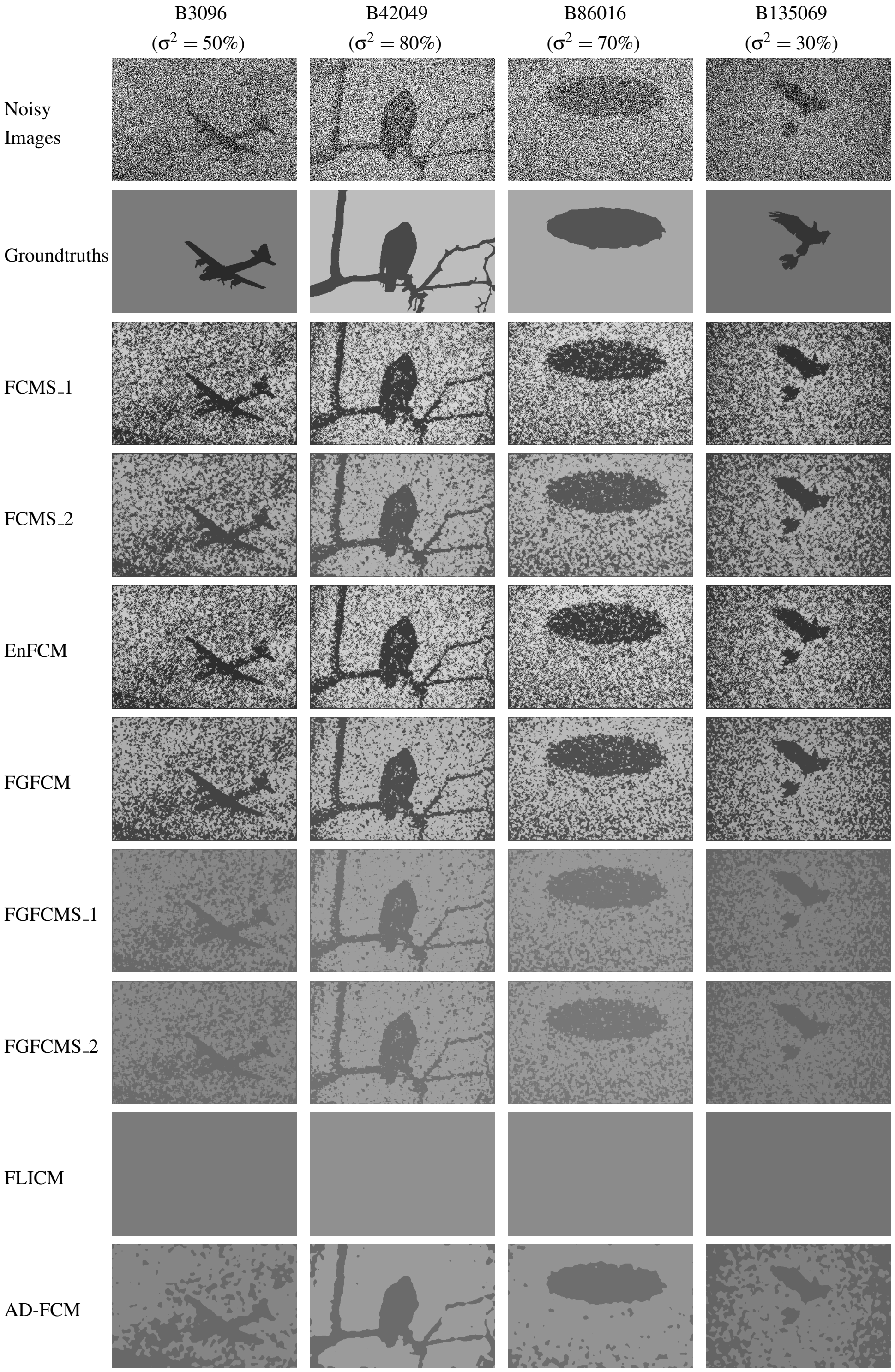
4.6 Chapter Summary

By taking advantages of wavelet-based thresholding and gradient-based denoising in the spatial domain, we introduced an FCM-based, detail-preserving severely noisy image segmentation algorithm. The following are the main highlights of this chapter:

- The wavelet domain has the ability to add back lost details of an image. An edge-preserving denoising algorithm such as AD can be used to supply those details.

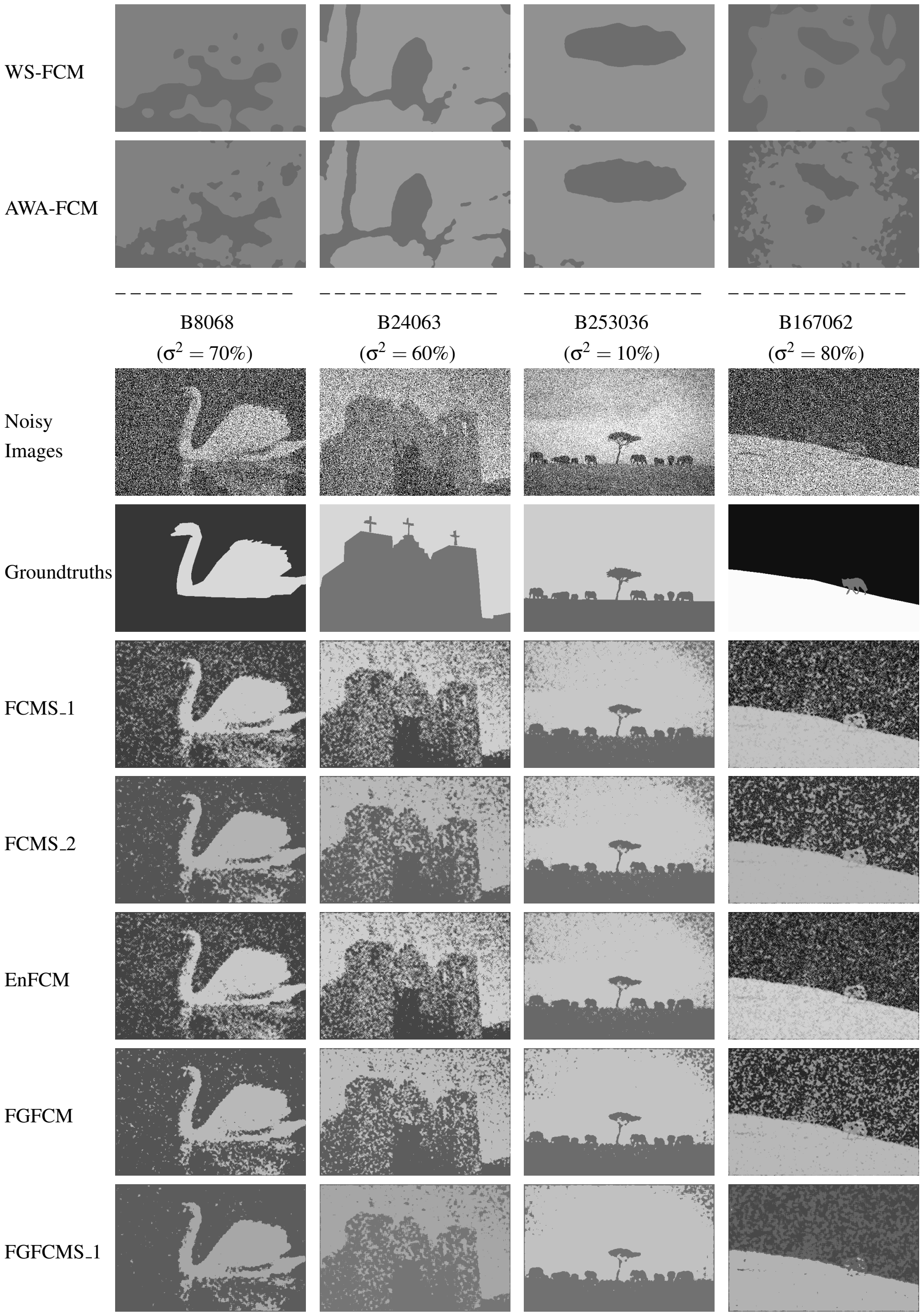
- While segmentation based on wavelet thresholding alone may not be able to respect the boundary details, and AD based segmentation may not be successful either in homogeneous/coherent region segmentation, the combination of the two denoising results in the wavelet domain is able to address the issues.

- Our fusion scheme considers the noise volume and works well in a wide range of noise level variation.



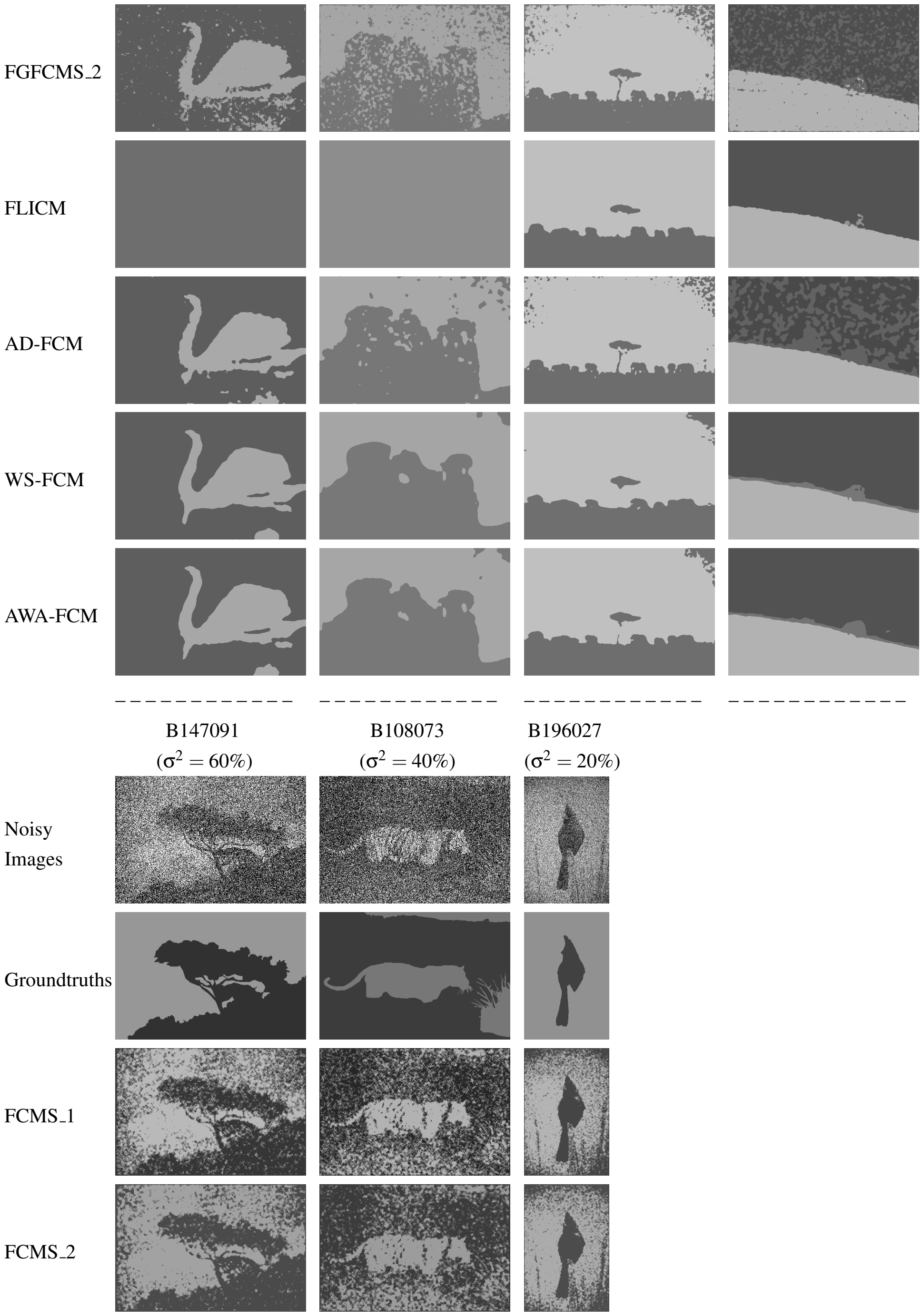
Continued on next page

Figure 4.8 – continued from previous page



Continued on next page

Figure 4.8 – continued from previous page



Continued on next page

Figure 4.8 – continued from previous page

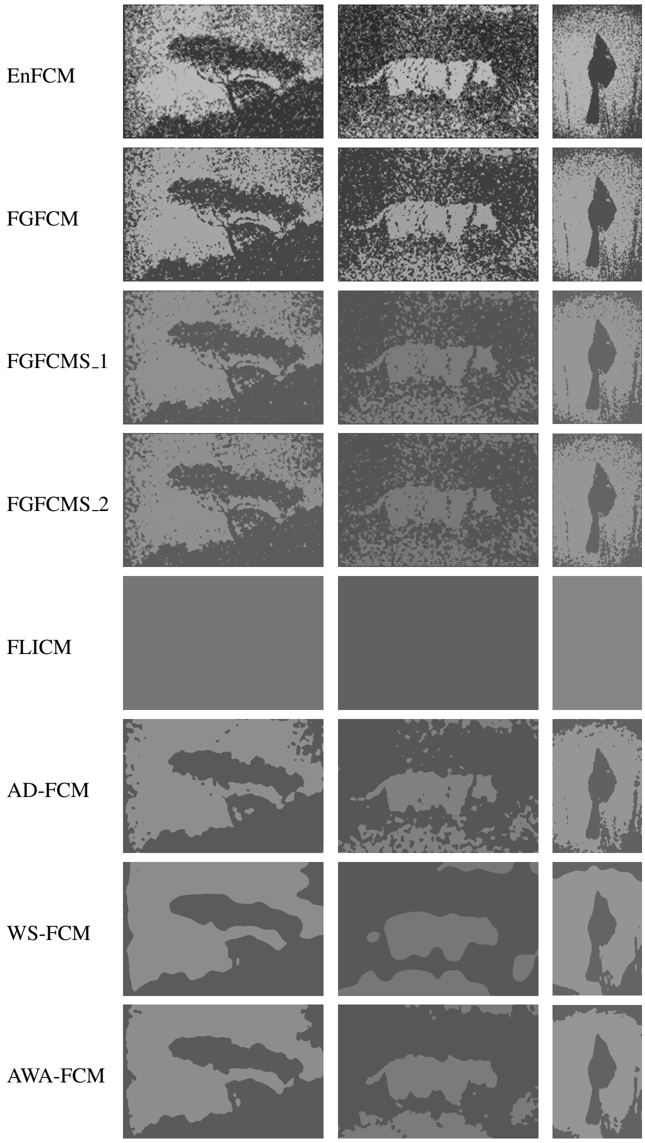


Figure 4.8: Qualitative comparison of the proposed AWA-FCM with FCM_S1, FCM_S2, EnFCM, FGFCM_S1, FGFCM_S2, FGFCM, FLICM, AD-FCM, and WS-FCM on some of the BerkDB instances. The instances are $B3096, \sigma^2 = 50\%$, $B8068, \sigma^2 = 70\%$, $B42049, \sigma^2 = 80\%$, $B167062, \sigma^2 = 80\%$, $B86016, \sigma^2 = 70\%$, $B196027, \sigma^2 = 20\%$, $B24063, \sigma^2 = 60\%$, $B253036, \sigma^2 = 10\%$, $B147091, \sigma^2 = 60\%$, $B108073, \sigma^2 = 40\%$, and $B135069, \sigma^2 = 30\%$.

- A post-processing step based on connected component analysis can be utilized to suppress any side effect of the fusion process.

- Severely noisy images segmentation based on the proposed approach shows greater accuracy and consistency compared to other FCM-based state-of-the-art algorithms.

Chapter 5

Wavelet Packets and Bilateral Filtering for Edge-Preserving Severely Noisy Image Segmentation

5.1 Introduction

Previous chapters showed how feature enhancement in wavelet domain can achieve great performances for FCM-based noisy image segmentation. More precisely, we demonstrated that adaptive soft-thresholding of wavelet coefficients with FCM clustering performance evaluation called WS-FCM, can successfully overcome the problem of FCM-based severely noisy image segmentation. However, soft-thresholding attenuates large coefficients greatly, and the segmented images look over-smoothed around the edges. Chapter 4 addressed this problem with a fusion of denoising results from WS-FCM and a noise-estimating anisotropic diffusion in wavelet domain. This chapter addresses the issue with a different strategy. We use wavelet packets from the family of spatial-frequency transformations which is claimed to provide a richer domain analysis for signal processing.

One serious problem with severely noisy images is that they are not sparse in the wavelet domain, and this makes wavelet-based thresholding less effective. We use Bilateral Filtering as an edge-preserving technique prior to shrinkage of coefficients to make the wavelet domain presentation more sparse. Also, we introduce a new thresholding scheme using smooth sigmoid-based shrinkage [9], which overcomes the over-smoothing problem of soft-thresholding.

Overall, this chapter introduces two algorithms for an FCM-based severely noisy image segmentation. The first algorithm is based on the same approach proposed as in Chapter 3 called WS-FCM, but using wavelet packets as the feature enhancing domain. We investigate which method is providing better results under similar schemes. The second algorithm, uses the first one in combination with image denoising in spatial domain and a new thresholding scheme based on a sigmoid function. We compare the algorithms introduced in this chapter with with WS-FCM proposed in Chapter 3, and AWA-FCM proposed in Chapter 4.

5.1.1 Chapter Goals

Like the previous two chapters, this chapter also addresses noisy image segmentation. The core focus of this chapter is feature manipulation/enhancement in wavelet packet domain. We use denoising strategies in both spatial and spatial-frequency domains, and then try to combine them in way that the image is denoised with higher segmentation accuracy. More specific goals pursued in this chapter are as follow:

- Exploring the difference between wavelets and wavelet packets under similar adaptive thresholding schemes provided by PSO for FCM-based noisy image segmentation.

- Investigating the possibilities of increasing the sparsity of a noisy image in wavelet domain using denoising algorithms in the spatial domain.

- Exploring other thresholding functions, and designing a new scheme to preserve more details in segmentation results.

- Combining the above ideas into a strategy under which the proposed algorithm works in a noise-varying situation.

5.1.2 Chapter Organization

The reminder of this chapter is as follows. Section 5.2 is dedicated to the methodology of the proposed approach in this chapter. Section 5.3 introduces the foundation of the new detail-preserving severely noisy image segmentation approach. Section 5.4 provides the datasets, comparison algorithms, and evaluation metrics. Section 5.5 is devoted to a comprehensive analysis of the results and discussions, and Section 5.6 concludes this chapter.

5.2 Methodology

This chapter uses another family member of the wavelet-based transforms called wavelet packets. Also, an edge-preserving technique called bilateral filtering is used in one of the proposed algorithms. This section provides a brief description of these methods. Their use and specific presentation in our methods is discussed in the next section.

5.2.1 Wavelet Packets

Wavelet Packets (WP), first introduced by Coifman and Wickerhauser [39], is a generalization of wavelet decomposition that provides a richer analysis domain. Wavelet packet atoms are waveforms indexed by three naturally interpreted parameters: position, scale (as in the wavelet decomposition), and frequency. Utilizing wavelet-based counterparts to analyze non-stationary signals is well established, because wavelets are localized in both time and frequency. Although the Discrete Wavelet Transform (DWT) is a common tool for time-frequency decomposition of a signal, the frequency resolution in the DWT is typically considered too coarse for practical time-frequency analysis. On the other hand, the Continuous Wavelet Transform (CWT) is a tool to monitor the whole spectrum of a signal with fine resolution, but is computationally expensive. In the trade-off between DWT and CWT, wavelet packets provides a computationally affordable solution with sufficient frequency resolution.

Wavelets vs Wavelet Packets

The generic procedure in a orthogonal 2D wavelet decomposition is to split the original image/approximation coefficients into four parts. This results in a summation of scaling and wavelet functions whose bandwidths are logarithmically related. That is, the low-frequency contents of an image have narrower bandwidths, and high-frequency contents have wider bandwidths. 2D-DWT results in three subbands of detail coefficients, and a subband of approximation coefficients at a coarser scale. Detail subbands capture the information lost between the successive approximation coefficients of two decompositions. For further decompositions, the new approximation subband is split, but detail coefficients are never considered for splitting for reanalysis. To have finer resolution of the spatial-frequency plane, narrower bands at the high-frequency decompositions have to be created. This more controllable decomposition is provided by wavelet packets at the cost of higher computational complexity.

In 2D wavelet packet decomposition, each detail subband is also split into four parts based on the same approach utilized in splitting of the approximation subbands. That is, wavelet packets are the conventional wavelet transforms in which the detail coefficients are also filtered iteratively. Therefore, wavelet packets decomposition provides a richer domain for image analysis due to better resolution to high-frequency contents.

Wavelet Packets Representation

Wavelet packet also uses the filter bank introduced in Chapter 3 (Fig. 3.1) for wavelet transform at each scale for each subband. Tree analysis (binary in the case of 1D signal, and quad-tree in the case of 2D signal) is utilized to represent wavelet packets since they are compact, informative, and simple to draw compared to their filter block diagrams. Figure 5.1 presents a partial depiction of the full quad-tree of a the three-scale wavelet packet decomposition in which the difference between the quad-trees of wavelet and wavelet packets is also shown. For wavelet packet transform, decomposition starts scale-oriented and then frequency devision of subbands on each set of coefficients is provided. The highest-scale set of approximation coefficients is called a *root node*, and the detail and approximation coefficients resulting from the transform at the bottom of the tree (nodes with no children) are called *leaves*. Each node is a set of coefficients that are the results of an expansion that produces a band-limited piece of its parent node.

In Fig. 5.1, subbands are represented based on the known scaling and frequency spectrum for the corresponding subband. The first letter indicates the subband, the string of other letters indicates the parent node, and the subscript indicates the scale number. For instance, DVH_3 is detail coefficients D, in the third scale from parent node VH_2, which is the detail coefficients V, in the second scale from parent node $H1$, and

$H1$ is the detail coefficients H in the first scale from parent node I.

Full decomposition quad-tree of level p has 4^p leaf nodes. Also, each tree can be presented with different decompositions. For instance, the three following decompositions are valid for the quad-tree in Fig. 5.1:

$$\begin{aligned}
&I = AA_2 \oplus AHA_3 \oplus HHA_3 \oplus VHA_3 \oplus DHA_3 \oplus VA_2 \oplus DA_2 \oplus H_1 \oplus V_1 \oplus D_1 \\
&I = A_1 \oplus AH_2 \oplus HH_2 \oplus VH_2 \oplus DH_2 \oplus V_1 \oplus D_1 \\
&I = A_1 \oplus H_1 \oplus V_1 \oplus D1
\end{aligned} \tag{5.1}$$

where $\oplus$ is the union of coefficients. The number of valid decompositions (union of nodes that can represent a tree with different frequency resolutions), N_d, of a p-scale wavelet packet quad-tree is obtained from:

$$N_d(p+1) = \Big[N_d(p)\Big]^4 + 1 \tag{5.2}$$

where $N_d(1) = 1$. Therefore, for a five-scale quad-tree there are 1024 leaf nodes in case of a full decomposition, and 5.6081×10^{78} possible decompositions to represent the quad-tree. Depending on the optimality criterion, each one of these decompositions could be considered suitable. How to find a good decomposition for our noisy image segmentation algorithm is discussed in the next section.

