Summer 2008

Medical Image Modeling and Processing

Ramu Pedada
Old Dominion University

MEDICAL IMAGE MODELING AND PROCESSING

by

Ramu Pedada
B.Tech, May 2006, Jawaharlal Nehru Technological University, India

A Thesis Submitted to the Faculty of Old Dominion University
in Partial Fulfillment of the Requirement for the Degree
of

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 2008

Approved by:

Yuzhong Shen (Director)

Jiang Li (Member)

Lee A. Belfore II (Member)

ABSTRACT

MEDICAL IMAGE MODELING AND PROCESSING

Ramu Pedada
Old Dominion University, August 2008
Director: Dr. Yuzhong Shen

During the last few decades of the twentieth century, medical imaging has been playing a prominent role in many fields of biomedical research and clinical practice. Image modalities such as x-rays, computed tomography (CT), and magnetic resonance images (MRI) have all been valuable additions to the radiologist's arsenal of imaging tools. Medical images assure quality diagnosis and patient safety by gathering valuable information without invading the human body. Apart from clinical diagnosis, medical images are used as tools for education where they are used for training individuals before operating on a patient. Many medical educators turn to simulation based training systems to provide meaningful learning experiences. Modeling and simulation of medical images is an essential part of such training systems. In this thesis work, we developed two algorithms for medical applications. The first part of he thesis deals with wound debridement simulation, and it automatically generates he wound textures used in the simulation. Previously, the images were generated by artists manually using image editing tools such as Photoshop. The proposed algorithm is a procedural method that has two passes. In the first pass, we use the Perlin noise function to generate noise at every pixel location of the wound image. In the second pass, the noise is confined to the wound area by elliptical shape composition. By automatically generating texture images, each execution of the wound debridement simulation produces new wound conditions thereby

greatly enhancing the training scenarios and improving the training outcome. The second part of the thesis deals with MRI image denoising, and it optimizes the parameter values associated with the denoising algorithm. Many image denoising algorithms have been proposed for MRI images. Noise removal is accomplished at the expense of blurred subtle features in the image. Therefore, there is always a trade-off between noise removal and structure information preservation. In the proposed algorithm, we define several cost functions to assess the outcome of denoising methods and utilize the Strength Pareto Evolutionary Algorithm (SPEA2) to simultaneously optimize those cost functions by modifying parameters associated with the denoising algorithm. The proposed optimization procedure is illustrated by optimizing the parameters of image denoising based on block matching and 3D collaborative filtering. Using the proposed optimization procedure, instead of a single solution, we obtain a set of solutions so that the designer can choose the best trade-offs.