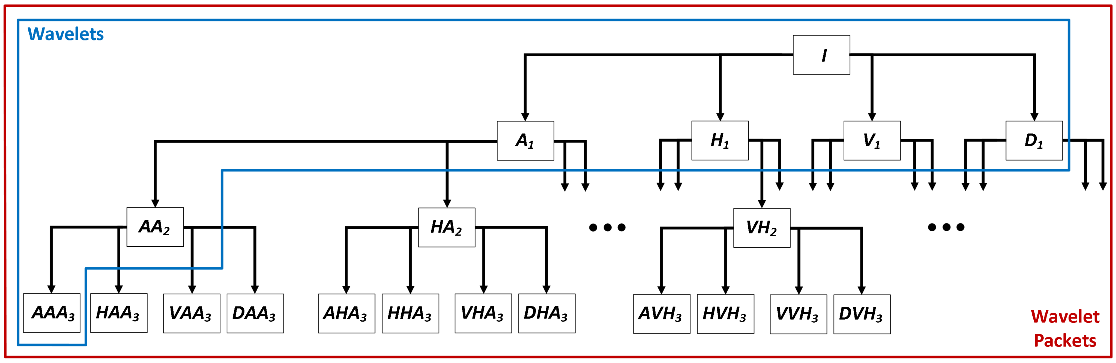


Figure 5.1: Depiction of the difference between wavelets and wavelet packets decompositions in a 2D three-scale transform. Blue box shows the quad-tree for a wavelet transform, and the red box shows the partial quad-tree of wavelet packet transform.

5.2.2 Bilateral Filtering

Bilateral filtering (BF) is an image smoothing method that goes beyond traditional filters by taking into account distance and similarity of neighboring pixels. BF preserves the edge information by using a nonlinear combination of neighboring pixels' intensity values based on spatial closeness and photometric similarity. Priority is given to near values in both *domain* and *range*. By domain here we mean distance between pixels' spatial location, and by range we mean distance between pixels' intensity values.

Kernel-based filtering is a fundamental traditional approach in image processing. For example, one very common filtering technique is the Gaussian low-pass filtering via kernel filtering which calculates a weighted average of pixels in a neighborhood while the weights decrease when the distance from the center of kernel increases. Slow spatial variation is the reason behind this averaging. In other words, images change slowly over the space that means neighborhood pixels are more likely to have similar information. Since noise is less correlated with these neighboring pixels, averaging them removes the noise while preserves the signal. However, the main challenge in local averaging filters is how to treat edges so they are not over-smoothed. Gaussian filtering is effective when the dosage of the noise is small. For the images with high levels of noise, it is required to average more pixels in order to suppress the effect of noise, and this causes over-smoothing of edges and corners.

On the other hand, bilateral filtering introduces decaying weights for the kernel based on both similarity and closeness. Two pixels are *close* if they are positioned in a small spatial distance from each other, and are *similar* if they have nearby intensities. Overall, BF is simple, non-iterative and respectful of edges.

In this chapter, we consider a standard bilateral filtering in which both the range and domain filtering are performed using Gaussian kernels [35]. The bilateral filtering of an image $\{f(i) : i \in \mathbb{I}\}$ ($\mathbb{I}$ is a finite rectangular domain of $\mathbb{Z}^2$) is given as:

$$f_{BF}(i) = \frac{\sum_{j\in\Omega} \omega(j) \times g_{\sigma_r}\Big(f(i-j) - f(i)\Big) \times f(i-j)}{\sum_{j\in\Omega} \omega(j) \times g_{\sigma_r}\Big(f(i-j) - f(i)\Big)} \tag{5.3}$$

where G_{σ_r} is the range kernel and G_{σ_s} is the spatial kernel:

$$G_{\sigma_r}(t) = exp\left(-\frac{t^2}{2\sigma_r^2} \right) \tag{5.4}$$

$$G_{\sigma_s}(j) = exp\left(-\frac{\parallel j \parallel^2}{2\sigma_s^2} \right) \qquad (j \in \Omega). \tag{5.5}$$

where σ_r is the photometric spread (or width of range Gaussian) determining the required degree of combination of pixel values. Range filtering is used to restrict the averaging to neighboring pixels whose intensity are similar or close to that of the pixel under consideration. σ_s is the domain spread (width of spatial Gaussian) determining the required amount of low-pass filtering. A larger value of σ_s integrates pixels from farther locations, and therefore, it has more smoothing effects. More details about the functioning of bilateral filtering can be found in [122] and [97].

The domain of the spatial kernel in the preceding equations is a square neighborhood of $\Omega = [-W,W] \times [-W,W]$, where $W = 3\sigma_s$ in our application. For practical settings of W, the calculation of Eq. 5.3 is computationally expensive [35]. Many research works have been proposed to address this issue by presenting an approximation of the filter while providing a trade-off between the speed and quality of the approximation [49, 97, 34, 35]. One category of these methods in particular decomposes bilateral filtering into a series of spatial filtering by approximating Eq. 5.4 with polynomial and trigonometric functions [103, 117, 36]. In our approach we use the fast and accurate bilateral filtering [35] that proposes a new approximation to Eq. 5.4 and converts bilateral filtering into a series of spatial convolutions. This approach was shown to be simpler, faster and more accurate than other state-of-the-art bilateral algorithms [35].

5.3 The Proposed Algorithms

In this section, based on the idea of adaptive wavelet shrinkage, we introduce two algorithms that use wavelet packets as another member of the family of spatial-frequency transformations for feature manipulation/enhancement. The first algorithm follows a similar pathway of adaptive shrinkage of wavelet coefficients (WS-FCM) introduced in Chapter 3 but using wavelet packets. We named this algorithm Wavelet Packet Shrinkage-based FCM or WPS-FCM. The introduction of WPS-FCM provides an interesting comparison between wavelets and wavelet packets in an adaptive shrinkage scheme for FCM-based noisy image segmentation. Since it has been claimed in the literature that wavelet packets provide a richer domain for signal analysis, we put this idea to the test in an FCM-based noisy image segmentation scheme. The second algorithm uses the feature enhancing system introduced in WPS-FCM based on adaptive wavelet packet shrinkage with two main differences. First, a preprocessing step is added for partially denoising the noisy image. Second, based on the idea of Smooth Sigmoid-Based Shrinkage (SSBS) [9], we incorporate a new thresholding function that preserves more details from a noisy image. We call the second algorithm, Bilateral filtered Wavelet Packet Shrinkage-based FCM or BWPS-FCM.

5.3.1 WPS-FCM

WPS-FCM follows the same approach as in WS-FCM. The block-diagram of the proposed WPS-FCM is given in Fig. 5.2. First, the input image is transformed using wavelet packets. Like WS-FCM, we use a five-scale transformation. Next is the adaptive thresholding of nodes (subbands) using PSO. Recalling from the introduction to wavelet packets, a full five-scale transformation will lead to a quad-tree with 1024 leaf nodes. This number of nodes is 64 times more than that of a wavelet tree. Thresholding such a large number of subbands is computationally very expensive in the PSO-based scheme proposed for WS-FCM. One way to reduce the number of leaf nodes is to select a sub-tree based on other valid decompositions of the quad-tree. In a five-scale wavelet packet transform, the number of possible decompositions is so large that it is not possible to examine all of them. It is possible to find good decompositions using a greedy algorithm based on a *cost function*. This cost function is designed to determine the cost value for each node and allows the search algorithm to determine whether it is worth including the children of a node into the final decomposition or not.

In a search for simple criterion that is computationally efficient, classical entropy-based criteria are selected. They have been shown to be a good measure of efficiency of a decomposition, and can describe information cost for an accurate representation of a signal [39]. Therefore, classical entropy-based are well suited for such situations. Entropy-based cost function based on the following additive function is a good criterion to efficiently search tree structures for a proper sub-tree:

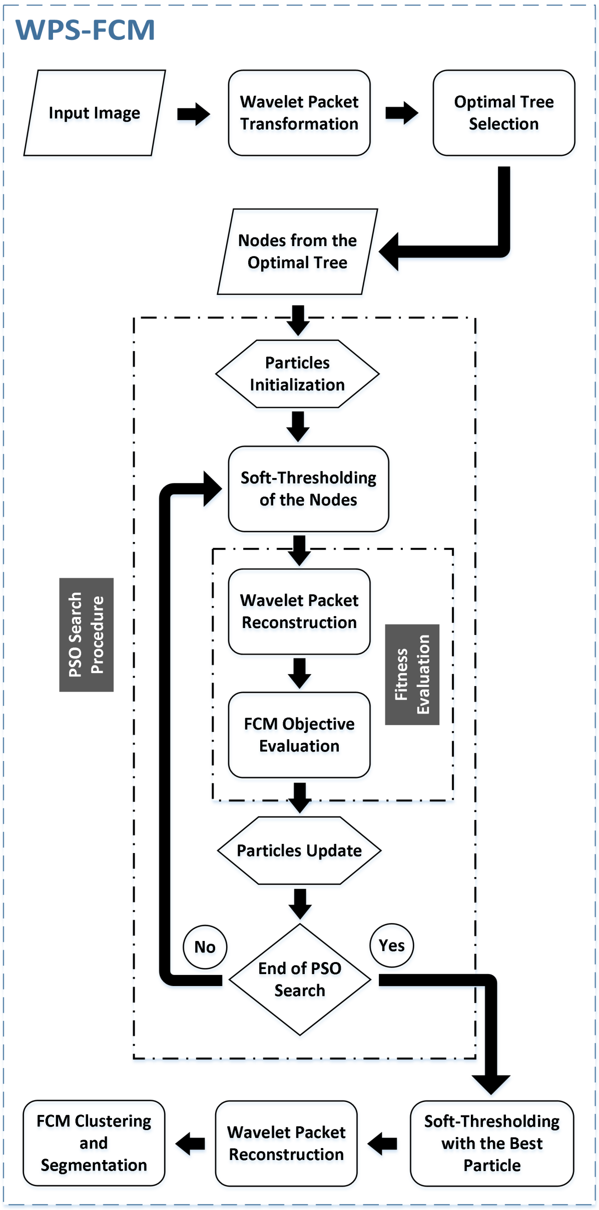

WPS-FCM
Input Image
Wavelet Packet Transformation
Optimal Tree Selection
Nodes from the Optimal Tree
Particles Initialization
Soft-Thresholding of the Nodes
Wavelet Packet Reconstruction
FCM Objective Evaluation
Fitness Evaluation
PSO Search Procedure
Particles Update
End of PSO Search
No
Yes
Soft-Thresholding with the Best Particle
Wavelet Packet Reconstruction
FCM Clustering and Segmentation

Figure 5.2: Block diagram of the proposed WPS-FCM algorithm.

$$E(f) = \sum_{m,n} |f(m,n)| \tag{5.6}$$

where f is the union of all the coefficients of image I, and E provides a possible value for the energy content of f with $E(0) = 0$. Many entropies are available in the literature among which we select [39] in which motivated by *Shannon* entropy

$$E_S(w_i) = -w_{mn}^2 \log(w_{mn}^2) \tag{5.7}$$

then we have:

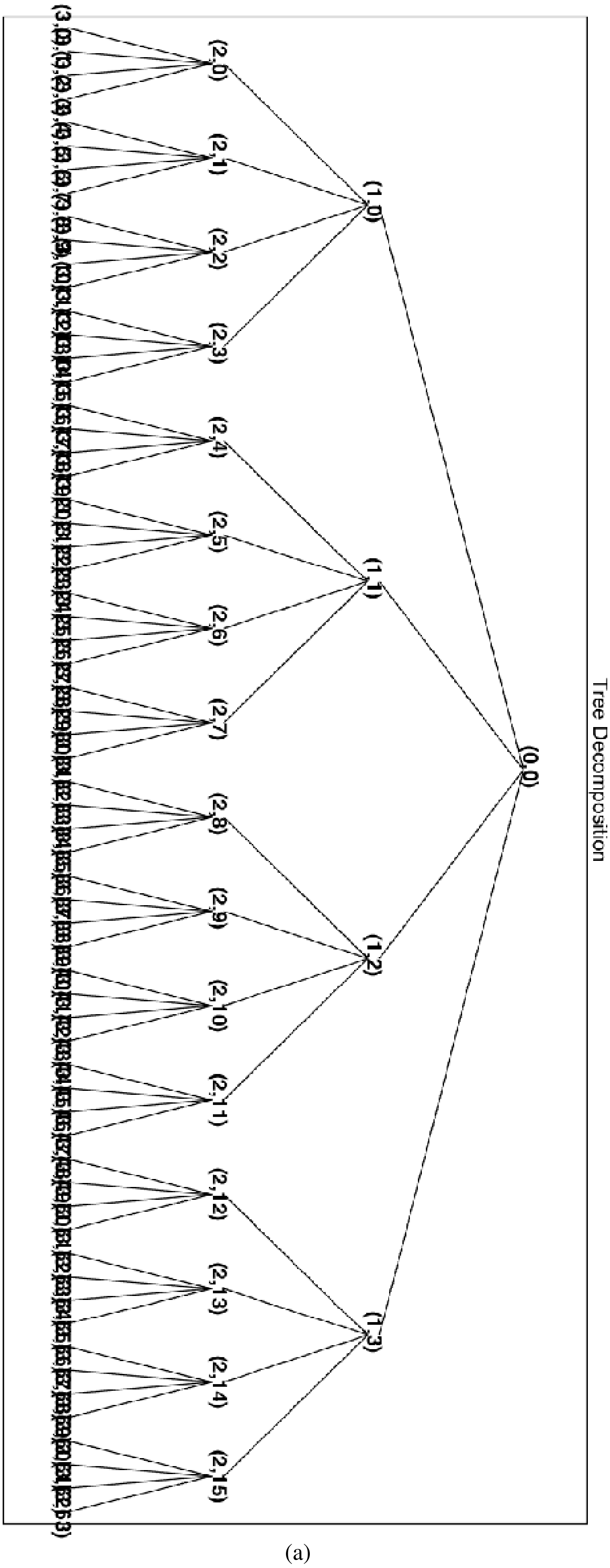
$$E_S(f) = -\sum_{m,n} w_{mn}^2 \log(w_{mn}^2) \tag{5.8}$$

where w_{mn} are the coefficients of f. High values of E_S indicate functions with many non-zero values. On the other hand, low values of E_S shows a cost function with coefficients with zero or near zero values. In a denoising scheme, small coefficients are thresholded or truncated. Therefore, the optimization algorithm in the process of looking for low noise images looks for minimum values of the Shannon entropy-based cost function. Starting from the root node, the energy of a node and its four children is computed. A node is decomposed if and only if the sum of the Shannon entropy of its children is lower than that of the node. Figure 5.3-a shows a three-scale full quad-tree and Figure 5.3-b shows the corresponding optimal tree obtained from the full tree. For this example, the wavelet filter for the wavelet packet transform is *Coiflet1*. As one can see, the optimal tree has considerably fewer nodes which makes the thresholding task more efficient.

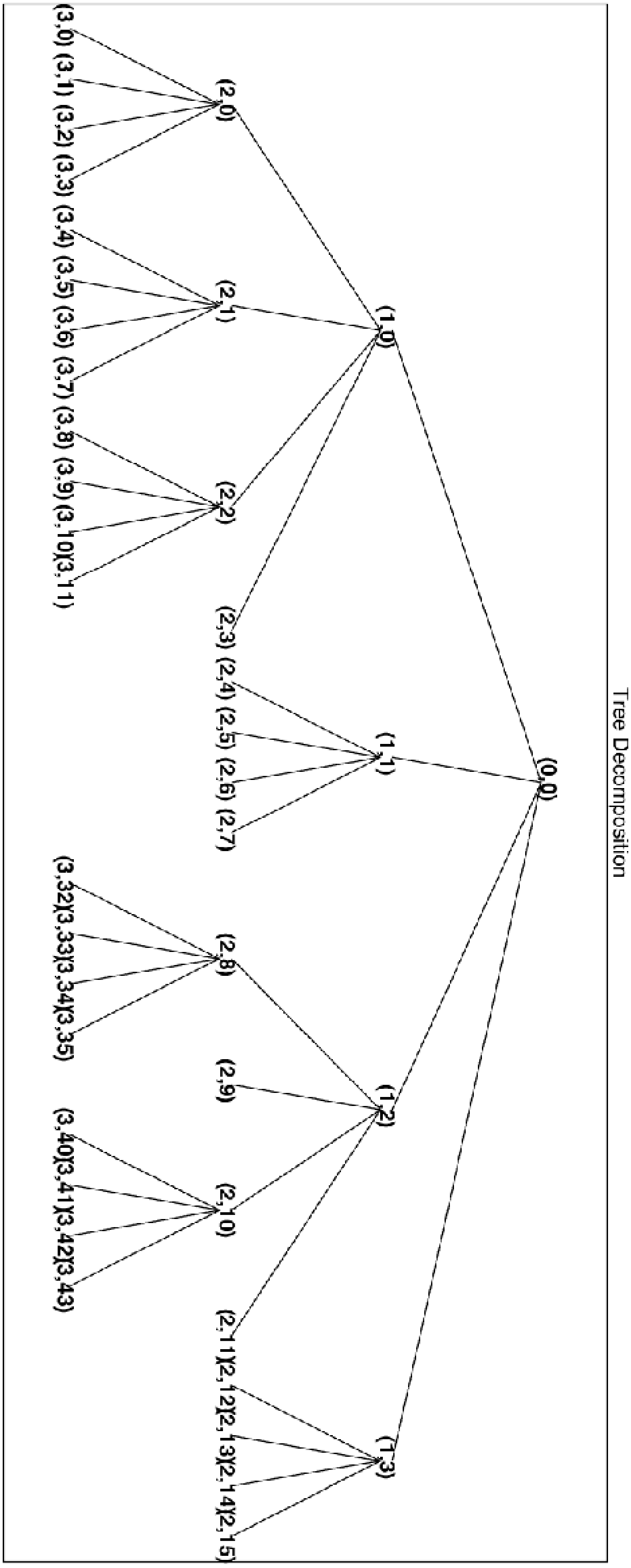
When the optimal tree is obtained, the adaptive shrinkage scheme proposed in WS-FCM is applied to find the proper threshold values based on a fitness metric provided by FCM clustering performance.

PSO Representation

PSO is used here to adaptively look for threshold values based on the fitness function provided by FCM. These threshold values are evolved in the course of the PSO search in a way that they have effective feature enhancement properties. For effectiveness and efficiency purposes, we set a threshold value for each scale of the wavelet packet


Tree Decomposition
(0,0)
(1,0)
(1,1)
(1,2)
(1,3)
(2,0)
(2,1)
(2,2)
(2,3)
(2,4)
(2,5)
(2,6)
(2,7)
(2,8)
(2,9)
(2,10)
(2,11)
(2,12)
(2,13)
(2,14)
(2,15)

(a)



(b)

Figure 5.3: Depiction of the wavelet packet tree of a three-scale decomposed image. (a) The full tree. (b) The optimal tree using Shannon entropy.

decomposition tree. Since our in our approach we have a five-scale decomposition, each particle P_i, is represented as:

$$P_i = [\theta_1, \theta_2, ..., \theta_5] \tag{5.9}$$

in which θ_j is the threshold value for the jth scale of the optimal decomposition tree. A minimum and maximum is set for the value a threshold can take during the PSO search:

$$0 < \theta_j < \theta_{max} \tag{5.10}$$

where θ_{max} is motivated by the Universal threshold, θ_U [47]:

$$\theta_{max} = \sigma\sqrt{2\ln(N)} \tag{5.11}$$

where σ is the standard deviation of noise, and N is the number of pixels in the image. Since we make no assumption about the volume of noise, the σ value will be obtained using the robust median estimator:

$$\sigma = \frac{median\big[|w_{mn} \in \{H_1, D_1, V_1\}|\big]}{0.6745} \tag{5.12}$$

where w_{mn} are noisy coefficients of the detail nodes H_1, V_1, and D_1 in the first scale of the quad-tree.

The initial solutions for particles are selected from the $[0, \theta_{max}]$ interval, and in each iteration the members of particles outside this interval will be replaced by the corresponding member of the $Xpbest$ particles.

Wavelet Packet-based Shrinkage

The coefficients inside the nodes from the optimal quad-tree are thresholded to remove the noise and enhance the intensity feature. A thresholding function is a function under which threshold values are applied for attenuation, manipulation or preservation of coefficients, and play an important role on how accurate the final segmentation results are. We again use the soft-thresholding function proposed in [45] to manipulate wavelet packet coefficients. The soft-thresholding function is continuous everywhere and unlike hard-thresholding [45] does not generate a high variance in the thresholded signal. Equation 5.13 show the mathematical representation for the soft-thresholding function.

$$Y = \begin{cases} sign(w)(|w| - \theta), & \text{if } |w| > \theta \\ 0, & \text{if } |w| \leq \theta \end{cases} \tag{5.13}$$

where w is a wavelet packet coefficient and θ is a threshold value.

Fitness Evaluation

As in the previous chapter, to evaluate the performance of each particle, a fitness function is utilized based on clustering performance of FCM. After all the nodes from the optimal decomposition tree are thresholded with a sample particle, the resulting nodes are used again to build a denoised intensity image. For this, we perform a wavelet packet reconstruction on the thresholded coefficients. The intensity information of the feature-enhanced image is clustered with the predefined number of clusters, and then the objective function of FCM is calculated. The result is taken as the fitness value, and has to be minimized. To recall, the FCM objective function (the fitness function) is:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} d^2(x_i, v_j) \tag{5.14}$$

Differences between WS-FCM and WPS-FCM

There are two differences between WS-FCM and WPS-FCM. First, the number of subbands are fixed in WS-FCM (15), whereas in the WPS-FCM the number of subbands could vary depending on the results of the optimal decomposition tree. Theoretically, this value could range between 16 and 1024. Using a good choice wavelet filter, we can make sure this value is not too high, to prevent the computation from being too expensive. Second, there is a threshold value for each subband in WS-FCM algorithm while there is a threshold value for each scale of decomposition in WPS-FCM. Therefore, there are 15 threshold values in WS-FCM and only five threshold values in WPS-FCM. This difference in WPS-FCM keeps the computation cost low and segmentation accuracy high. For example, having a threshold value for each node (subband) can increase the number of threshold values to the point that the search space becomes too complicated, and PSO is unable to provide effective threshold values.

5.3.2 BWPS-FCM

BWPS-FCM follows the same generic scheme that was proposed for WPS-FCM with two differences. First, it uses a preprocessing step in order to partially denoise the image. As we mentioned in the previous chapter, thresholding in the wavelet domain is well established in image denoising. One feature that plays a significant role in this area is the sparsity of coefficients in the wavelet domain. This way a minority of the coefficients that carry most of the energy of the signal can be preserved while the rest are omitted. One side effect of natural images or images corrupted with high volume of noise is that they are not really sparse in wavelet domain [102], but smoothing results in a sparse representation in wavelet domain. Therefore, smoothing an image could pave the way for a better coefficients thresholding. A pre-denoising step can make the spatial representation of an image smooth, its spatial-frequency sparse, and the wavelet thresholding effective [9]. However, smoothing should not happen on edges, otherwise it would have an unfavorable effect. That is why we utilize the non-iterative detail-preserving bilateral filtering in our algorithm to partially remove the noise from an image, and provide a better sparse space for thresholding of the coefficients.

The second difference is that we use a sigmoid thresholding function for better accuracy. One side effect of soft-thresholding is that it attenuates the large coefficients which results in an over-smoothed image. Also, the zero-forcing effect of the soft-thresholding function causes singularities of the thresholding function. This would create a serious variance of the estimation of the signal due to the sensitivity of the inverse wavelet transform [9]. A proper thresholding function should meet the three following criteria [9]. This sigmoid thresholding which satisfies these criteria has no discontinuity, and has fewer side effects when manipulating coefficients.

- **Smoothness**: enforcing small variability among coefficients of the same amplitude.

- **Penalized shrinkage**: coefficients with small values have less information about the signal, and the ones with higher values contain more information. Therefore, strong attenuation has to apply to small coefficients and weak attenuation to big ones.

- **Vanishing attenuation at infinity**: the attenuation property of the function applied to a signal reduces when the amplitude of the coefficients tends to infinity. This resolve the over-smoothing.

Possessing the above properties, a sigmoid-based thresholding

The block diagram of the proposed BWPS-FCM with more details is shown in Fig. 5.4.

Bilateral Filtering Parameter Setting

One important important step in Bilateral Filtering is its parameter setting according to the noise level. The width of the range Gaussian, σ_r and the width of the spatial Gaussian, σ_s are two parameters that need tuning in a noise-volume-varying situation for effective results. Eq. 5.15 shows how the two parameters can be estimated using the σ value obtained in Eq. 5.12:

$$\sigma_r = \sigma \qquad , \qquad \sigma_s = \frac{c}{1 + e^{\left(-\tau(b\sigma - a)\right)} + d} \tag{5.15}$$

in which the equation estimating σ_s is a sigmoid function whose parameters are experimentally set at $\tau = 0.5$, $a = 12$, $b = 35$, $c = 2$, and $d = 1.1$. The plot of the sigmoid function is shown in Fig. 5.5. This non-linear curve shows that σ_s increases with higher levels of noise estimated by σ. To estimate σ (noise standard deviation) according to Eq. 5.12, a wavelet packet decomposition of the noisy image is required. This setting is aimed at removing the noise partially, but not completely. To prevent Bilateral Filtering from creating redundant borders, we pad the original image with a 20-pixel-wide margin. This margin is set to repeat border pixels of the image. Having σ_r and σ_s estimated, and the image padded, we can now apply the Bilateral Filtering to the noisy image.

5.3.3 PSO Representation

The PSO representation, shrinkage scheme, and fitness evaluation is the same as WPS-FCM. The only difference is the maximum limit of each threshold value. The pre-denoising based on Bilateral Filtering decreases the magnitude of the coefficients. Therefore, we could set a lower maximum limit for particles:

$$0 < \theta_i < \frac{\theta_{max}}{2} \tag{5.16}$$

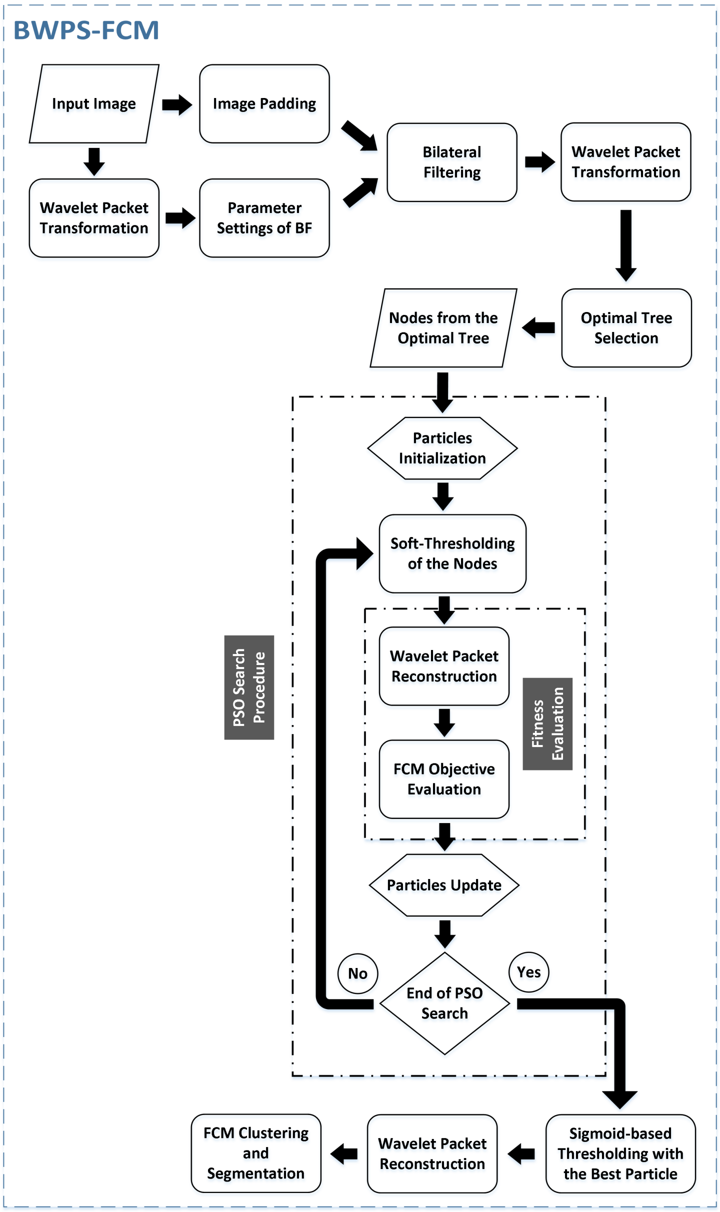

BWPS-FCM
Input Image
Image Padding
Wavelet Packet Transformation
Parameter Settings of BF
Bilateral Filtering
Wavelet Packet Transformation
Optimal Tree Selection
Nodes from the Optimal Tree
Particles Initialization
Soft-Thresholding of the Nodes
Wavelet Packet Reconstruction
FCM Objective Evaluation
Fitness Evaluation
PSO Search Procedure
Particles Update
No
Yes
End of PSO Search
Sigmoid-based Thresholding with the Best Particle
Wavelet Packet Reconstruction
FCM Clustering and Segmentation

Figure 5.4: Block diagram of the proposed BWPS-FCM algorithm.

where θ_{max} is calculated from Eq. 5.11.

Sigmoid Shrinkage

The Smooth Sigmoid-Based Shrinkage (SSBS) function is defined as below [9]:

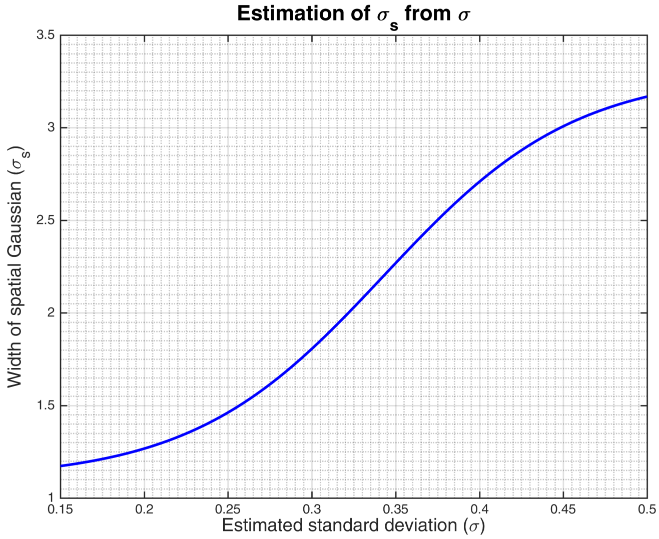


Figure 5.5: The plot of the sigmoid function used to estimate σ_s from σ.

$$\delta_{\tau,\theta} = \frac{w}{1+e^{\left(-\tau(|w|-\theta)\right)}} \tag{5.17}$$

in which w is a coefficient, θ is a threshold value, and τ determines the curvature of the function. A depiction of the sigmoid function for different values of τ is shown in Fig. 5.6. Increasing τ increases the soft-thresholding behavior of the SSBS function, and decreasing τ decreases its manipulating properties on the coefficients.

In BWPS-FCM, τ is determined by the number of large coefficients in each node. A coefficient is considered large if its value is bigger than the standard deviation of noise, σ. After extensive experiments on different noisy images, τ is defined:

$$\tau = \begin{cases} 2, & N_l > \frac{N_n}{20} \\ K, & otherwise \end{cases} \tag{5.18}$$

$$K = \begin{cases} 8, & \sigma \leq 0.2 \\ 10, & 0.2 < \sigma \leq 0.3 \\ 12, & 0.3 < \sigma \leq 0.4 \\ 14, & \sigma > 0.4 \end{cases} \tag{5.19}$$

in which N_l is the number of large coefficients in node n and N_n is the total number of coefficients in the node. Different limits on σ to determine K in Eq. 5.19 are defined for a normalized image. This setting makes sure to have a conservative thresholding,

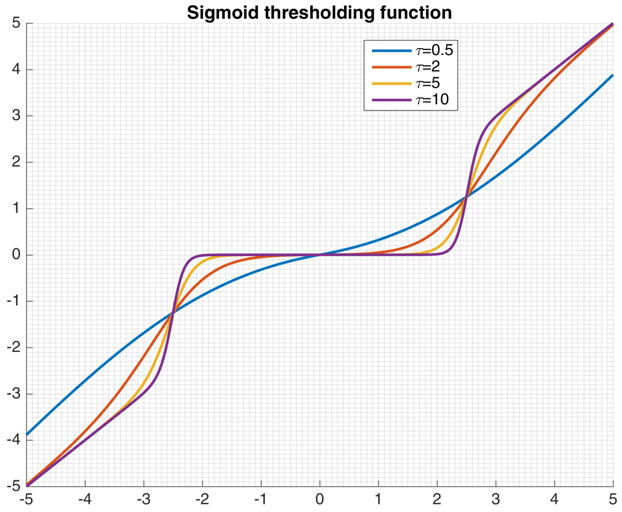


Figure 5.6: Sigmoid shrinkage function applied to a simple signal for different values of τ and a fixed threshold value.

by having a low value for τ, when a node contains many large coefficients. This way, attenuation of the large coefficients is less likely to happen. For the rest of the nodes, τ is reasonably high, and increases with the volume of noise. These nodes contain less useful information about the noisy image and therefore a more strict thresholding on them results in a denoised image with fewer artifacts caused by noise. When the volume of noise increases, bigger values of τ suppress the small coefficients more drastically.

5.4 Experiment Design

5.4.1 Datasets and Evaluation

The datasets, images, and instances are the same as the previous chapter. For qualitative evaluation we have adopted the common Segmentation Accuracy, SA metric [6], which was explained in Chapter 3 (page 45).

We have also taken 10 other state-of-the-are algorithms to compare our method with. Seven of these methods are FCM-modified named FCM_S1 and FCM_S2 [38], EnFCM [118], FGFCM, FGFCM_S1 and FGFCM_S2 [25], and FLICM [74]. The three rest are wavelet thresholding-based algorithms introduced in this or previous chapters named

WS-FCM, WPS-FCM, and AWA-FCM.

5.4.2 Parameter Design

All the parameters related to FCM, PSO, wavelet transform, Bilateral Filtering and their adopted values in this chapter are depicted in Table 5.1 based on empirical search through experiments.

Table 5.1: Parameter settings of the proposed method.

Parameter/Setting	**Value/Type**
Wavelet filter	Coiflets family (Coiflet 5)
Scale number (Wavelet)	5
Termination threshold (FCM)	0.001
Maximum number of iterations (FCM)	100
Weighting exponent, m (FCM)	2
Padding margin size (in pixels) (BF)	20
Population size & iterations in WPS-FCM (PSO)	30 & 100
Population size & iterations in BWPS-FCM (PSO)	20 & 150
c_1 and c_2 (PSO)	1
Inertia weight (ω) (PSO)	Linearly decreasing from 1 to 0.5
σ_r & σ_s (BF)	σ & $\frac{2}{1+e^{\left(-\tau(35\sigma-12)\right)}+1.1}$ (σ is estimated according to Eq. 5.12)

5.4.3 Statistical Significance Test

To analyze the non-deterministic behavior of PSO in our algorithms a pair-wise statistical significance test is performed. Our algorithm is run 30 times independently on each image, and the results in the form of the SA values are compared with the results of other methods using the test. We select the Wilcoxon test with a significance level of 0.05 for this comparison. For more information about this test please refer to [135]. If the p-value (the probability of observing a test statistic as or more extreme than the observed value under the null hypothesis) is greater than the significance level, the pair-wise comparison is not considered significantly different. Otherwise, one method is significantly better than the other. Tables that are providing *p* values also use **+**, **-** and **=** signs to show that one method is significantly better, worse, or not different respectively for an instance.

5.5 Results and Discussion

This section analyzes all the experiments in comparison to other methods both quantitatively and qualitatively. To have a proper comparison in the quantitative part, both SA and *p* values are considered for an instance. The comparison is made in separate sub-sections for different datasets. The first part in each sub-section, presents a comprehensive quantitative evaluation and the second part presents some qualitative evaluations. In the quantitative evaluation part, we look at the comparisons from four different aspects. We first compare the results of the adaptive wavelet-based shrinkage, WS-FCM, and its equivalent in wavelet packet domain, WPS-FCM. Second, we compare the results of the proposed BWPS-FCM with the comparison methods on each instance in a pair-wise manner. Throughout the pair-wise comparison, a better performance is determined according to SA value unless the *p* value from the Wilconxon test for that pair is bigger than the significance level (0.05). Third, another evaluation is done by ranking the performance of all the algorithms on each instance. Based on the number of the best and second-best performances for each algorithm, the overall best, second-best, and third best performers are determined. Fourth, the quantitative analysis in each sub-section concludes with a performance evaluation over the noise variation on each sample image. This evaluation investigates how SA metric changes in terms of mean and variance when the noise variance ranges between 10% to 80%.

5.5.1 SynthDB (Quantitative Analysis)

WPS-FCM vs WS-FCM

As mentioned before, it is an interesting comparison to know how an adaptive thresholding of wavelet and wavelet packet coefficients in similar representations differ in feature enhancement.

Tables 5.2 provides the SA values for the two algorithms, and 5.3 provides the *p* values from the Wilcoxon test. There are a total of 10 instances that the two methods perform not significantly different from each other. These value are bold in Table 5.3. For the remaining 30 instances, WS-FCM performs better than WPS-FCM in all the instances with mostly very small differences as show in Table 5.2. This indicates that from the quantitative point of view, WPS-FCM cannot add much to the segmentation accuracy of SynthDB Gaussian instances.

BWPS-FCM vs Other Methods in SynthDB

The Wilcoxon test results from the comparison between BWPS-FCM and other methods is provided in Table 5.4 in which there are an overall 15 instances that *p* is bigger than the significance level. Seven instances belong to the comparison to WS-FCM, four instances belong to the comparison to WPS-FCM, and four others belong to the comparison to AWA-FCM. Now, we can compare BWPS-FCM with other methods pair-wisely according to the SA values provided in Table 5.2 to determine the better performance where the difference is significant.

BWPS-FCM performs better on all 40 instances when compared to FCM_S1 and FCM_S2, EnFCM, FGFCM, FGFCM_S1 and FGFCM_S2. The number of better performances in comparison to FLICM is 39, to WS-FCM is 22, to WPS-FCM is 29, and to AWA-FCM is only three. Thus, BWPS-FCM performs mostly better than FLICM, WS-FCM, and WPS-FCM. However, in comparison to AWA-FCM, there are four instances of not significantly different results, three instances of better BWPS-FCM results, and 33 instances of better AWA-FCM results.

For the overall best performer on each instance, as expected, AWA-FCM is the best performer among all being the best performer on 31 instances, and second-best performer on four further instances. BWPS-FCM is the overall second-best performer being the best performer on three instances and second-best performer on 18 more instances. FLICM is the the overall third-best performer, having one instance of best performance and none second-best performance.

The results of analysis for segmentation accuracy over noise variation is provided in Fig. 5.7. Fig. 5.7(a) shows the mean of the SA values resulted from all the 11 methods for eight different noise levels for each image. AWA-FCM introduced in the previous chapter, has the highest mean SA in all five images. The second-best performer from this perspective is BWPS-FCM having the second-highest mean SA in four images. WS-FCM is the next good performer being the second-highest in one and the third-highest in four images. The results of the same analysis for the variance of SA is shown in the bar graph presented in Fig. 5.7(b). BWPS-FCM have a low variance over the noise level variation except for image $Synth4$ which has a few unusual standard deviations for a few instances in 30 independent runs. These bigger standard deviations exist due to the sensitivity of wavelet packet reconstruction to zero forcing and not due to non-deterministic characteristics of PSO. We mentioned before that zero-forcing of thresholding functions such as hard/soft-thresholding could cause this sensitivity of the reverse wavelet transform. For high values of τ the sigmoid function turns into a hard-thresholding function which causes the sensitivity to wavelet packet reconstruction.

Table 5.2: SA values for the SynthDB instances. The bold number indicates the best performance for each instance where the difference is significant.

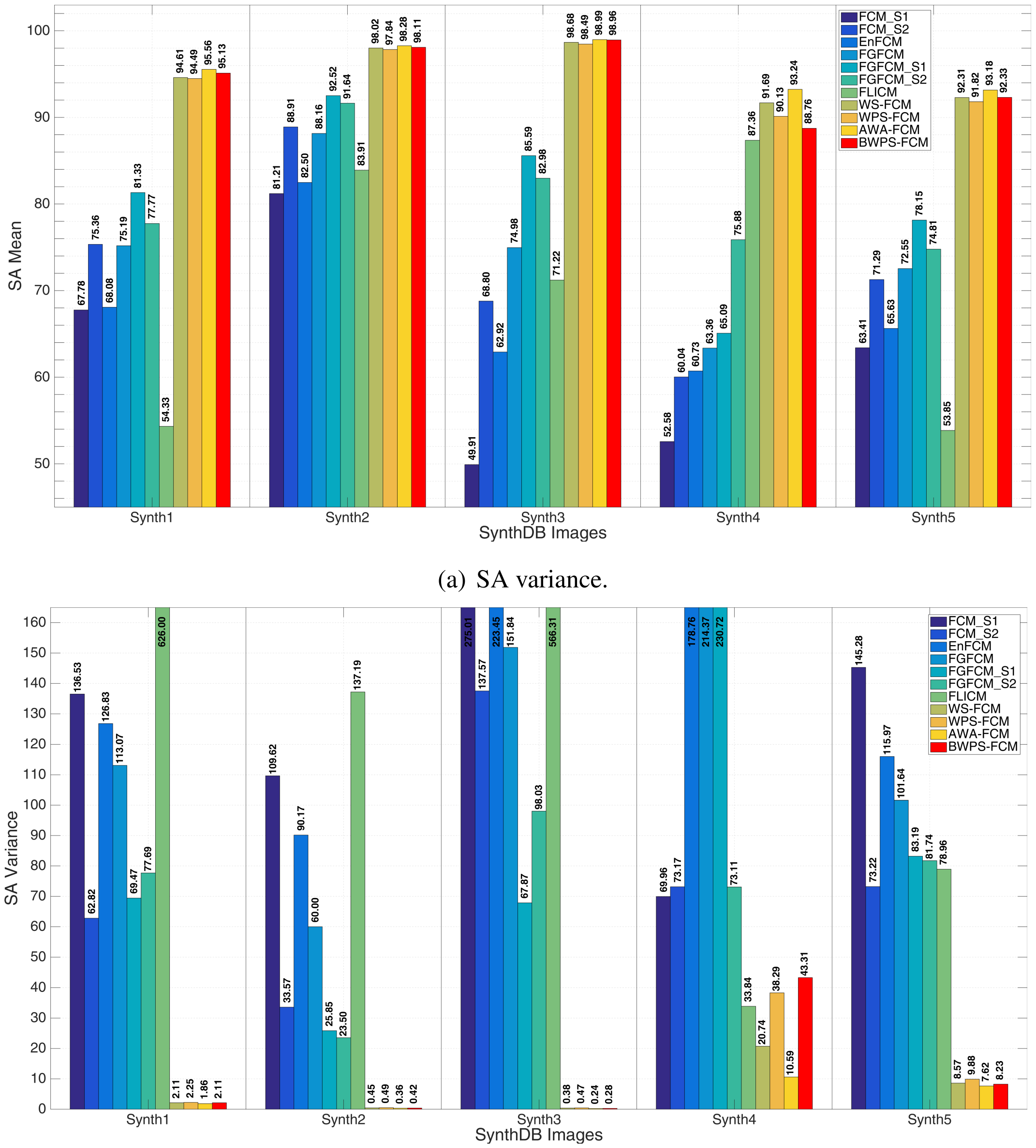
		Algorithm										
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**WPS-FCM**	**AWA-FCM**	**BWPS-FCM**
Synth1	10%	88.98	89.86	88.98	93.12	94.33	93.01	95.90	97.06± 0.03	97.07± 0.34	97.74±0.03	97.75±0.05
	20%	79.41	82.85	78.95	86.22	89.66	87.22	88.04	95.94± 0.05	95.77± 0.47	**96.80±0.02**	95.77±0.28
	30%	71.62	78.14	71.44	79.86	85.99	81.37	58.15	95.46± 0.09	95.37± 0.39	**96.14±0.05**	95.68±0.12
	40%	67.16	74.94	67.10	75.31	82.15	77.10	44.95	94.28± 0.10	93.95± 0.32	**95.53±0.05**	95.17±0.07
	50%	62.77	71.79	63.34	71.38	78.93	72.83	43.12	94.07± 0.07	94.07± 0.64	**95.07±0.01**	94.91±0.06
	60%	59.78	70.42	60.64	67.94	76.36	72.67	29.27	94.17± 0.10	94.06± 0.26	**95.56±0.03**	95.00±0.04
	70%	57.25	68.61	58.12	64.82	73.33	70.26	44.69	93.19± 0.14	93.26± 0.36	94.11±0.07	**94.16±0.10**
	80%	55.29	66.27	56.10	62.87	69.89	67.67	30.49	92.70± 0.04	92.37± 0.45	**93.56±0.03**	92.64±0.24
Synth2	10%	97.85	98.41	97.89	99.11	99.02	98.73	99.18	99.12± 0.01	99.03± 0.04	99.29±0.00	99.28±0.03
	20%	92.26	95.05	92.42	96.51	97.85	96.98	97.35	98.67± 0.03	98.40± 0.19	**98.85±0.00**	98.66±0.07
	30%	87.25	91.49	87.72	93.58	96.51	94.87	96.28	98.36± 0.03	98.17± 0.13	**98.59±0.01**	98.36±0.11
	40%	81.72	89.21	82.85	89.11	93.49	92.02	74.52	98.13± 0.01	98.13± 0.12	**98.30±0.01**	98.01±0.07
	50%	76.60	86.52	78.22	85.39	91.32	89.71	82.32	97.64± 0.05	97.40± 0.49	**97.97±0.02**	97.81±0.03
	60%	74.44	85.53	76.40	83.65	90.05	88.70	74.51	97.38± 0.01	97.26± 0.25	**97.80±0.01**	97.64±0.04
	70%	71.04	83.49	73.47	80.16	86.40	86.54	73.24	97.68± 0.03	97.43± 0.60	**98.02±0.01**	97.87±0.04
	80%	68.52	81.58	71.02	77.76	85.56	85.59	73.89	97.18± 0.02	96.93± 0.22	**97.44±0.01**	97.21±0.05
Synth3	10%	81.10	93.30	96.04	97.68	97.25	97.15	99.65	99.61± 0.00	99.52± 0.08	**99.72±0.00**	99.71±0.02
	20%	67.89	76.50	69.33	88.37	95.48	94.37	99.19	99.33± 0.01	99.09± 0.21	99.46±0.01	99.46±0.02
	30%	52.89	71.85	64.92	76.71	90.76	88.78	98.63	98.92± 0.04	98.81± 0.13	**99.16±0.01**	99.15±0.01
	40%	45.92	67.07	60.81	72.89	86.83	84.82	65.52	98.73± 0.02	98.65± 0.14	99.10±0.00	**99.16±0.01**
	50%	41.31	62.38	57.00	69.78	80.65	78.62	55.60	98.54± 0.06	98.20± 0.26	**98.96±0.02**	98.94±0.01
	60%	38.72	61.55	54.03	68.78	80.23	77.18	54.36	98.56± 0.02	98.41± 0.18	**98.83±0.01**	98.81±0.02
	70%	36.99	59.21	51.58	63.26	78.07	72.38	51.58	97.79± 0.03	97.37± 0.64	98.23±0.02	**98.27±0.02**
	80%	34.49	58.54	49.69	62.40	75.42	70.59	45.22	97.96± 0.08	97.87± 0.21	**98.46±0.01**	98.21±0.02
Synth4	10%	72.59	77.48	85.28	93.80	92.25	93.70	**98.16**	96.49± 0.01	95.96± 0.60	97.11±0.01	97.08±0.03
	20%	54.42	67.71	72.10	72.65	84.55	82.02	92.41	95.16± 0.08	94.34± 0.68	95.94±0.05	95.10±3.06
	30%	51.00	60.31	66.59	65.40	65.40	75.39	82.16	93.93± 0.04	93.12± 0.94	**94.93±0.04**	79.15±3.22
	40%	49.36	58.30	61.76	62.01	60.43	76.61	81.68	93.21± 0.03	92.50± 1.16	**94.26±0.02**	87.56±8.04
	50%	48.69	55.94	53.88	57.90	57.28	72.07	81.86	91.89± 0.04	90.33± 1.37	**93.23±0.02**	89.01±5.15
	60%	48.82	54.17	50.62	56.55	56.25	70.21	88.39	90.13± 0.32	89.19± 1.38	**92.02±0.12**	91.65±0.27
	70%	48.04	54.01	48.08	51.60	54.57	68.73	88.55	90.99± 0.15	89.66± 1.04	**91.88±0.05**	91.03±1.74
	80%	47.76	52.40	47.51	47.01	50.01	68.26	85.70	81.70±10.08	75.97±10.52	86.56±4.63	79.46±1.58
Synth5	10%	85.62	86.90	85.54	90.31	93.15	90.68	57.80	96.84± 0.09	96.48± 0.44	**97.38±0.07**	96.91±0.08
	20%	75.55	79.95	76.33	82.99	88.01	84.38	58.73	95.06± 0.07	94.83± 0.40	**95.76±0.01**	94.68±0.32
	30%	67.21	74.05	69.03	76.47	82.39	78.03	58.25	93.96± 0.11	93.81± 0.40	**95.19±0.03**	94.49±0.09
	40%	62.39	70.96	64.81	72.43	78.87	74.89	59.96	93.09± 0.06	92.64± 0.71	**93.43±0.04**	92.62±0.14
	50%	57.20	66.68	60.11	67.50	73.84	70.19	58.19	90.54± 0.13	89.40± 0.64	**91.85±0.09**	91.08±0.14
	60%	54.73	65.21	57.89	65.19	71.62	68.06	39.66	90.98± 0.15	90.39± 0.37	**91.80±0.07**	90.88±0.11
	70%	52.90	63.78	56.20	63.36	69.29	66.56	58.90	89.99± 0.06	89.90± 0.69	**90.85±0.02**	89.58±0.24
	80%	51.73	62.83	55.17	62.18	68.06	65.67	39.33	88.04± 0.12	87.11± 0.28	**89.20±0.05**	88.44±0.26

Table 5.3: Wilconxon provided *p* values resulted from the pair-wise comparison of WPS-FCM with WS-FCM for the SynthDB instances. The bold numbers indicate *p* values greater than the significance level. **+**, **-** and **=** respectively show where WPS-FCM performs significantly better, significantly worse or not significantly different than WS-FCM.

	Noise Level							
Img.	10%	20%	30%	40%	50%	60%	70%	80%
Synth1	**0.18** (=)	**0.2** (=)	**0.85** (=)	4.3e-06 (-)	**0.49** (=)	**0.11** (=)	**0.079** (=)	1.6e-05 (-)
Synth2	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	**0.69** (=)	0.0035 (-)	0.02 (-)	0.0082 (-)	1.7e-06 (-)
Synth3	1.7e-06 (-)	1.7e-06 (-)	0.0002 (-)	0.0023 (-)	2.1e-06 (-)	1.3e-05 (-)	5.8e-05 (-)	0.047 (-)
Synth4	0.00014 (-)	9.3e-06 (-)	4.1e-05 (-)	0.0028 (-)	3.9e-06 (-)	0.00072 (-)	1.7e-06 (-)	0.0053 (-)
Synth5	8.2e-05 (-)	**0.06** (=)	**0.16** (=)	0.005 (-)	1.7e-06 (-)	1.7e-06 (-)	**0.56** (=)	1.7e-06 (-)

Table 5.4: Wilconxon provided p values resulted from the pair-wise comparison of BWPS-FCM with other methods for the SynthDB instances. The bold numbers indicate p values greater than the significance level.

		Algorithm									
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**WPS-FCM**	**AWA-FCM**
Synth1	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (=)	**0.16** (=)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.0023 (-)	**0.77** (+)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.6e-06 (+)	0.00024 (+)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.011 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.38** (=)	0.02 (+)	1.7e-06 (-)
Synth2	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (=)	**0.085** (=)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.71** (=)	1.7e-06 (+)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.86** (=)	2e-05 (+)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	6.9e-06 (-)	0.00032 (-)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.14** (=)	1.7e-06 (+)	1.7e-06 (-)
Synth3	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.0023 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (=)	**0.21** (=)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00061 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.0015 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00013 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.1e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (-)
Synth4	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	3.9e-05 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.4e-06 (+)	1.7e-06 (+)	0.00036 (+)	0.0026 (-)	0.00042 (=)	**0.75** (=)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.4e-06 (+)	8.2e-05 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.2e-05 (+)	0.0014 (+)	**0.082** (=)	**0.094** (+)	0.0077 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.5e-05 (+)	**0.21** (=)	**0.89** (+)	1.7e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	5.2e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	3.1e-05 (+)	3.1e-05 (+)	1.9e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.98** (=)	0.023 (+)	3.1e-05 (-)
Synth5	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.0034 (+)	3.2e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.7e-05 (-)	0.033 (-)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	**0.39** (+)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.0087 (-)	1.7e-06 (+)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	7e-06 (-)	0.017 (-)	1.7e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	9.3e-06 (+)	1.7e-06 (+)	1.7e-06 (-)



(a) SA variance.

(b) SA mean.

Figure 5.7: Demonstration of different algorithms performance on noise level variation (SynthDB).

SynthDB (Qualitative Analysis)

Some segmentation results are shown in Fig. 5.8 for all the 11 algorithms. A sample instance for each image is provided in this visual comparison. These instances for *Synth*1, *Synth*2, *Synth*3, *Synth*4, and *Synth*5 are the instances with noise variance of $10\%, 60\%, 70\%, 80\%$, and 30% respectively. Algorithms based on modification of FCM objective function (except for FLICM) suffer from over-segmentation. Segmentation of compact/homogeneous regions is hardly possible from the outcomes of these algorithms. This deficiency is more serious in FCM_S1, FCM_S2, and EnFCM, and less serious in FGFCM, FGFCM_S1, and FGFCM_S2. FLICM can perform better form this perspective for example on the $Synth1$ instance, or suffer from over-segmentation for example in the $Synth3$ instances, and sometimes miss one or two distinguishable regions existing in the groundtruth for the instances of $Synth2$, $Synth4$, and $Synth5$.

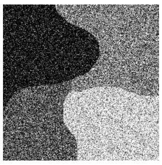
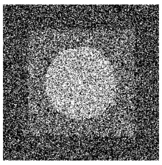
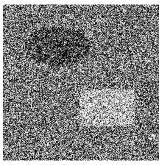
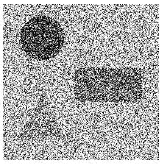
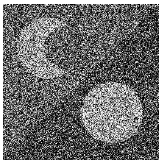
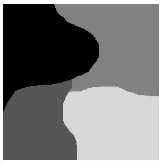
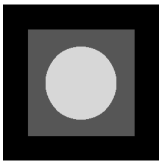
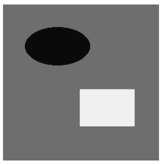
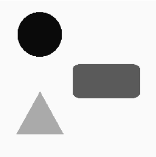
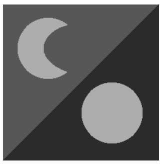
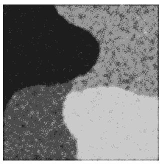
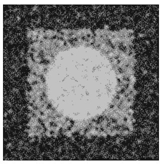
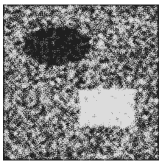
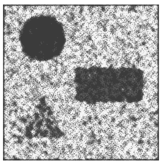
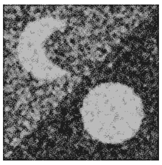
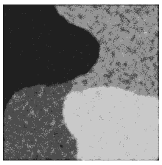
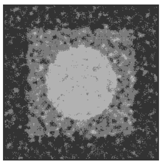
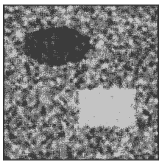
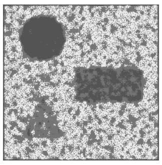
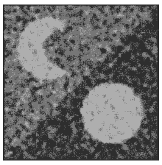
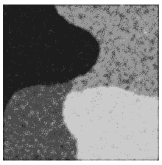
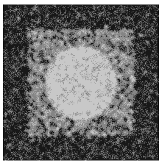
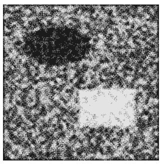
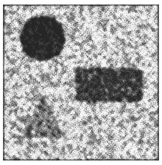
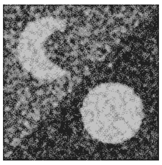
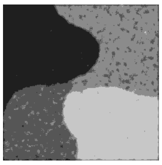
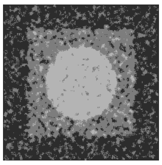
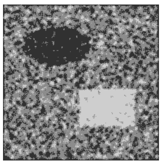
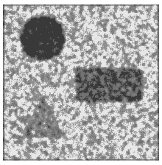
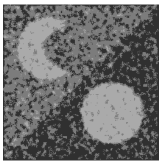
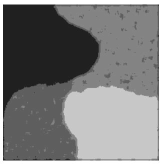
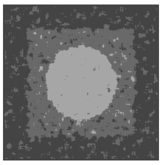
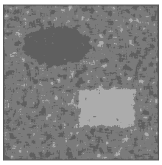
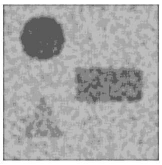
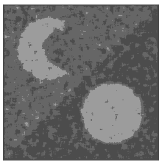
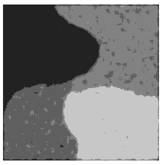
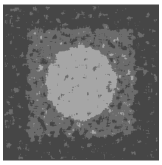
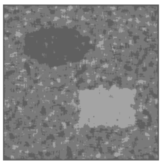
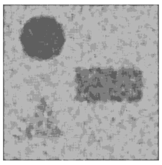
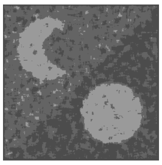
There is no significant visual difference in the segmentation results of WS-FCM and WPS-FCM. Generally, it seem that boundaries are less over-smoothed in WPS-FCM segmentation results, and more details are related to them. This is observable from the $Synth1$ and $Synth3$ instances, although this is not true for the $Synth4$ instance. AWA-FCM and BWPS-FCM mostly produce the most acceptable visual segmentations by having not so much redundancy around the boundaries. AWA-FCM performs even better by not producing spurious regions caused by noise.

5.5.2 BerkDB (Quantitative Analysis)

Similar to the SynthDB dataset results, we first present a comparison of WS-FCM and WPS-FCM, and then a comprehensive comparison of the proposed BWPS-FCM with all other methods.

WPS-FCM vs WS-FCM in BerkDB

Table 5.6 shows the SA values for all the methods tested on BerkDB instances, and Table 5.5 shows the p values from the Wilcoxon test for the comparison of WS-FCM and WPS-FCM. There are an overall 25 instances in Table 5.5 that the p value is bigger than the significance level (0.05). These are the instances that the methods perform not significantly different. Having 25 out of 88 not significantly different results is an indication of the fact that WPS-FCM and WS-FCM are performing almost the same in term of segmentation accuracy in BerkDB. In fact, this is confirmed to some extent when the number of significantly better results are obtained. WPS-FCM performs better in 30 instances, and WS-FCM does so in 33 instances. Although WS-FCM is overall performing slightly better than WPS-FCM, having a look at table 5.6 shows that the per-

	S1 ($\sigma^2 = 10\%$)	S3 ($\sigma^2 = 60\%$)	S3 ($\sigma^2 = 70\%$)	S4 ($\sigma^2 = 80\%$)	S5 ($\sigma^2 = 30\%$)
Noisy Images					
Groundtruths					
FCMS_1					
FCMS_2					
EnFCM					
FGFCM					
FGFCMS_1					
FGFCMS_2					

Continued on next page

Figure 5.8 – continued from previous page

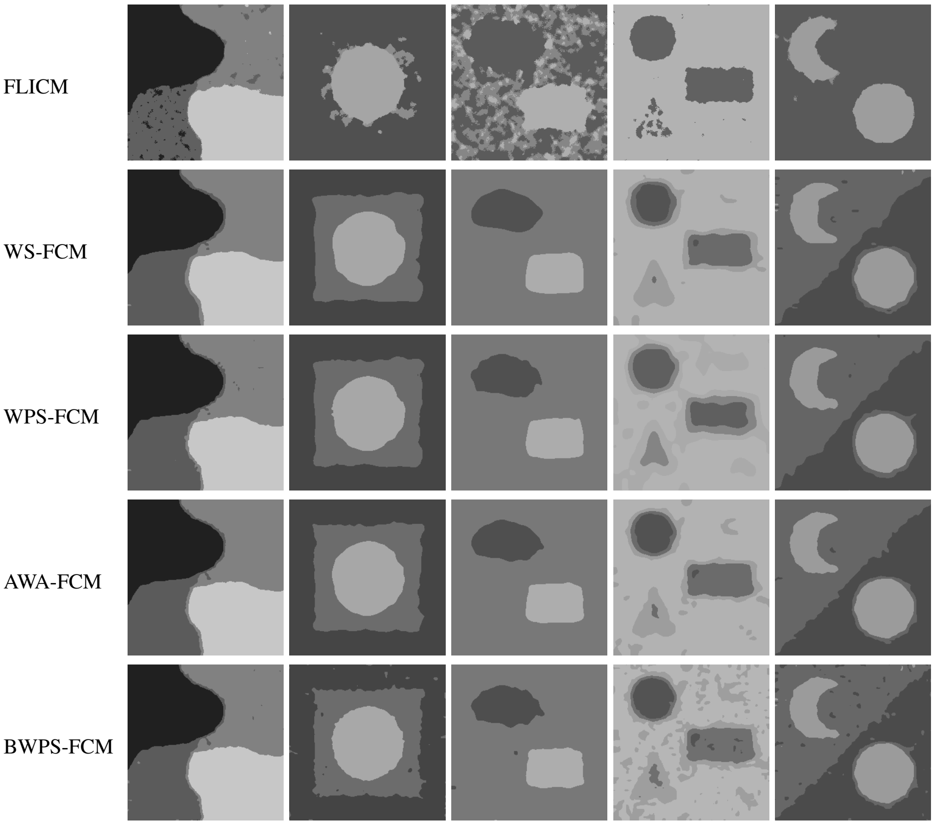


Figure 5.8: Qualitative comparison of the proposed BWPS-FCM with FCM_S1, FCM_S2, EnFCM, FGFCM_S1, FGFCM_S2, FGFCM, FLICM, WS-FCM, WPS-FCM, and AWA-FCM on some instances from SynthDB. *Synth*1, *Synth*2, *Synth*3, *Synth*4, and *Synth*5 are corrupted by Gaussian noise of variance 10%, 60%, 70%, 80%, and 30% respectively.

formance difference is mostly very low. Having said that, there are a total of six images (*B*3096, *B*8068, *B*253036, *B*147091, *B*108073, and *B*135069) for which the number of significantly better performances over the eight instances is bigger for WPS-FCM in comparison to WS-FCM. In one image the number of significantly better performances are equal (*B*24063), and in four other images WS-FCM is possessing the majority of significantly better performances (*B*42049, *B*167062, *B*86016, and *B*196027). Therefore, image-wise, WPS-FCM performs better than WS-FCM.

Table 5.5: Wilconxon provided p values resulted from the pair-wise comparison of WPS-FCM with WS-FCM for the BerkDB dataset corrupted with Gaussian noise. The bold numbers indicate p values greater than the significance level. **+**, **-** and **=** respectively show where WPS-FCM performs significantly better, significantly worse or not significantly different than WS-FCM.

	Noise Level							
Img.	10%	20%	30%	40%	50%	60%	70%	80%
B3096	2e-05 (+)	**0.25** (=)	**0.32** (=)	0.0039 (+)	**0.55** (=)	0.0039 (+)	0.028 (-)	**0.24** (=)
B8068	**0.21** (=)	**0.16** (=)	2.6e-05 (-)	0.00055 (+)	0.00023 (-)	0.006 (+)	1.7e-06 (+)	0.0022 (+)
B42049	0.00026 (-)	0.0012 (-)	7.7e-06 (-)	**0.072** (=)	**0.14** (=)	**0.53** (=)	2.4e-06 (-)	**0.11** (=)
B167062	1.7e-06 (-)	2.6e-06 (-)	0.0045 (-)	1.7e-06 (-)	0.038 (-)	1.7e-06 (-)	1.7e-06 (-)	2.6e-06 (-)
B86016	0.019 (+)	0.012 (-)	1.7e-06 (-)	3.1e-05 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	5.3e-05 (+)
B196027	**0.082** (=)	4.7e-06 (-)	0.0011 (+)	0.021 (-)	0.00089 (-)	0.00053 (-)	**0.13** (=)	0.019 (-)
B24063	0.00039 (-)	**0.78** (=)	0.0017 (+)	8.2e-05 (-)	2.6e-06 (-)	2e-05 (+)	0.00039 (-)	0.007 (+)
B253036	**1** (=)	0.00049 (+)	0.0012 (+)	0.047 (-)	1.7e-06 (+)	**0.77** (=)	4.9e-05 (+)	0.0066 (-)
B147091	0.0008 (+)	1.7e-06 (+)	2.4e-06 (+)	0.0047 (+)	0.00028 (-)	1.7e-06 (-)	**0.63** (=)	1.9e-06 (+)
B108073	**0.34** (=)	**0.95** (=)	**0.48** (=)	0.012 (+)	**0.13** (=)	**0.45** (=)	**0.054** (=)	0.043 (+)
B135069	1.2e-05 (+)	**0.15** (=)	0.00015 (+)	1.7e-06 (+)	0.014 (+)	**0.15** (=)	**0.18** (=)	0.012 (+)

Table 5.6: SA values for the BerkDB dataset corrupted with Gaussian noise. The bold number indicates the best performance for each instance. The bold numbers indicate the best performance for each instance where the difference is significant.

		Algorithm										
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**WPS-FCM**	**AWA-FCM**	**BWPS-FCM**
B3096	10%	63.55	66.46	68.06	74.04	80.67	83.04	6.13	80.74±0.03	81.19±0.54	83.26±0.03	**85.69±0.17**
	20%	58.93	62.44	61.03	66.89	73.65	74.45	6.13	77.97±0.02	78.04±0.38	80.15±0.01	**82.31±0.13**
	30%	57.04	60.84	57.65	63.23	70.29	70.70	6.13	79.40±0.04	79.45±0.32	81.97±0.04	**82.83±0.08**
	40%	55.57	59.27	54.87	59.98	66.78	65.59	6.13	80.59±0.07	80.78±0.30	81.50±0.03	**83.44±0.13**
	50%	55.06	58.49	54.91	58.47	64.92	62.43	6.13	78.45±0.01	78.59±0.85	79.59±0.01	**81.12±0.41**
	60%	54.65	58.38	53.80	58.65	63.42	63.23	6.13	77.48±0.05	77.70±0.52	78.64±0.02	**80.41±0.15**
	70%	54.08	57.55	53.94	57.66	61.80	61.18	6.13	73.98±0.06	73.93±0.71	74.13±2.08	**74.28±0.18**
	80%	53.71	57.39	53.10	56.40	62.73	61.20	6.13	75.57±0.03	75.10±2.04	77.22±0.03	77.15±0.22
B8068	10%	94.56	94.80	94.73	94.94	95.22	95.12	**95.78**	95.05±1.17	95.29±0.07	95.38±0.00	95.24±0.02
	20%	93.60	94.06	93.90	94.40	94.72	94.61	**96.45**	95.16±0.01	95.17±0.05	95.08±0.00	94.94±0.03
	30%	92.25	93.30	92.98	93.84	94.35	93.97	**96.92**	94.93±0.01	94.86±0.07	94.90±0.00	94.92±0.02
	40%	90.20	92.74	91.81	93.51	94.33	93.81	**96.96**	94.98±1.19	95.16±0.05	95.29±0.01	95.19±0.03
	50%	87.97	92.36	90.07	93.23	94.18	93.57	**96.93**	95.40±0.00	95.36±0.04	95.37±0.00	95.32±0.02
	60%	85.27	91.17	87.59	92.04	93.72	92.81	77.69	95.07±0.01	95.08±0.03	95.27±0.00	**95.28±0.02**
	70%	82.27	89.89	84.47	90.66	93.03	92.02	77.69	95.12±0.02	95.26±0.04	**95.28±0.00**	95.06±0.03
	80%	80.46	89.37	82.31	89.81	92.73	91.54	77.69	94.87±0.01	94.91±0.07	**94.94±0.00**	94.88±0.01
B42049	10%	92.98	93.46	93.34	94.04	93.38	93.79	95.15	94.64±0.01	94.50±0.17	94.89±0.01	**95.20±0.05**
	20%	88.83	90.87	90.70	92.70	92.79	92.65	94.02	93.40±0.01	93.20±0.31	94.07±0.00	**94.78±0.03**
	30%	84.09	89.13	86.73	91.81	92.50	92.08	94.53	93.24±0.00	92.81±0.34	94.24±0.00	94.54±0.18
	40%	78.97	86.15	81.55	88.72	90.90	89.92	19.09	92.36±0.01	92.26±0.43	93.38±0.00	**93.64±0.21**

Continued on next page

Table 5.6 – continued from previous page

		Algorithm										
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**WPS-FCM**	**AWA-FCM**	**BWPS-FCM**
	50%	75.56	84.18	77.37	86.63	90.31	89.27	19.09	92.54±0.11	92.35±0.45	93.51±0.04	**93.67±0.16**
	60%	72.97	81.88	74.13	83.44	88.77	86.74	19.09	90.64±0.09	90.77±0.62	91.96±0.05	**92.84±0.12**
	70%	71.45	81.31	72.68	82.41	87.78	86.31	19.09	91.52±0.05	90.60±0.68	92.29±0.03	**92.93±0.14**
	80%	69.66	79.04	70.59	78.54	85.29	83.61	19.09	89.30±0.05	89.66±0.95	91.10±0.04	**92.22±0.06**
B167062	10%	79.13	79.68	85.42	97.92	97.16	81.66	**99.08**	98.54±0.00	98.34±0.14	98.68±0.00	98.56±0.02
	20%	77.43	79.21	82.03	79.52	97.04	81.68	**99.19**	98.23±0.01	97.91±0.15	98.38±0.00	93.87±5.66
	30%	76.92	78.87	81.69	78.32	76.14	80.34	**99.24**	97.78±0.01	97.70±0.12	98.01±0.01	96.02±4.98
	40%	76.11	77.77	79.81	77.17	74.39	80.02	**99.23**	97.81±0.00	97.41±0.23	97.97±0.00	94.72±5.57
	50%	75.60	77.21	78.63	76.06	73.42	80.47	**99.27**	97.60±0.01	97.57±0.24	97.78±0.00	97.67±0.04
	60%	74.64	77.10	76.99	75.75	88.21	81.03	**98.80**	97.51±0.01	96.56±2.04	97.41±0.01	97.32±0.02
	70%	73.67	76.45	75.14	74.50	71.25	80.64	**98.93**	97.53±0.00	97.24±0.35	97.76±0.00	97.66±0.04
	80%	72.58	75.61	73.56	74.51	71.11	79.92	**98.93**	97.16±0.01	96.57±0.35	96.09±4.04	95.53±4.74
B86016	10%	85.75	86.62	88.58	92.21	93.73	92.71	**99.09**	98.46±0.02	98.49±0.08	98.37±0.01	97.65±0.07
	20%	76.81	80.21	79.30	86.78	90.70	88.64	16.36	**97.65±0.01**	97.60±0.09	96.85±2.68	97.08±0.08
	30%	71.68	76.25	72.80	80.45	86.48	83.51	16.36	**97.76±0.02**	97.54±0.13	97.38±0.01	97.32±0.04
	40%	69.44	75.43	70.95	78.20	84.64	82.00	16.36	96.94±2.86	97.10±0.11	**97.11±0.04**	97.00±0.06
	50%	67.24	73.56	67.63	75.02	83.35	79.75	16.36	**98.44±0.01**	98.21±0.19	98.37±0.01	98.01±0.10
	60%	65.75	72.43	66.10	73.31	80.68	78.14	16.36	**97.84±0.00**	97.64±0.14	97.63±0.00	97.15±0.10
	70%	63.92	69.97	64.21	69.92	77.25	73.41	16.36	97.13±0.01	96.81±0.23	96.55±3.14	96.70±0.14
	80%	63.32	70.21	63.88	69.91	76.07	74.34	16.36	94.93±3.11	95.27±0.16	**95.73±0.03**	95.64±0.04
B196027	10%	73.62	75.25	76.34	78.23	79.70	80.38	**90.42**	79.55±0.02	79.49±0.19	80.42±0.01	80.52±0.04
	20%	67.89	70.80	70.11	74.12	76.61	77.32	11.57	79.79±0.05	79.31±0.41	80.26±0.02	80.24±0.07
	30%	65.12	68.67	66.27	71.24	75.21	74.64	11.57	78.78±1.78	78.85±0.25	79.87±0.02	**80.40±0.01**
	40%	62.99	67.39	63.80	69.02	73.96	72.83	11.57	76.58±0.20	76.42±0.49	77.91±0.05	**78.25±0.07**
	50%	61.51	65.88	62.54	66.95	71.91	70.44	11.57	80.18±0.02	80.00±0.34	80.36±0.06	79.93±2.17
	60%	60.52	65.28	61.13	65.74	71.09	69.83	11.57	79.87±0.02	79.66±0.38	80.28±0.02	**80.80±0.03**
	70%	59.39	64.34	59.91	64.49	70.24	67.66	11.57	76.38±1.48	76.68±0.44	**77.78±0.03**	77.65±0.04
	80%	59.05	63.80	59.04	63.40	68.64	66.68	11.57	**80.29±0.14**	79.63±2.23	80.14±0.12	80.14±0.08
B24063	10%	91.41	92.76	91.52	94.15	94.77	94.91	80.52	95.93±0.01	95.72±0.21	96.60±0.00	**96.92±0.03**
	20%	85.99	88.53	85.99	89.82	92.24	91.51	56.98	95.07±0.12	95.09±0.25	95.70±0.00	**95.91±0.07**
	30%	81.95	85.50	81.77	86.70	90.36	88.46	53.42	95.74±0.00	95.78±0.13	96.36±0.00	**96.38±0.02**
	40%	79.27	83.65	78.99	84.02	87.79	86.19	58.70	94.89±0.17	94.61±0.30	95.49±0.05	**95.59±0.02**
	50%	76.42	81.39	76.11	81.86	86.08	83.39	58.70	94.06±0.06	93.88±0.13	94.68±0.01	94.67±0.06
	60%	74.74	80.45	74.39	80.23	84.18	82.20	58.70	93.75±0.06	94.12±0.28	94.52±0.01	**94.71±0.04**
	70%	74.40	80.34	74.09	79.77	84.20	81.61	58.70	94.51±0.01	94.39±0.14	94.79±0.01	**94.85±0.01**
	80%	71.89	78.65	71.73	77.45	81.54	79.93	58.70	93.34±0.06	93.44±0.16	**94.15±0.06**	94.07±0.08
B253036	10%	93.15	92.63	93.34	94.88	95.07	93.79	**98.46**	96.56±0.03	96.57±0.15	96.88±0.02	97.04±0.02
	20%	88.51	89.38	89.08	91.98	93.61	91.67	**97.60**	95.91±0.01	96.04±0.17	96.46±0.01	96.65±0.08
	30%	84.94	87.91	85.57	90.07	91.98	90.49	29.45	94.73±0.01	94.82±0.14	95.54±0.01	**95.76±0.03**
	40%	81.94	86.23	82.61	87.70	90.58	88.66	29.45	95.77±0.04	95.68±0.21	**95.97±0.01**	95.91±0.02
	50%	78.68	84.00	78.87	84.78	88.19	86.16	29.45	93.54±0.00	93.80±0.19	93.76±1.12	**94.41±0.01**
	60%	76.79	83.23	77.18	83.17	87.17	85.84	29.45	94.70±1.35	94.89±0.23	94.81±1.31	94.97±0.04
	70%	75.35	82.45	75.84	81.87	86.48	84.72	29.45	94.45±0.05	94.63±0.19	94.91±0.04	**95.33±0.02**
	80%	73.22	81.06	73.54	79.86	84.57	83.11	29.45	93.96±0.02	93.87±0.17	94.13±0.01	**94.51±0.12**

Continued on next page

Table 5.6 – continued from previous page

		Algorithm										
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**WPS-FCM**	**AWA-FCM**	**BWPS-FCM**
B147091	10%	87.83	88.55	87.90	88.72	88.81	89.40	88.88	89.43±0.02	89.52±0.11	89.68±0.01	**89.82±0.01**
	20%	85.12	86.74	85.12	87.23	88.11	88.25	65.41	89.06±0.06	89.34±0.13	89.56±0.05	**89.78±0.04**
	30%	82.34	84.69	82.23	85.03	86.78	86.01	42.74	88.21±0.02	88.50±0.13	88.66±0.01	**88.72±0.09**
	40%	80.74	84.15	80.92	84.25	86.38	86.01	42.74	88.61±0.07	88.80±0.27	**89.06±0.03**	88.91±0.06
	50%	78.47	82.72	78.36	82.64	8high5.40	84.13	42.74	88.84±0.00	88.75±0.11	89.49±0.00	89.48±0.04
	60%	76.37	81.41	76.26	81.10	84.26	83.02	42.74	88.31±0.01	87.84±0.40	**88.65±0.00**	88.54±0.13
	70%	74.87	80.71	74.68	79.93	83.55	82.06	42.74	87.50±0.05	87.48±0.13	**87.69±0.02**	87.63±0.02
	80%	73.53	79.50	73.46	78.81	82.57	80.60	42.74	88.34±0.01	88.53±0.07	88.78±0.02	**88.89±0.14**
B108073	10%	78.11	77.06	80.78	82.86	83.68	80.08	**87.13**	82.41±0.16	82.48±0.27	83.56±0.04	84.43±0.04
	20%	72.42	73.38	75.01	78.57	79.73	76.00	**86.22**	79.63±0.10	79.63±0.15	80.89±0.11	81.55±0.12
	30%	69.52	71.56	71.44	75.79	77.13	74.20	77.45	79.97±0.19	79.97±0.37	81.44±0.08	**81.92±0.04**
	40%	67.13	70.31	69.26	72.70	74.92	73.32	77.45	79.99±0.02	80.13±0.25	81.01±1.22	**81.34±0.09**
	50%	65.76	69.00	67.36	70.59	73.98	70.91	77.45	80.21±0.01	80.18±0.38	81.73±0.01	**81.81±0.05**
	60%	64.10	67.66	65.37	68.64	70.67	69.32	77.45	76.21±0.85	76.39±0.16	77.60±0.06	**77.91±0.09**
	70%	63.37	67.62	64.60	67.77	70.51	69.26	77.45	80.05±0.14	80.16±0.29	**81.26±0.12**	81.17±1.24
	80%	63.05	67.47	64.32	67.33	69.55	69.16	77.45	81.40±1.21	81.69±0.22	82.88±0.03	**83.10±0.03**
B135069	10%	58.28	60.61	59.40	64.99	69.70	**72.74**	5.23	66.49±0.01	66.88±0.44	67.92±0.00	68.48±0.09
	20%	55.19	57.95	55.79	60.00	63.99	66.68	5.23	68.13±0.04	68.25±0.54	67.70±0.02	67.82±0.36
	30%	53.39	56.13	51.88	56.99	61.40	63.04	5.23	61.01±0.61	61.42±0.40	**64.68±0.01**	63.24±0.06
	40%	52.86	56.10	52.25	56.17	62.44	61.20	5.23	60.13±0.05	60.83±0.79	63.51±0.97	**64.23±0.15**
	50%	52.52	55.63	51.99	55.54	59.66	59.62	5.23	63.68±0.05	63.98±0.76	65.63±1.19	**66.60±0.36**
	60%	52.27	55.58	50.86	54.43	59.86	59.29	5.23	64.91±0.33	64.53±1.38	64.94±0.10	**67.23±0.11**
	70%	52.47	55.69	51.13	54.73	60.11	58.89	5.23	61.36±0.04	61.73±0.96	64.40±0.06	**65.05±0.17**
	80%	51.95	55.33	51.00	54.58	60.59	58.95	5.23	67.29±1.23	67.66±0.38	69.43±0.04	**70.02±0.25**

BWPS-FCM vs Other Methods in BerkDB

Table 5.7 provides the results of the Wilconxon test on the pair-wise comparison of BWPS-FCM vs other methods. This table shows that there are a total of 20 pair-wise comparisons in which the *p* value is bigger than the significance level. These cases belong to the comparison of BWPS-FCM to, FGFCM_S1 (one case), FLICM (one case), WS-FCM (four cases), WPS-FCM (six cases), and AWA-FCM (8 cases). This means that out of 880 pair-wise comparisons maid, there are 20 cases that BWPS-FCM performs not significantly different from the aforementioned methods. Putting aside these cases, for the rest of the comparisons BWPS-FCM is performing always/mostly better than the other methods. More importantly, BWPS-FCM outperforms AWA-FCM in this dataset.

BWPS-FCM outperforms the performance of FCM_S1, FCM_S2, EnFCM, and FGFCM on all the 88 instances. It outperforms the other FCM-objective-modified algorithms FGFCM_S1, FGFCM_S2, and FLICM, on 86, 87, and 68 instances respectively. Also, BWPS-FCM outperforms, the adaptive shrinkage-based algorithms, WS-FCM, WPS-FCM, and the powerful AWA-FCM on 70, 67, and 51 instances respectively.

To determine the overall best performer based on the highest number of best and second-best performances, the best performer of all is the proposed BWPS-FCM being the best performer on 42 instances, and second-best on 13 further instances. In this manner, the overall second-best performer is FLICM with 19 instances as the best performer, and two further instance as the second-best performer. The third- and fourth-best performer are respectively AWA-FCM and WS-FCM having 12 and five best performances respectively, and 46 and five further second-best performances. Therefore, in addition to the previous pair-wise comparison, the mentioned overall best comparison also specifies BWPS-FCM as the best noisy image segmentation algorithm in the BerkDB dataset.

Table 5.7: *p* values provided by the Wilcoxon significance test for the BerkDB dataset corrupted with Gaussian noise. The bold numbers indicate *p* values greater than the significance level.

		Algorithm									
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**WPS-FCM**	**AWA-FCM**
B3096	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	4.3e-06 (+)	0.003 (+)	3.1e-05 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.25** (=)
B8068	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.0013 (+)	1.7e-06 (+)	1.7e-06 (-)	0.002 (+)	0.0047 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	0.0058 (-)	8.2e-05 (+)	0.00027 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	**0.53** (=)	0.005 (+)	1.7e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	9.7e-05 (-)	1.9e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00042 (+)	0.009 (-)	1.7e-06 (-)
B42049	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00022 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.39** (=)	1.7e-06 (+)	1.7e-06 (+)	7.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00011 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B167062	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.3e-05 (+)	1.7e-06 (+)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.21** (=)	1.7e-06 (+)	1.7e-06 (-)	**0.2** (=)	**0.21** (=)	1.7e-06 (-)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	**0.057** (=)	**0.057** (=)	**0.057** (=)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	**0.98** (=)	**0.67** (=)	**0.85** (=)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	0.0012 (+)	2.1e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	0.015 (-)	0.015 (-)	0.0012 (-)
B86016	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	3.1e-05 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	8.5e-06 (-)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	0.00083 (-)	5.2e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	3.1e-05 (-)	1.7e-06 (-)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	0.028 (-)	3.1e-05 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.4e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
B196027	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.084** (=)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	6.3e-05 (-)	0.00036 (-)	**0.11** (=)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.00028 (-)	**0.2** (=)	**0.92** (=)
	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)

Continued on next page

Table 5.7 – continued from previous page

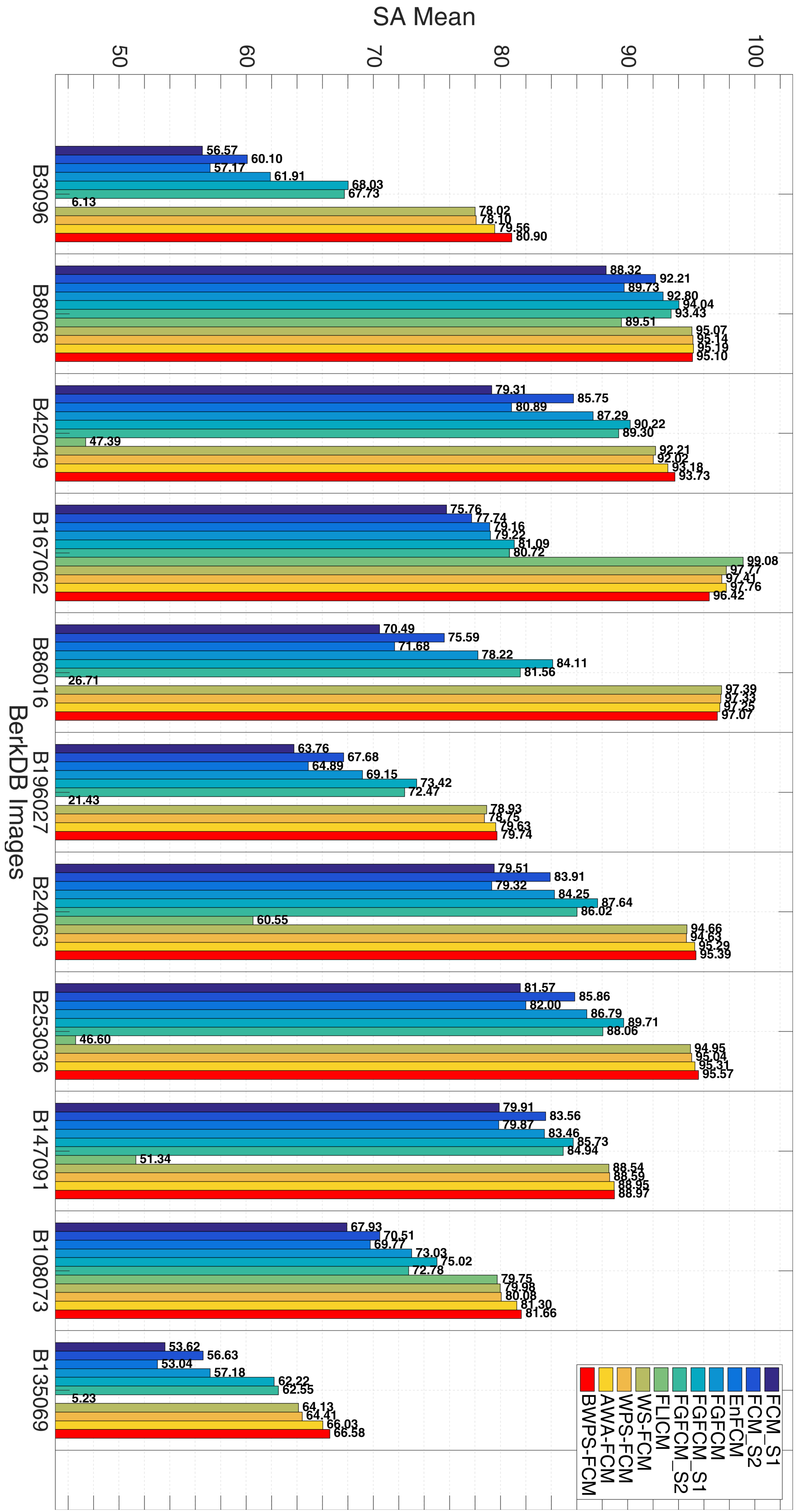
		Algorithm									
Img.	**Vol.**	**FCM_S1**	**FCM_S2**	**EnFCM**	**FGFCM**	**FGFCM_S1**	**FGFCM_S2**	**FLICM**	**WS-FCM**	**WPS-FCM**	**AWA-FCM**
B24063	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.5e-05 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.27** (=)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.0023 (-)
B253036	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	2.1e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	4.7e-06 (+)	1.9e-06 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.029 (+)	**0.15** (=)	0.00023 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B147091	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	0.019 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.22** (=)	2.3e-05 (-)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	**0.16** (=)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	3.1e-05 (+)	1.9e-06 (+)	1.9e-06 (-)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	4.3e-06 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	6.3e-06 (+)	3.1e-05 (+)
B108073	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	4.3e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	3.1e-05 (+)	3.1e-05 (+)	0.001 (-)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
B135069	10%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	1.7e-06 (-)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	20%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)	1.7e-06 (-)	0.00066 (-)	0.015 (+)
	30%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (-)
	40%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	50%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.6e-06 (+)
	60%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)
	70%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.9e-06 (+)	1.7e-06 (+)
	80%	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	1.7e-06 (+)	2.6e-06 (+)

The final quantitative experiment shows the performance of the proposed BWPS-FCM over noise level variation compared to other methods. Fig. 5.9 provides the mean and the variance of the SA values while the noise level changes. Fig. 5.9(a) shows the mean values according to which in 8 (out of 11) images the proposed BWPS-FCM has the highest mean values. AWA-FCM, WS-FCM, and FLICM each have one highest mean performance. Therefore, BWPS-FCM is the best performer having the largest number of the highest mean performances. Also, Fig. 5.9(a) shows that WPS-FCM has a higher mean performance on six images versus five higher mean performances of WS-FCM. This indicates that in this quantitative experiment (the largest number of the highest mean performances), WPS-FCM performs better than WS-FCM. For the lowest variance of the SA value over noise variation, Fig. 5.9(b) is provided. This bar graph indicates that each of BWPS-FCM, AWA-FCM, and WS-FCM has the lowest SA variance on three images, and each of WPS-FCM and FLICM has the lowest SA variance on one image. This way the best performer ranking (considering the largest number of the lowest variance of SA), jointly belongs to BWPS-FCM, AWA-FCM, and WS-FCM algorithms.

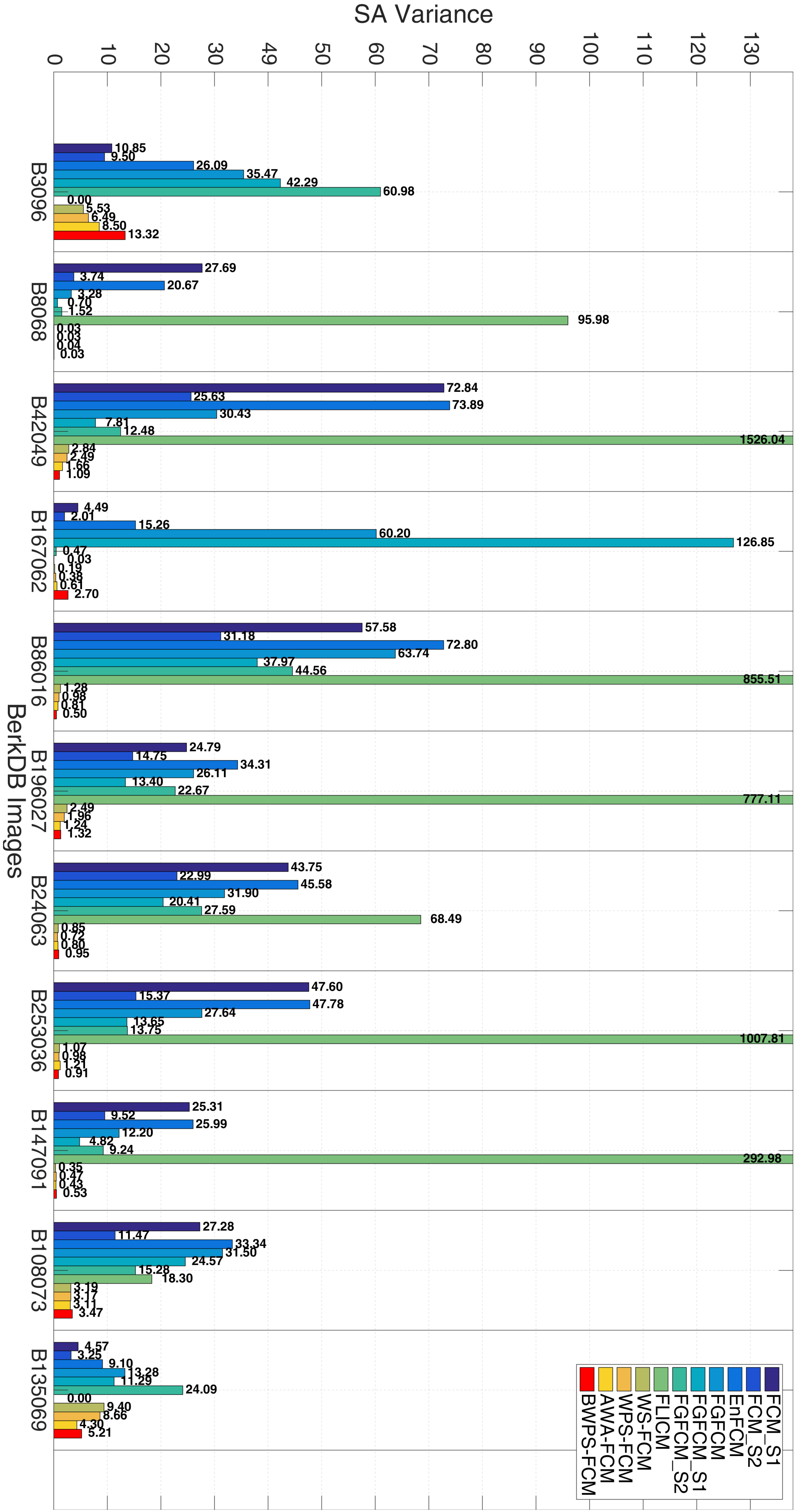
BerkDB (Qualitative Analysis)

Some segmentation results from all the algorithms are shown in Fig. 5.10 in which an instance is selected from each image (due to space limitations the results are shown in this order: first, images B3096, B42049, B86016, and B135069, then images B8068, B24063, B253036, and B167062, and finally images B147091, B108073, and B196027). FCM-objective-modified methods such as FCM_S1 and FCM_S2, EnFCM, FGFCM, FGFCM_S1 and FGFCM_S2 suffer from over-segmentation even on noisy images with low volume of noise. Redundancy of spurious regions caused by noise makes the detection of the main regions almost impossible. FLICM provides a mixture of bad and good results. There are instances such as the one for image B167062 that the method has outstanding results, and there are instances such as the ones related to images B3096, B42049, B86016, B135069, and more that the method produces uninterpretable results for not being able to comply with the required number of clusters in the segmented image.

Generally, WPS-FCM produces results that show better performance on preserva-


SA Mean
50
60
70
80
90
100
BerkDB Images
B3096
56.57
60.10
57.17
61.91
68.03
67.73
6.13
78.02
78.10
79.56
80.90
B8068
88.32
92.21
89.73
92.80
94.04
93.43
89.51
95.07
95.14
95.19
95.10
B42049
79.31
85.75
80.89
87.29
90.22
89.30
47.39
92.21
92.02
93.18
93.73
B167062
75.76
77.74
79.16
79.22
81.09
80.72
99.08
97.77
97.41
97.76
96.42
B86016
70.49
75.59
71.68
78.22
84.11
81.56
26.71
97.39
97.33
97.25
97.07
B196027
63.76
67.68
64.89
69.15
73.42
72.47
21.43
78.93
78.75
79.63
79.74
B24063
79.51
83.91
79.32
84.25
87.64
86.02
60.55
94.66
94.63
95.29
95.39
B253036
81.57
85.86
82.00
86.79
89.71
88.06
46.60
94.95
95.04
95.31
95.57
B147091
79.91
83.56
79.87
83.46
85.73
84.94
51.34
88.54
88.59
88.95
88.97
B108073
67.93
70.51
69.77
73.03
75.02
72.78
79.75
79.98
80.08
81.30
81.66
B135069
53.62
56.63
53.04
57.18
62.22
62.55
5.23
64.13
64.41
66.03
66.58
FCM_S1
FCM_S2
EnFCM
FGFCM
FGFCM_S1
FGFCM_S2
FLICM
WS-FCM
WPS-FCM
AWA-FCM
BWPS-FCM

(a) SA mean.



(b) SA variance.

Figure 5.9: Demonstration of different algorithms performance on noise level variation (BerkDB).

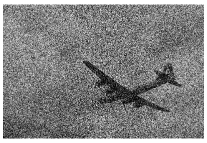
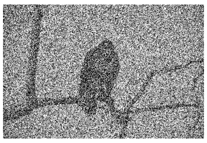
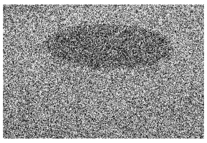
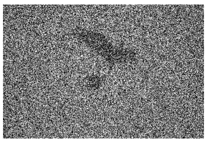
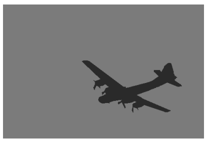
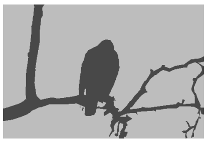
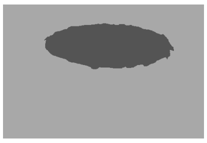
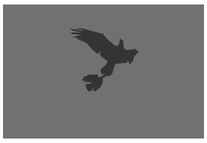
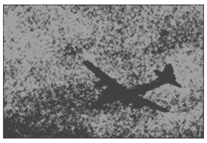
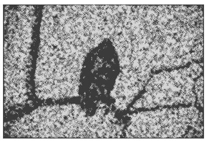
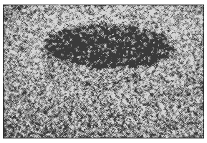
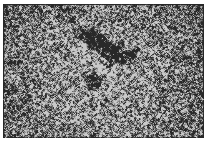
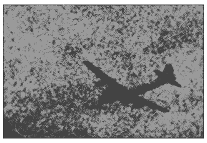
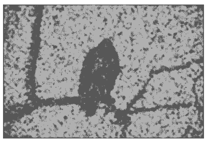
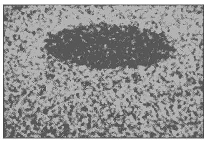
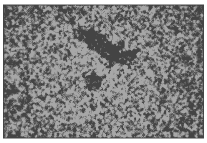
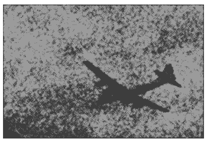
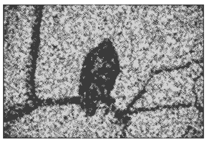
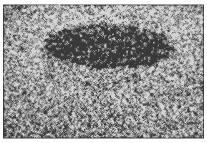
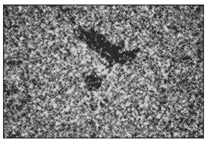
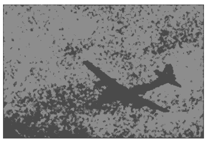
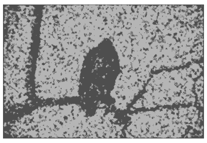
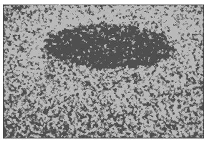
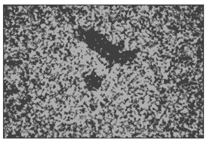
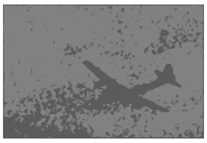
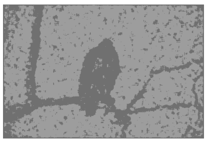
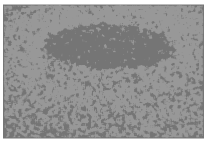
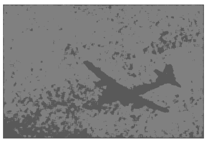
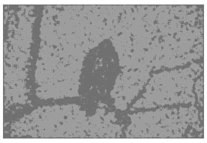
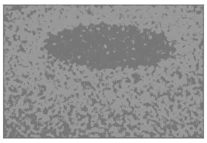
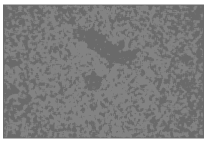
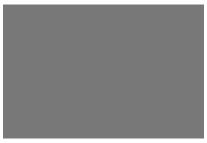
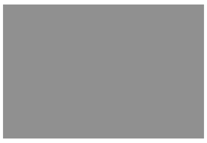
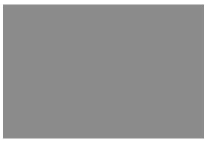
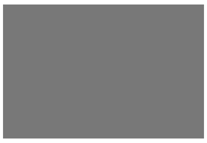
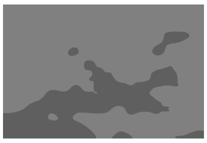
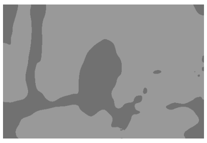
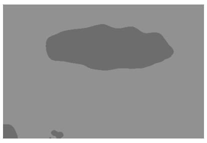
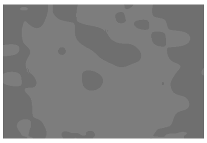
tion of edges compared to WS-FCM. This is clearly visible on the instances of images $B3096$, $B42049$, $B253036$, $B147091$, $B108073$, and $B196027$. From the detail-preserving perspective, BWPS-FCM shows better performance than other adaptive wavelet-shrinkage-based methods (WS-FCM, WPS-FCM, and AWA-FCM). This is observable on the instances of images $B42049$, $B253036$, and $B108073$. However, BWPS-FCM produces more redundant regions such as the ones in the instance of $B108073$. Since the results produces by BWPS-FCM still have higher SA values compared to other methods in the majority of the instances, one can realize the importance of edge preserving on noisy segmentation applications.

5.6 Chapter Summary

Motivated by the adaptive wavelet shrinkage algorithm proposed in chapter 3 (WS-FCM), we introduced another algorithm named WPS-FCM using wavelet packets as a domain for feature enhancement. WPS-FCM did not produce overall better results than its counterpart, WS-FCM, based on the number of better performances according to the SA metric on the instances of the SynthDB and BerkDB. However, considering the instances related to each image, WPS-FCM performs better than WS-FCM in BerkDB by holding a larger number of better performances, as well as possessing a higher mean performance over the noise level variation. In addition, qualitatively, WPS-FCM tends to preserve more details related to edges and boundaries compared to WS-FCM.

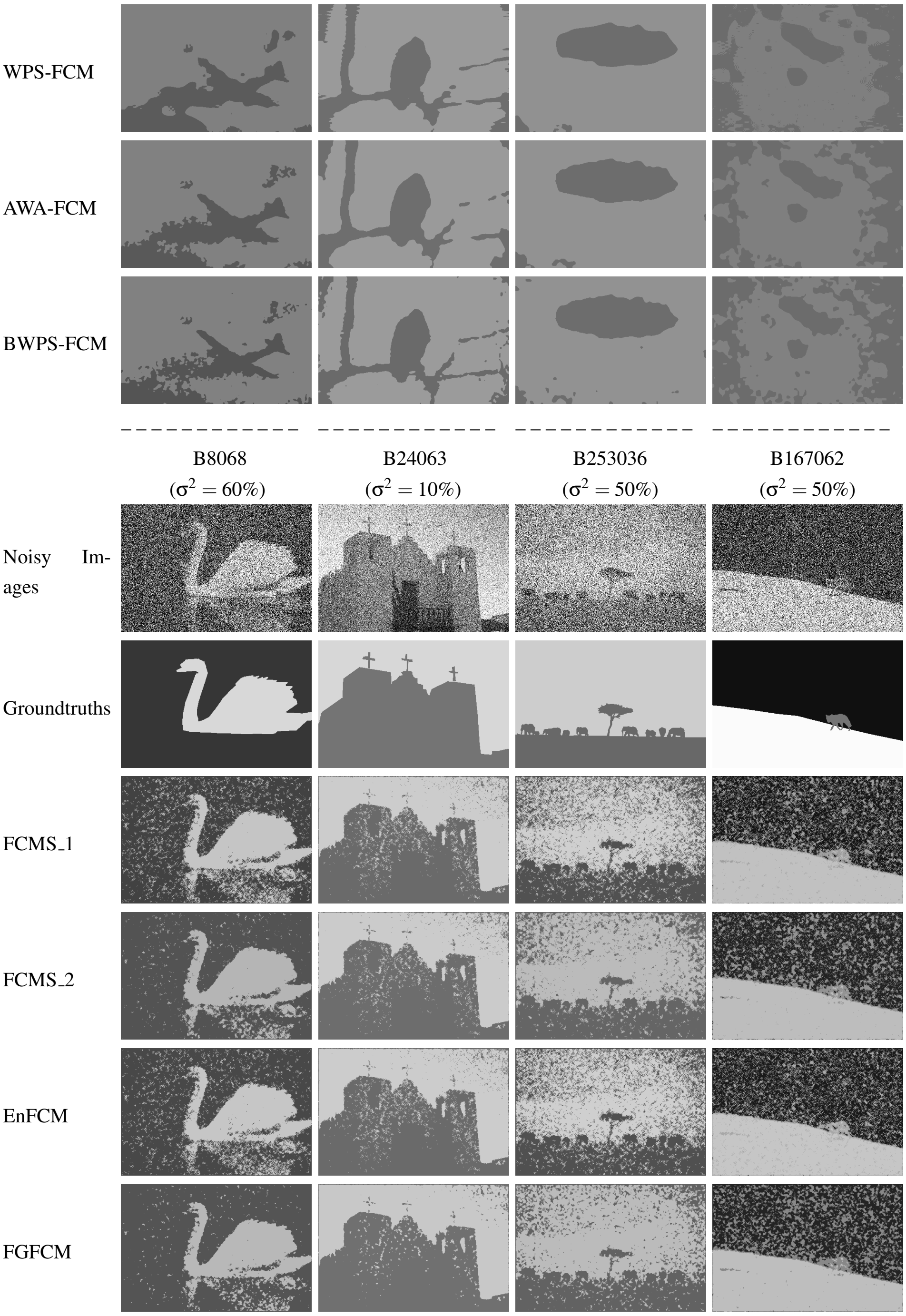
Next, using the same idea as in WPS-FCM, we introduced another algorithm termed BWPS-FCM with two differences. Naturally noisy images, or images destroyed by severe noise volume may suffer from non-sparsity. A pre-denoising step is applied using bilateral filtering which returns the sparsity back to the wavelet coefficients. Also, a new thresholding scheme based on smooth sigmoid-based shrinkage was designed to overcome the problems caused by zero-crossing and attenuation of soft-thresholding function. Like the other previous wavelet-shrinkage-based algorithms introduced in this thesis, BWPS-FCM is parameter-free for the volume of noise, is effective in severely noisy image segmentation, and showed consistency with noise volume variations.

Overall, BWPS-FCM outperforms WS-FCM and WPS-FCM quantitatively and qualitatively. However, in comparison to AWA-FCM proposed in Chapter 4, it showed different behaviors in SynthDB and BerkDB. In the SynthDB, since the regions are single-

	B3096 ($\sigma^2 = 10\%$)	B42049 ($\sigma^2 = 80\%$)	B86016 ($\sigma^2 = 70\%$)	B135069 ($\sigma^2 = 60\%$)
Noisy Images				
Groundtruths				
FCMS_1				
FCMS_2				
EnFCM				
FGFCM				
FGFCMS_1				
FGFCMS_2				
FLICM				
WS-FCM				

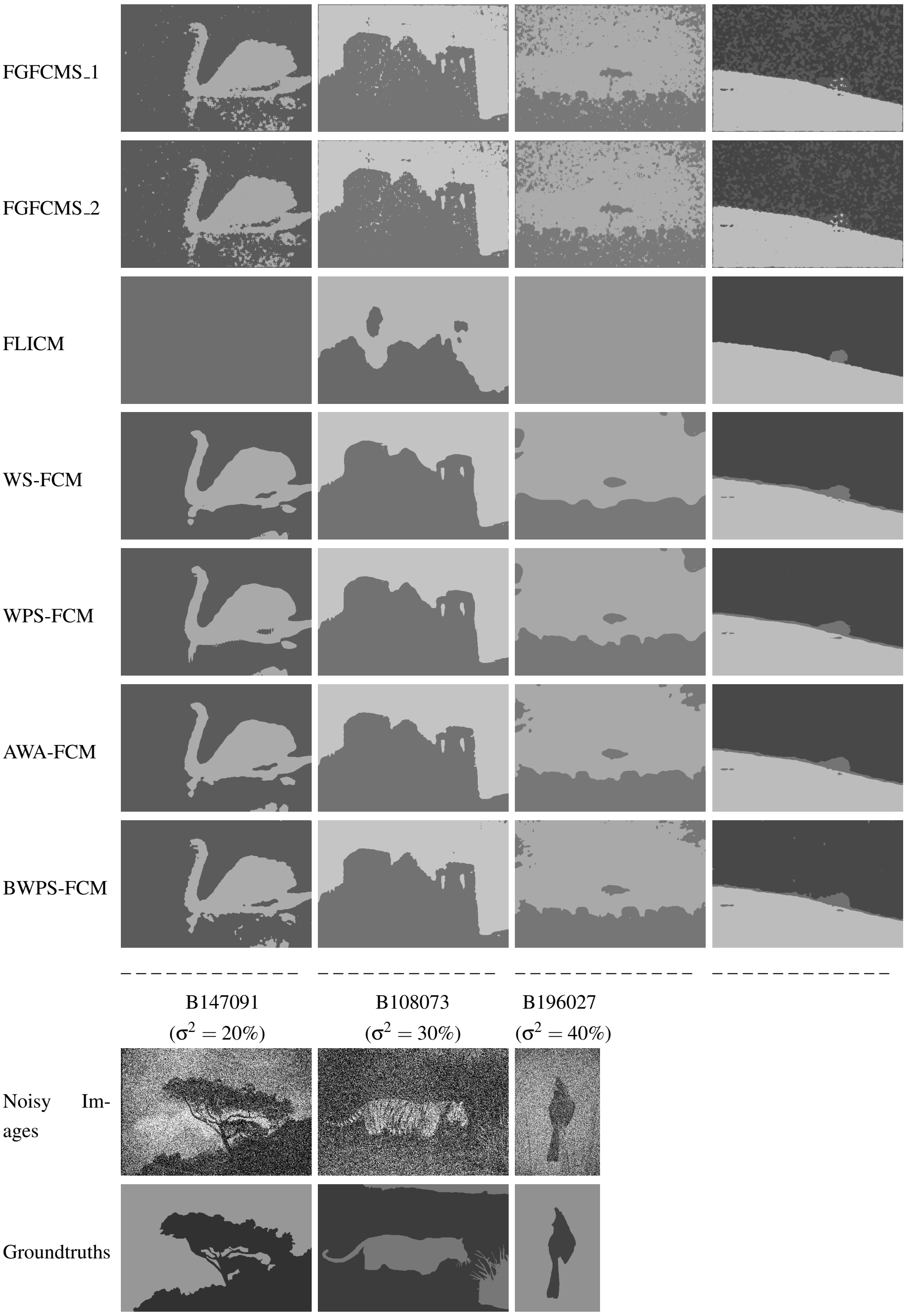
Continued on next page

Figure 5.10 – continued from previous page



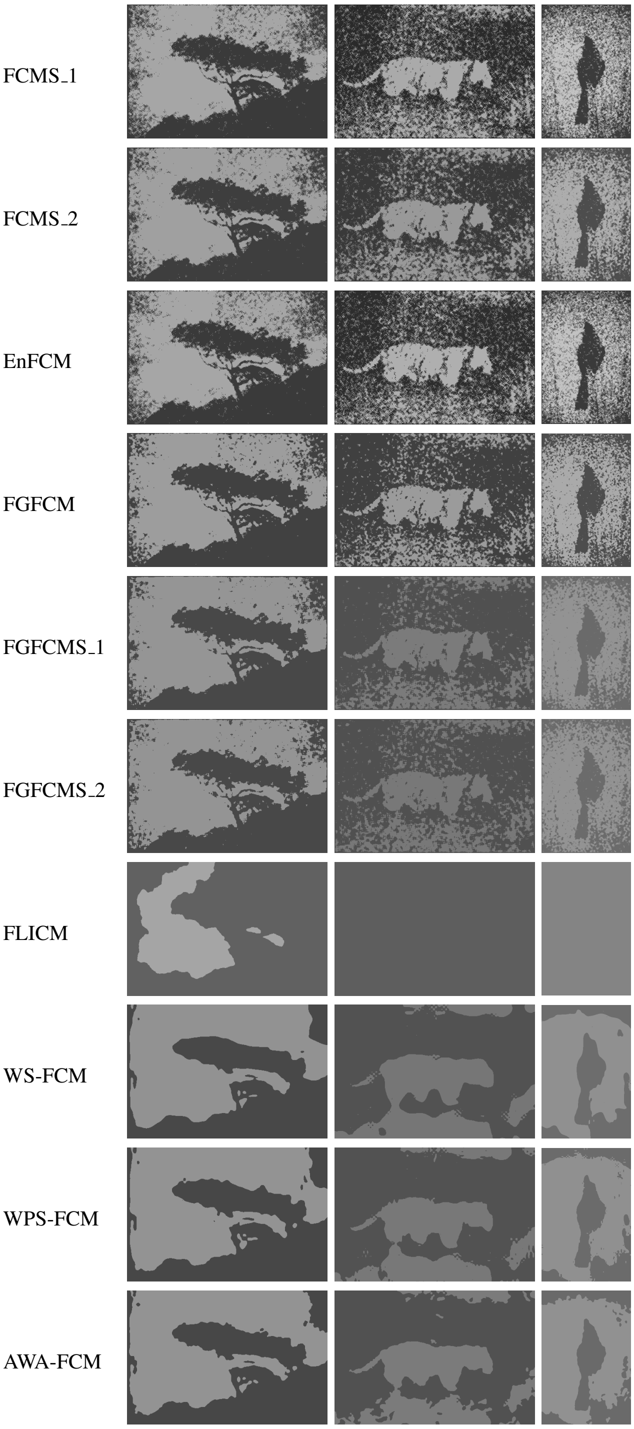
Continued on next page

Figure 5.10 – continued from previous page



Continued on next page

Figure 5.10 – continued from previous page



Continued on next page

Figure 5.10 – continued from previous page

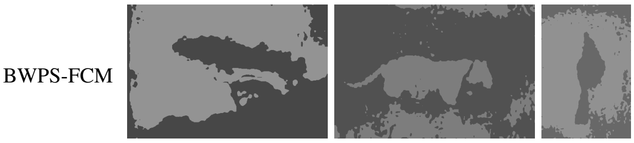


Figure 5.10: Qualitative comparison of the proposed BWPS-FCM with FCM_S1, FCM_S2, EnFCM, FGFCM_S1, FGFCM_S2, FGFCM, FLICM, WS-FCM, WPS-FCM and AWA-FCM on some instances from the BerkDB. *B*3096, *B*8068, *B*42049, *B*167062, *B*86016, *B*196027, *B*24063, *B*253036, *B*147091, *B*108073, and *B*135069 are corrupted by Gaussian noise with variances of 10%, 60%, 80%, 50%, 70%, 40%, 10%, 50%, 20%, 30%, and 60% respectively.

intensity, noise corruption does not cause as much damage it causes in multi-intensity regions. In an FCM-based segmentation scheme, this is justifiable by the experimental intuition that more distinguishable regions can be segmented in SynthDB compared to the BerkDB. Having said that, in SynthDB, it is the new thresholding scheme that has more effect than the pre-denoising step. In fact, in this dataset there is nothing the preprocessing step can do that wavelet packet shrinkage can not do. Overall, AWA-FCM performs better than BWPS-FCM in SynthDB. On the other hand, in BerkDB where the natural images have multi-intensity distinguishable regions, the pre-denoising contributes more. The pre-denoising step provides better thresholding process by creating a better sparsity of coefficients. This causes the overall better performance of BWPS-FCM compared to AWA-FCM in BerkDB.

Chapter 6

Conclusion and Future Work

This chapter is dedicated to highlighting the achieved objectives, main conclusions, and future research directions. This thesis was focused on PSO for FCM-based noisy image segmentation. The overall goal of the thesis was to develop new domain-independent PSO approaches for an automatic non-supervised FCM-based segmentation of severely noisy images which is capable of extracting main coherent/homogeneous regions while preserving details and is robust to noise variation. We have successfully achieved this goal in this thesis where domain-independence was achieved by using image raw pixels directly as the input of the developed systems, although due to the usage of FCM, we need to know the preferred number of clusters in advance. The key approach taken in the thesis is to explore the use of PSO to manipulate and enhance local spatial and spatial-frequency information. To this end, a number of approaches were successfully developed in this thesis using PSO. The proposed methods were evaluated extensively using different image segmentation benchmarks, and were compared to state-of-the-art algorithms. The results showed that appropriate utilization of PSO for feature enhancement and proper incorporation of other detail-preserving methods in noisy images produce mostly better performances compared to state-of-the-art algorithms.

6.1 Answers to the Proposed Research Questions

We raised three research questions in the first Chapter that are answered here based on our achievements in the Chapters 3, 4, and 5:

(i) *"How can PSO be used in an FCM-based noisy image segmentation system for an effective enhancement of features?"*

Response: We demonstrated that PSO can be effectively used for adaptive thresholding in both wavelet and wavelet packet domains. Using a specific filter (Coiflet family in this thesis) in five-scales transformation provides subbands that can be searched for proper threshold values using PSO. Particles in PSO are in fact sets of threshold values that can swing in a predefined interval and are evolved using a FCM clustering metric.

(ii) *"How can details be preserved in a noisy image segmentation? How can we improve the results of the previously created system with a stress on detail preservation?"*

Response: We found that preserving details in severely noisy image segmentation can be a very difficult task. Common approaches such as edge detection or gradient-based methods do not work. We introduced detail-preserving algorithms in this thesis using edge detection and edge-preservative denoising methods tailored through many qualitative and quantitative experiments based on intuitive and logical processes.

(iii) *"How could a detail-preserving feature-enhancing system cope with noise type or volume variations while remaining consistently effective and parameter-tuning-free?"*

Response: To make the proposed algorithms effective on images with different levels of noise we introduced two nonlinear noise estimating systems based on the variance of noise measurable in the wavelet domain. Again, extensive empirical and hypothetical experiments were conducted to make the systems practical. Also, the adaptive nature PSO helps immensely with the accurate segmentation in images with noise level or type variation.

6.2 Achieved Objectives

The fulfilled research objectives this thesis presents are as follows:

- The thesis proposes a new PSO approach for adaptive shrinkage of wavelet coefficients for enhancement of features in an FCM-based noisy image segmentation process. PSO adaptively evolves threshold values that have a proper denoising performance using a soft-thresholding function and FCM clustering evaluation. Next, we extend the method by incorporating the well-known Canny edge detector. The extension benefits from adaptive thresholding of detail coefficients and edge enhancement on some coefficients that are less likely to be corrupted by noise. This causes the extended algorithm to perform slightly better than its predecessor by preserving more edge information. The new algorithms perform well in elimination of severe noise and consequently severely noisy image segmentation particularly for Gaussian noise. Also, a very good consistency of segmentation accuracy is observable when noise level varies. Visually, this approach performs well on extracting the main objects/regions while not so much respectful of small region, edges, and details.

- This thesis proposes a new combination of two denoising algorithms for detail-preserving in severely noisy image segmentation. The new method uses the adaptive wavelet shrinkage PSO approach introduced above and the edge-preserving anisotropic diffusion. This approach aims at returning some of the details that exist in the anisotropic diffusion denoising results, and are removed in the PSO approach, without returning back the noise. To this end, we introduced a new scheme to fuse the coefficients of both denoising results in the wavelet domain. To remove side effects of the fusion process, a final elimination of small regions based on connected component analysis is performed. The new method benefits from the advantages of both denoising algorithms, and partially overcomes the drawbacks of each. Details related to boundaries are preserved better, and segmentation accuracy is higher compared to both primary algorithms.

- This thesis introduces a new PSO approach for adaptive shrinkage of wavelet packet coefficients for feature enhancement in an FCM-based noisy image segmentation. We use wavelet packets as another tool for multiscale analysis. The new approach includes formation of an optimum wavelet packet decomposition tree based on Shannon entropy, and adaptive shrinkage of different subbands in the optimum tree using PSO. The threshold values are again evaluated using

FCM clustering performance. Next we extend the approach by adding a pre-denoising step, and a new shrinkage scheme. The pre-denoising step uses an edge-preserving bilateral filtering, and the new thresholding scheme uses smooth sigmoid-based shrinkage. By a proper thresholding and detail preservation processes, the new approach achieves the highest segmentation accuracy so far on some of the image benchmarks (in comparison to algorithms proposed in this thesis and the state-of-the-art comparison algorithms) .

6.3 Summary of the Research

Overall, this thesis finds that PSO can be effectively used for feature enhancement to FCM-based noisy image segmentation. While the common approach is to modify FCM objective in order to extract and enhance features from a local neighboring window around each pixel, we showed that keeping the original FCM but using a proper feature enhancement scheme can result in outstanding performances. This thesis also introduces a number of approaches to enhance the edge information in segmentation results. The new approaches introduced in this thesis are parameter-tuning free for the volume of noise (sometime the type of noise), effective in case of severe noise, and consistent with noise volume variations. In the following, we discuss the detailed conclusions from each contribution chapter.

6.3.1 Adaptive Wavelet Shrinkage for Severely Noisy Image Segmentation

Chapter 3 introduces a new PSO approach for adaptive shrinkage of wavelet coefficients, and the extension of the approach introduces the application of edge detection to edge enhancement.

Wavelets

Thresholding wavelet coefficients could result in proper image denoising. The scheme for thresholding plays an important role in the results. The number of scales, number of threshold values, and the thresholding function are the factors to be investigated precisely. We showed the two dimensional discrete wavelet transform in a five-scale transformation can provide a proper decomposition of a noisy image. In our approach, there is a threshold value for each subband, and the soft-thresholding function [47] is used. Any other settings would result in over/under segmentation, expensive computation, or unstable results. The choice of wavelet filter also has different segmentation results. Denoising, compression or segmentation, each require their own choice of filter. We have used the *Coiflet* filter family in our noisy image segmentation approach.

PSO Representation and Fitness Evaluation

Having a threshold value for each subband in a five-scale transformation requires 15 threshold values, which need proper tuning otherwise they result in over-smoothed or over-segmented images. An array of 15 threshold values can effectively and efficiently be represented as particles in the PSO search space in our approach. We set restrictions for threshold vales to keep them positive and less than the *Universal* threshold [47]. This helps proper thresholding while prevents over-smoothing. The fitness metric provided by the FCM objective function guides the search effectively to find good threshold values. We also found that, PSO search guided by an unsupervised denoising performance as in [148] under the provided scheme eliminates many of the large coefficients and results in over-smoothed segmentation results. in

Edge Enhancement

This part is an extension to the the above adaptive wavelet shrinkage algorithm by adding up some enhanced edge information. For this extension, we introduce an scheme under which edge detection is incorporated into wavelet thresholding. Being applicable in the wavelet domain makes the extension simple and efficient because no extra transformation is required, and the only extra computational cost comes from the edge detection process. Applying edge detection algorithms on detail coefficients in case of severe noise produces many redundant regions in segmentation results. Our scheme aggregates the approximation coefficients with an edge map obtained from these coefficients. To obtain the edge map, Canny edge detector is applied to approximation coefficients. Results show that emphasizing on edge coefficients before the reconstruction process leads to more accurate edge information, and therefore better segmentation accuracy.

6.3.2 Wavelet Shrinkage and Anisotropic Diffusion for Severely Noisy Image Segmentation

Chapter 4 introduces an edge-preserving noisy image segmentation to address the over-smoothing drawback of the previously proposed approach. Soft-thresholding causes over-smoothing of edges due to attenuation of large coefficients. In an attempt to an edge-respectful approach, we use anisotropic diffusion in integration with the PSO approach introduced in the previous chapter. The PSO approach performs well in noise elimination and anisotropic performs well in edge preservation.

Anisotropic Diffusion

We found that the iterative edge-preserving denoising algorithm, anisotropic diffusion [101], has the abilities to enhance features in an FCM-based severely noisy image segmentation scheme. However, it needs parameter tuning according to noise volume. This specific parameter is the number of iterations according to which the filter has to apply to the image. Larger volumes of noise require larger numbers of iteration. Experimentally, we found a quadratic curve according to which this number could be estimated. This number of iterations does not guarantee the best performance of anisotropic diffusion, but results in a good denoised image with enough edge information.

Fusion Scheme

The denoise image resulted from anisotropic diffusion and the thresholded coefficients resulted from the PSO approach are fused in wavelet domain. After wavelet decomposition of the resulted image from the anisotropic diffusion process, detail coefficients from the two methods are aggregated with different weightings. We found that this aggregation not only adds more edge information to the image, but also removes some of the noise existing in the anisotropic denoised image. However, some noise is yet added to the reconstructed image as a side effect of the fusion process. This results in some small redundant regions in the segmentation results. These redundant regions are easily detectable, because they are sparsely located in other wider segments belonging to main regions or objects. We introduced a new elimination strategy using the concept of connected component analysis which can safely eliminate these redundant regions.

6.3.3 Wavelet Packets and Bilateral Filtering for Edge-Preserving Severely Noisy Image Segmentation

The final contribution chapter, Chapter 5, introduces a PSO approach for adaptive thresholding of wavelet packet coefficients, similar to the PSO approach in Chapter 3. We then extend the approach by adding a preprocessing step based on bilateral filtering, and a final sigmoid-based shrinkage function.

Wavelet Packets

Wavelet packet transform decomposes low-frequency as well as high-frequency subbands in each scale. This provides a finer frequency resolution domain of high frequency content. Since there are too many nodes in a wavelet packet decomposition tree, for efficiency matters, an optimal tree needs to be formed. This optimal tree is formed through minimizing an additive energy function using the Shannon entropy [39]. Having a threshold value for each scale, and restricting them as before (the maximum limit is different this time) resulted in the best enhancement of features under the PSO search space and FCM clustering evaluation for Gaussian corrupted noisy images. Although wavelet packets are known to provide a richer domain analysis, under the proposed scheme, they could not quantitatively produce better results than the wavelet approach in Chapter 3. However, in a visual comparison, they could preserve more edge information in some cases.

Bilateral Filtering

Bilateral filtering is another edge-preserving denoising algorithm applicable in spatial domain. We found that a pre-denoising step before the wavelet packet approach can make the wavelet packet coefficients more sparse. Sparsity might be eliminated in natural images or severely corrupted noisy images. This preprocessing step is designed to only partially remove noise. Again, predicting the parameters related to the algorithm while noise volume increases is required. Experimentally, we showed that a sigmoid function could be used to tune the parameter (width of spatial Gaussian) related to the specific bilateral filtering [35] we used in this approach.

Smooth Sigmoid-Based Shrinkage

PSO uses soft-thresholding in evaluation process which causes over-smoothing of edges. We use a new scheme for thresholding of coefficients based on smooth sigmoid-based shrinkage [9] that addresses this issue to some extent. We found that threshold values obtained via a soft-thresholding function could be safely used in the sigmoid thresholding function. In other words, threshold values obtained from a soft-thresholding process are large enough to suppress the effect of noise, but they are not properly used in confrontation with large coefficients. In the new scheme, nodes with many large coefficients are treated cautiously by having a low-slope transiting sigmoid thresholding function, and other nodes depending on the estimated volume of noise are shrunk by an abrupt transiting sigmoid function that resembles soft-thresholding. This new shrinkage scheme preserve more large coefficients with their original magnitude and therefore preserves more details in the segmentation results.

6.4 Future Work

This section mentions the conceivable key areas for the future work.

6.4.1 Other Multiscale Analysis Tools

One thing we partially investigated in this thesis but has tremendous potential for more contributions is the utilization of other mathematical and computational tools for multiscale analysis. These features are other members of the wavelets family, and they propose different representations of the spatial-frequency transformation. Depending on the scheme, they could lead to better accuracy to FCM-based noisy image segmentation compared to the original wavelets. Here, we briefly mention a few of them.

- **Curvelets**: was first introduced by Candes et al. [27], and then the second-generation 2D discrete transform was presented in [26]. This transform introduces a new architecture to multiscale analysis. Wavelets are only localized in spatial and frequency (or position and scale), but curvelets are also localized in orientation. More particularly, curvelets provide an optimally sparse representation of curves along the boundaries of objects. They could be used as an effective tool for the analysis and computation of partial differential equations. This means that curvelets can act as waveforms that have enough frequency and spatial localization that they can behave as waves (since they have enough frequency localization), and at the same time as particles (since they have enough spatial localization). Also, curvelets have optimal image reconstruction abilities dealing with problems with missing data which is a case usually for severely corrupted noisy images.

- **Contourlets**: was initially introduced with a discrete-domain construction by Minh et al. [44]. It was intentionally introduced in discrete domain to address the challenges in presenting the geometrical properties of discrete data such as images. Two-dimensional wavelets are incapable of seeing smoothness along the contours in natural images. Also, in best case scenario (separable wavelets), wavelets can capture limited directional information. Inspired by characterization of human visual system and statistics of natural images, contourlet, apart from being multiscaled and localized like wavelets, is also directional and anisotropic. Directionality introduces much more basis elements in different directions compared to separable wavelets, and anisotropy helps to capture smooth contours using elongated shapes with different aspect ratios.

- **Shearlets**: motivated by the strengths of multiscale directional tools such as curvelets and contourlets, discrete shearlet transform was proposed by Wanf [81] to address the the drawbacks of band-limited shearlets [77], and other directional systems. The discrete shearlet transform has stable reconstruction properties, well-localized basis elements in both space and frequency domain, efficient implementation and sparse approximation.

- **Other Tools**: Apart from the mentioned tools, there are other multiscale tools for proper representations of geometrical properties of images such as bandelets [100], wedgelets [46], Gabor wavelets [41], the cortex transform [134], and brushlets [92].

6.4.2 Multi-Objective PSO

Having more than one objective in finding the proper threshold values could be another interesting future research direction. Although the clustering metric we used in this thesis provided us with good results, there are other criteria that could evaluate thresholding and segmentation results more accurately. Based on some edge, intensity and region metrics, multi-objective PSO has shown to show promising results [14] in an FCM-based noisy image segmentation paradigm. Introducing new objectives or refining the objectives as in [14] could further improve the segmentation accuracy. To this end, unsupervised segmentation metrics [142], fuzzy validity indices [8, 128], edge detection evaluation metrics such as [140] or other experimental edge-based metrics can be considered.

6.4.3 Color Noisy Image Segmentation

Image segmentation of severely color noisy images using FCM has rarely being investigated in the literature. Extension of the proposed methods in this thesis to the case of severely noisy color images is another interesting field of research. However, there are issues that have to be addressed. For instance, when dealing with gray images, applying any low-pass filters, will produce a blurring effect on edges. This is due to the fact that the smoothing or low-pass filtering results in intensity values between the intensities of edge and non-edge pixels which are yet intermediate gray level values. However, in color images, assuming that there are three color channels, low-pass filtering will produce other colors which are actually disturbing color bands produced by smoothing the edges [123]. This phenomena becomes even more serious in case of severely noisy images with many false and redundant edge information. Therefore, new strategies has to be developed to address this issue.

Bibliography

[1] ABDUL-NASIR, A., MASHOR, M., AND MOHAMED, Z. Colour image segmentation approach for detection of malaria parasites using various colour models and k-means clustering. *WSEAS Transactions on Biology and Biomedicine 10*, 1 (2013), 41–55. cited By (since 1996)0.

[2] ADHIKARI, S. K., SING, J. K., BASU, D. K., AND NASIPURI, M. Conditional spatial fuzzy C-means clustering algorithm for segmentation of {MRI} images . *Applied Soft Computing 34* (2015), 758–769.

[3] AFIFI, A., NAKAGUCHI, T., AND TSUMURA, N. Segmentation of deformable organs from medical images using particle swarm optimization and nonlinear shape priors, 2010.

[4] AGHAJARI, E., AND CHANDRASHEKHAR, G. D. Self-Organizing Map based Extended Fuzzy C-Means (SEEFC) algorithm for image segmentation. *Applied Soft Computing 54* (2017), 347–363.

[5] AGRAWAL, S., PANDA, R., AND DORA, L. A study on fuzzy clustering for magnetic resonance brain image segmentation using soft computing approaches. *Applied Soft Computing Journal 24* (2014), 522–533. cited By 14.

[6] AHMED, M., YAMANY, S., MOHAMED, N., AND FARAG, A. A Modified Fuzzy C-Means Algorithm for MRI Bias Field Estimation and Adaptive Segmentation. vol. 1679 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1999, pp. 72–81.

[7] AMUTHA, S., RAMESH BABU, D. R., MAMATHA, R., VIDHYA SUMAN, S., AND RAVI SHANKAR, M. *Frontier and Future Development of Information*

Technology in Medicine and Education: ITME 2013. Springer Netherlands, Dordrecht, 2014, ch. Speckle Noise Reduction in Breast Ultrasound Images for Segmentation of Region Of Interest (ROI) Using Discrete Wavelets, pp. 747–754.

[8] ARBELAITZ, O., GURRUTXAGA, I., MUGUERZA, J., PÉREZ, J. M., AND PERONA, I. N. An extensive comparative study of cluster validity indices . *Pattern Recognition 46*, 1 (2013), 243–256.

[9] ATTO, A. M., PASTOR, D., AND MERCIER, G. Smooth sigmoid wavelet shrinkage for non-parametric estimation. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing* (March 2008), pp. 3265–3268.

[10] BACK, T., FOGEL, D. B., AND MICHALEWICZ, Z., Eds. *Handbook of Evolutionary Computation*, 1st ed. IOP Publishing Ltd., Bristol, UK, UK, 1997.

[11] BANERJEE, B., VARMA, S., BUDDHIRAJU, K., AND EETI, L. Unsupervised multi-spectral satellite image segmentation combining modified mean-shift and a new minimum spanning tree based clustering technique. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of 7*, 3 (March 2014), 888–894.

[12] BEN CHAABANE, S., FNAIECH, F., SAYADI, M., AND BRASSART, E. Estimation of the mass function in the dempster-shafer's evidence theory using automatic thresholding for color image segmentation. In *Signals, Circuits and Systems, 2008. SCS 2008. 2nd International Conference on* (2008), pp. 1–5.

[13] BENAICHOUCHE, A., OULHADJ, H., AND SIARRY, P. Improved spatial fuzzy c-means clustering for image segmentation using {PSO} initialization, Mahalanobis distance and post-segmentation correction. *Digital Signal Processing 23*, 5 (2013), 1390–1400.

[14] BENAICHOUCHE, A. N., OULHADJ, H., AND SIARRY, P. Multiobjective improved spatial fuzzy c-means clustering for image segmentation combining Pareto-optimal clusters. *Journal of Heuristics 22*, 4 (2016), 383–404.

[15] BENI, G., AND WANG, J. *Swarm Intelligence in Cellular Robotic Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993, pp. 703–712.

[16] BERGHOLM, F., ADLER, J., AND PARMRYD, I. Analysis of Bias in the Apparent Correlation Coefficient Between Image Pairs Corrupted by Severe Noise. *Journal of Mathematical Imaging and Vision 37*, 3 (2010), 204–219.

[17] BEZDEK, J., EHRLICH, R., AND FULL, W. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences 10*, 2-3 (1984), 191–203.

[18] BHANDARI, A., GADDE, M., KUMAR, A., AND SINGH, G. Comparative analysis of different wavelet filters for low contrast and brightness enhancement of multispectral remote sensing images. In *Machine Vision and Image Processing (MVIP), 2012 International Conference on* (Dec 2012), pp. 81–86.

[19] BHANDARI, A. K., KUMAR, A., SINGH, G. K., AND SONI, V. Performance study of evolutionary algorithm for different wavelet filters for satellite image denoising using sub-band adaptive threshold. *Journal of Experimental & Theoretical Artificial Intelligence 0*, 0 (0), 1–25.

[20] BHUTADA, G., ANAND, R., AND SAXENA, S. PSO-based learning of sub-band adaptive thresholding function for image denoising. *Signal, Image and Video Processing 6*, 1 (2012), 1–7. cited By 17.

[21] BONABEAU, E., DORIGO, M., AND THERAULAZ, G. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Inc., New York, NY, USA, 1999.

[22] BOVIK, A. C. *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Academic Press, Inc., Orlando, FL, USA, 2005.

[23] BOYKOV, Y., AND FUNKA-LEA, G. Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision 70*, 2 (Nov. 2006), 109–131.

[24] BOYKOV, Y., AND JOLLY, M.-P. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (2001), vol. 1, pp. 105–112 vol.1.

[25] CAI, W., CHEN, S., AND ZHANG, D. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern Recognition 40*, 3 (2007), 825–838.

[26] CANDES, E., DEMANET, L., DONOHO, D., AND YING, L. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation 5*, 3 (2006), 861–899.

[27] CANDES, E. J., AND DONOHO, D. L. Curvelets: A surprisingly effective nonadaptive representation for objects with edges. Tech. rep., DTIC Document, 2000.

[28] CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 6 (Nov 1986), 679–698.

[29] CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*, 6 (Nov 1986), 679–698.

[30] CARLOTTO, M. J. Model-based detection, segmentation, and classification of compact objects , 2015.

[31] CERCIELLO, T., BIFULCO, P., CESARELLI, M., PAURA, L., ROMANO, M., PASQUARIELLO, G., AND ALLEN, R. *Noise reduction in fluoroscopic image sequences for joint kinematics analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 323–326.

[32] CHAE, S.-H., MOON, H.-M., CHUNG, Y., SHIN, J., AND PAN, S. B. Automatic lung segmentation for large-scale medical image management. *Multimedia Tools and Applications* (2014), 1–17.

[33] CHANG, L.-Y., AND HSU, W. Foreground segmentation for static video via multi-core and multi-modal graph cut. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on* (2009), pp. 1362–1365.

[34] CHAUDHURY, K. N. Acceleration of the shiftable o(1) algorithm for bilateral filtering and nonlocal means. *Trans. Img. Proc. 22*, 4 (Apr. 2013), 1291–1300.

[35] CHAUDHURY, K. N., AND DABHADE, S. D. Fast and provably accurate bilateral filtering. *Trans. Img. Proc. 25*, 6 (June 2016), 2519–2528.

[36] CHAUDHURY, K. N., SAGE, D., AND UNSER, M. Fast o(1) bilateral filtering using trigonometric range kernels. *IEEE Transactions on Image Processing 20*, 12 (Dec 2011), 3376–3382.

[37] CHEN, Q., AND WAN, Y. A new framework for image impulse noise removal with postprocessing. In *2014 IEEE Visual Communications and Image Processing Conference* (Dec 2014), pp. 442–445.

[38] CHEN, S., AND ZHANG, D. Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 34*, 4 (Aug 2004), 1907–1916.

[39] COIFMAN, R. R., AND WICKERHAUSER, M. V. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory 38*, 2 (March 1992), 713–718.

[40] DANA H. BALLARD, C. M. B. *Computer Vision*, 1st ed. Prentice Hall INC., Englewood Cliffs, New Jersey 07632, May 1982.

[41] DAUGMAN, J. G. Two-dimensional spectral analysis of cortical receptive field profiles . *Vision Research 20*, 10 (1980), 847–856.

[42] DEBOEVERIE, F., VEELAERT, P., AND PHILIPS, W. Image segmentation with adaptive region growing based on a polynomial surface model. *Journal of Electronic Imaging 22*, 4 (2013). cited By (since 1996)0.

[43] DIWAKAR, M., SONAM, AND KUMAR, M. Ct image denoising based on complex wavelet transform using local adaptive thresholding and bilateral filtering. In *Proceedings of the Third International Symposium on Women in Computing and Informatics* (New York, NY, USA, 2015), WCI '15, ACM, pp. 297–302.

[44] DO, M. N., AND VETTERLI, M. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing 14*, 12 (Dec 2005), 2091–2106.

[45] DONOHO, D., JOHNSTONE, I., AND JOHNSTONE, I. M. Ideal spatial adaptation by wavelet shrinkage. *Biometrika 81* (1993), 425–455.

[46] DONOHO, D. L. Wedgelets: Nearly minimax estimation of edges. *The Annals of Statistics 27*, 3 (1999), 859–897.

[47] DONOHO, D. L., AND JOHNSTONE, J. M. Ideal spatial adaptation by wavelet shrinkage. *Biometrika 81*, 3 (1994), 425–455.

[48] DUNN, J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics 3*, 3 (1973), 32–57.

[49] DURAND, F., AND DORSEY, J. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph. 21*, 3 (July 2002), 257–266.

[50] EBERHART, R., AND KENNEDY, J. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on* (1995), pp. 39–43.

[51] ENGELBRECHT, A. P. *Computational Intelligence: An Introduction*, 2nd ed. Wiley Publishing, 2007.

[52] FENG, J., JIAO, L., ZHANG, X., GONG, M., AND SUN, T. Robust non-local fuzzy c-means algorithm with edge preservation for {SAR} image segmentation . *Signal Processing 93*, 2 (2013), 487–499.

[53] FOROUZANFAR, M., FORGHANI, N., AND TESHNEHLAB, M. Parameter optimization of improved fuzzy c-means clustering algorithm for brain {MR} image segmentation . *Engineering Applications of Artificial Intelligence 23*, 2 (2010), 160–168.

[54] FU, K., AND MUI, J. A survey on image segmentation. *Pattern Recognition 13*, 1 (1981), 3–16.

[55] GAO, L., WANG, G., AND LIU, J. Image denoising based on edge detection and prethresholding Wiener filtering of multi-wavelets fusion. *International Journal of Wavelets, Multiresolution and Information Processing 13*, 05 (2015), 1550031.

[56] GHARIEB, R., GENDY, G., ABDELFATTAH, A., AND SELIM, H. Adaptive local data and membership based KL divergence incorporating C-means algorithm for fuzzy image segmentation. *Applied Soft Computing 59* (2017), 143–152.

[57] GHARIEB, R. R., GENDY, G., AND ABDELFATTAH, A. A robust local data and membership information based fcm algorithm for noisy image segmentation. In *2016 12th International Computer Engineering Conference (ICENCO)* (Dec 2016), pp. 93–98.

[58] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[59] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing*, 3rd ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2008.

[60] GUO, F. F., WANG, X. X., AND SHEN, J. Adaptive fuzzy c-means algorithm based on local noise detecting for image segmentation. *IET Image Processing 10*, 4 (2016), 272–279.

[61] GUOFENG, J., WEI, Z., ZHENGWEI, Y., ZHIYONG, H., YUANJIA, S., DONGDONG, W., AND GAN, T. Image segmentation of thermal waving inspection based on particle swarm optimization fuzzy clustering algorithm. *Measurement Science Review 12*, 6 (2012), 296–301. cited By 9.

[62] HAO, M., ZHANG, H., SHI, W., AND DENG, K. Unsupervised change detection using fuzzy c-means and MRF from remotely sensed images. *Remote Sensing Letters 4*, 12 (2013), 1185–1194.

[63] HARRABI, R., AND BRAIEK, E. Color image segmentation using multi-level thresholding approach and data fusion techniques: Application in the breast cancer cells images. *Eurasip Journal on Image and Video Processing 2012*, 1 (2012). cited By (since 1996)2.

[64] HATHAWAY, R., BEZDEK, J., AND HU, Y. Generalized fuzzy c-means clustering strategies using lp norm distances. *Fuzzy Systems, IEEE Transactions on 8*, 5 (Oct 2000), 576–582.

[65] HUANG, Z.-K., LI, Z.-H., HUANG, H., LI, Z.-B., AND HOU, L.-Y. Comparison of different image denoising algorithms for Chinese calligraphy images . *Neurocomputing 188* (2016), 102–112. Advanced Intelligent Comput-

ing Methodologies and ApplicationsSelected papers from the Tenth International Conference on Intelligent Computing (ICIC 2014).

[66] HUMMEL, R. A. Readings in computer vision: Issues, problems, principles, and paradigms. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987, ch. Representations Based on Zero-crossings in Scale-space, pp. 753–758.

[67] IBRAHEEM, M. R., AND ELMOGY, M. *Automated Segmentation and Classification of Hepatocellular Carcinoma Using Fuzzy C-Means and SVM*. Springer International Publishing, Cham, 2016, pp. 193–210.

[68] JAIN, A. K., AND DUBES, R. C. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[69] JAIN, P., AND TYAGI, V. A survey of edge-preserving image denoising methods. *Information Systems Frontiers 18*, 1 (2016), 159–170.

[70] KENNEDY, J., AND EBERHART, R. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on* (1995), vol. 4, pp. 1942–1948 vol.4.

[71] KIM, S., MIN, B., OH, W., AND LEE, J. Medical image enhancement algorithm using edge-based denoising and adaptive histogram stretching. *International Journal of Bio-Science and Bio-Technology 5*, 5 (2013), 25–38. cited By 2.

[72] KLETTE, R. *Concise Computer Vision*, 1st ed. Springer., London, UK, 2014.

[73] KOENDERINK, J. J. The structure of images. *Biological Cybernetics 50*, 5 (1984), 363–370.

[74] KRINIDIS, S., AND CHATZIS, V. A robust fuzzy local information c-means clustering algorithm. *Image Processing, IEEE Transactions on 19*, 5 (May 2010), 1328–1337.

[75] KUMAR, R., SUBODH, S., AND RAJEEV, S. A fourth order PDE based fuzzy c- means approach for segmentation of microscopic biopsy images in presence

of Poisson noise for cancer detection. *Computer Methods and Programs in Biomedicine 146* (2017), 59–68.

[76] KUNDU, M. K., CHOWDHURY, M., AND BANERJEE, M. Interactive image retrieval using M-band wavelet, earth mover's distance and fuzzy relevance feedback. *International Journal of Machine Learning and Cybernetics 3*, 4 (Dec 2012), 285–296.

[77] LABATE, D., LIM, W.-Q., KUTYNIOK, G., AND WEISS, G. Sparse multidimensional representation using shearlets, 2005.

[78] LEZORAY, O., AND CARDOT, H. Segmentation of color images by clustering 2d histogram and merging regions.

[79] LI, F., AND QIN, J. Robust fuzzy local information and $l_p$lp-norm distance-based image segmentation method. *IET Image Processing 11*, 4 (2017), 217–226.

[80] LI, Q., MA, G., HUO, G., ZHOU, Y., AND SHENG, H. New segmentation method of side-scan sonar image based on edge detection in NSCT domain. *Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument 34*, 8 (2013), 1795–1801. cited By (since 1996)0.

[81] LIM, W. Q. The discrete shearlet transform: A new directional transform and compactly supported shearlet frames. *IEEE Transactions on Image Processing 19*, 5 (May 2010), 1166–1180.

[82] LIU, G., WANG, A., AND ZHAO, Y. An efficient image segmentation method based on fuzzy particle swarm optimization and markov random field model. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on* (2011), pp. 1–4.

[83] LIU, J., AND QIAO, S. A image segmentation algorithm based on differential evolution particle swarm optimization fuzzy c-means clustering. *Computer Science and Information Systems 12*, 2 (2015), 873–893. cited By 0.

[84] MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical*

Statistics and Probability, Volume 1: Statistics (Berkeley, Calif., 1967), University of California Press, pp. 281–297.

[85] MAJEED, I., AND ARIF, O. Non-linear eigenspace visual object tracking . *Engineering Applications of Artificial Intelligence 55* (2016), 363–374.

[86] MALLAT, S. G. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell. 11*, 7 (July 1989), 674–693.

[87] MANDAL, S., KUMARI, S., BHAVSAR, A., AND SAO, A. K. Multi-scale image denoising while preserving edges in sparse domain. In *2016 6th European Workshop on Visual Information Processing (EUVIP)* (Oct 2016), pp. 1–6.

[88] MARTIN, D., FOWLKES, C., TAL, D., AND MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision* (July 2001), vol. 2, pp. 416–423.

[89] MARTIN, V., AND THONNAT, M. Scene reconstruction, pose estimation and tracking. In *Scene Reconstruction, Pose Estimation and Tracking*, R. Stolkin, Ed. I-Tech Education and Publishing.

[90] MEENA PRAKASH, R., AND SHANTHA SELVA KUMARI, R. Spatial Fuzzy C Means and Expectation Maximization Algorithms with Bias Correction for Segmentation of MR Brain Images. *Journal of Medical Systems 41*, 1 (2016), 15.

[91] MEKHMOUKH, A., AND MOKRANI, K. Improved Fuzzy C-Means based Particle Swarm Optimization (PSO) initialization and outlier rejection with level set methods for {MR} brain image segmentation . *Computer Methods and Programs in Biomedicine 122*, 2 (2015), 266–281.

[92] MEYER, F. G., AND COIFMAN, R. R. Brushlets: A Tool for Directional Image Analysis and Image Compression . *Applied and Computational Harmonic Analysis 4*, 2 (1997), 147–187.

[93] MIN WANG, ZHEN LI, X. D., AND LI, W. An Image Denoising Method with Enhancement of the Directional Features Based on Wavelet and SVD Transforms. *Mathematical Problems in Engineering 2015*, 469350 (2015), 9.

[94] MIRGHASEMI, S., R. R. Z. M. A new modification of fuzzy c-means via particle swarm optimization for noisy image segmentation. In *Proceedings of the Second Australasian Conference, ACALCI 2016, Canberra, ACT, Australia* (2016), pp. 147–159.

[95] MIRGHASEMI, S., RAYUDU, R., AND ZHANG, M. A heuristic solution for noisy image segmentation using particle swarm optimization and fuzzy clustering. In *Proceedings of the 7th International Joint Conference on Computational Intelligence* (2015), pp. 17–27.

[96] MIRGHASEMI, S., SADOGHI YAZDI, H., AND LOTFIZAD, M. A target-based color space for sea target detection. *Applied Intelligence 36*, 4 (2012), 960–978.

[97] PARIS, S., KORNPROBST, P., TUMBLIN, J., AND DURAND, F. Bilateral Filtering: Theory and Applications. *Foundations and Trends® in Computer Graphics and Vision 4*, 1 (2009), 1–73.

[98] PARVEEN, AND SINGH, A. Detection of brain tumor in mri images, using combination of fuzzy c-means and svm. In *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)* (Feb 2015), pp. 98–102.

[99] PAVLIDIS, T., AND LIOW, Y.-T. Integrating region growing and edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 12*, 3 (Mar 1990), 225–233.

[100] PENNEC, E. L., AND MALLAT, S. Sparse geometric image representations with bandelets. *IEEE Transactions on Image Processing 14*, 4 (April 2005), 423–438.

[101] PERONA, P., AND MALIK, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 7 (Jul 1990), 629–639.

[102] PODERICO, M., PARRILLI, S., POGGI, G., AND VERDOLIVA, L. Sigmoid shrinkage for bm3d denoising algorithm. In *2010 IEEE International Workshop on Multimedia Signal Processing* (Oct 2010), pp. 423–426.

[103] PORIKLI, F. Constant time o(1) bilateral filtering. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (June 2008), pp. 1–8.

[104] QIU, C., XIAO, J., YU, L., HAN, L., AND IQBAL, M. N. A modified interval type-2 fuzzy C-means algorithm with application in {MR} image segmentation . *Pattern Recognition Letters 34*, 12 (2013), 1329–1338.

[105] RAHMAN, M. M., BHATTACHARYA, P., AND DESAI, B. C. A framework for medical image retrieval using machine learning and statistical similarity matching techniques with relevance feedback. *IEEE Transactions on Information Technology in Biomedicine 11*, 1 (Jan 2007), 58–69.

[106] RAJABY, E., AHADI, S., AND AGHAEINIA, H. Robust color image segmentation using fuzzy c-means with weighted hue and intensity. *Digital Signal Processing 51* (2016), 170–183.

[107] RASTISLAV LUKAC, K. N. P. *Color Image Processing: Methods and Applications*, 3rd ed. CRC Press, 2006.

[108] RICHARD O. DUDA, P. E. H. *Pattern Classification and Scene Analysis*, 1st ed. Wiley, Stanford Research Institute, Menlo Park, California, 1973.

[109] ROBERTS, L. G. Machine perception of three-dimensional solids, May 1965.

[110] SAHOO, P., SOLTANI, S., AND WONG, A. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing 41*, 2 (1988), 233–260.

[111] SCHMALZ, M. S., AND RITTER, G. X. Region segmentation techniques for object-based image compression: a review, 2004.

[112] SHANG, R., WEN, A., LIU, Y., JIAO, L., AND STOLKIN, R. Synthetic aperture radar image change detection based on improved bilateral filtering and fuzzy C mean. *Journal of Applied Remote Sensing 10*, 4 (2016), 046017.

[113] SHEN, S., SANDHAM, W., GRANAT, M., AND STERR, A. Mri fuzzy segmentation of brain tissue using neighborhood attraction with neural-network optimization. *IEEE Transactions on Information Technology in Biomedicine 9*, 3 (Sept 2005), 459–467.

[114] SHI, J., AND MALIK, J. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 8 (2000), 888–905.

[115] SHI, Y., YANG, X., AND GUO, Y. Translation invariant directional framelet transform combined with gabor filters for image denoising. *IEEE Transactions on Image Processing 23*, 1 (Jan 2014), 44–55.

[116] SIMA, H., LIU, L., AND GUO, P. Color Image Segmentation Based on Blocks Clustering and Region Growing. In *Neural Information Processing*, B.-L. Lu, L. Zhang, and J. Kwok, Eds., vol. 7064 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 459–466.

[117] SUGIMOTO, K., AND KAMATA, S. I. Compressive bilateral filtering. *IEEE Transactions on Image Processing 24*, 11 (Nov 2015), 3357–3369.

[118] SZILAGYI, L., BENYO, Z., SZILAGYI, S., AND ADAM, H. Mr brain image segmentation using an enhanced fuzzy c-means algorithm. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE* (Sept 2003), vol. 1, pp. 724–726 Vol.1.

[119] TIAN, X., JIAO, L., YI, L., GUO, K., AND ZHANG, X. The image segmentation based on optimized spatial feature of superpixel . *Journal of Visual Communication and Image Representation 26* (2015), 146–160.

[120] TIAN, X., JIAO, L., AND ZHANG, X. A clustering algorithm with optimized multiscale spatial texture information: application to SAR image segmentation. *International Journal of Remote Sensing 34*, 4 (2013), 1111–1126.

[121] TIAN, X., JIAO, L., AND ZHANG, X. A clustering algorithm with optimized multiscale spatial texture information: application to SAR image segmentation. *International Journal of Remote Sensing 34*, 4 (2013), 1111–1126.

[122] Tomasi, C., and Manduchi, R. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)* (Jan 1998), pp. 839–846.

[123] Tomasi, C., and Manduchi, R. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)* (Jan 1998), pp. 839–846.

[124] Tran, D., Wu, Z., and Tran, V. Fast Generalized Fuzzy C-means Using Particle Swarm Optimization for Image Segmentation. In *Neural Information Processing*, C. Loo, K. Yap, K. Wong, A. Teoh, and K. Huang, Eds., vol. 8835 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, pp. 263–270.

[125] Tushir, M., and Smriti, S. A new Kernelized hybrid c-mean clustering model with optimized parameters. *Applied Soft Computing 10*, 2 (2010), 381–389.

[126] Unnikrishnan, R., Pantofaru, C., and Hebert, M. Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 6 (June 2007), 929–944.

[127] Verma, H., Agrawal, R., and Sharan, A. An improved intuitionistic fuzzy c-means clustering algorithm incorporating local information for brain image segmentation . *Applied Soft Computing 46* (2016), 543–557.

[128] Wang, W., and Zhang, Y. On fuzzy cluster validity indices . *Fuzzy Sets and Systems 158*, 19 (2007), 2095–2117. Theme: Data Analysis.

[129] Wang, X., Lin, X., and Yuan, Z. An edge sensing fuzzy local information c-means clustering algorithm for image segmentation. vol. 8589 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, pp. 230–240.

[130] Wang, X.-Y., and Bu, J. A fast and robust image segmentation using {FCM} with spatial information. *Digital Signal Processing 20*, 4 (2010), 1173–1182.

[131] Wang, X.-Y., Wang, Q.-Y., Yang, H.-Y., and Bu, J. Color image segmentation using automatic pixel classification with support vector machine. *Neurocomput. 74*, 18 (Nov. 2011), 3898–3911.

[132] Wang, X.-Y., Wang, Q.-Y., Yang, H.-Y., and Bu, J. Color image segmentation using automatic pixel classification with support vector machine. *Neurocomput. 74*, 18 (Nov. 2011), 3898–3911.

[133] Wang, X.-Y., Zhang, X.-J., Yang, H.-Y., and Bu, J. A pixel-based color image segmentation using support vector machine and fuzzy c-means. *Neural Netw. 33* (Sept. 2012), 148–159.

[134] Watson, A. B. The cortex transform: Rapid computation of simulated neural images. *Computer Vision, Graphics, and Image Processing 39*, 3 (1987), 311–327.

[135] Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin 1*, 6 (1945), 80–83.

[136] Witkin, A. P. Scale-space filtering. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 2* (San Francisco, CA, USA, 1983), IJCAI'83, Morgan Kaufmann Publishers Inc., pp. 1019–1022.

[137] Wu, Y., Zhang, P., Li, M., Zhang, Q., Wang, F., and Jia, L. {SAR} image multiclass segmentation using a multiscale and multidirection triplet Markov fields model in nonsubsampled contourlet transform domain . *Information Fusion 14*, 4 (2013), 441–449.

[138] Xue, B. *Particle Swarm Optimisation for Feature Selection.* PhD thesis, School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand, 2014.

[139] Yamany, S. M., Farag, A. A., and Hsu, S.-Y. A fuzzy hyperspectral classifier for automatic target recognition (ATR) systems. *Pattern Recognition Letters 20*, 11 (1999), 1431–1438.

[140] YITZHAKY, Y., AND PELI, E. A method for objective edge detection evaluation and detector parameter selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 8 (Aug 2003), 1027–1033.

[141] YUM, P. Research methodology, 2009.

[142] ZHANG, H., FRITTS, J. E., AND GOLDMAN, S. A. Image segmentation evaluation: A survey of unsupervised methods . *Computer Vision and Image Understanding 110*, 2 (2008), 260–280.

[143] ZHANG, L., AND XU, L. A new method for choosing the regularization parameter of rof total variation image denoising. In *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Aug 2016), vol. 01, pp. 373–377.

[144] ZHANG, M., AND GUNTURK, B. K. Multiresolution bilateral filtering for image denoising. *IEEE Transactions on Image Processing 17*, 12 (Dec 2008), 2324–2333.

[145] ZHANG, Q., HUANG, C., LI, C., YANG, L., AND WANG, W. Ultrasound image segmentation based on multi-scale fuzzy c-means and particle swarm optimization. In *Information Science and Control Engineering 2012 (ICISCE 2012), IET International Conference on* (Dec 2012), pp. 1–5.

[146] ZHANG, X., SUN, Y., WANG, G., GUO, Q., ZHANG, C., AND CHEN, B. Improved fuzzy clustering algorithm with non-local information for image segmentation. *Multimedia Tools and Applications 76*, 6 (Mar 2017), 7869–7895.

[147] ZHANG, X., WANG, G., SU, Q., GUO, Q., ZHANG, C., AND CHEN, B. An improved fuzzy algorithm for image segmentation using peak detection, spatial information and reallocation. *Soft Computing 21*, 8 (Apr 2017), 2165–2173.

[148] ZHANG, X.-P., AND DESAI, M. D. Adaptive denoising based on sure risk. *IEEE Signal Processing Letters 5*, 10 (Oct 1998), 265–267.

[149] ZHAO, F., FAN, J., AND LIU, H. Optimal-selection-based suppressed fuzzy c-means clustering algorithm with self-tuning non local spatial information for

image segmentation. *Expert Systems with Applications 41*, 9 (2014), 4083–4093. cited By (since 1996)0.

[150] ZHOU, X.-C., SHEN, Q.-T., AND LIU, L.-M. New two-dimensional fuzzy C-means clustering algorithm for image segmentation. *Journal of Central South University of Technology (English Edition) 15*, 6 (2008), 882–887. cited By 0.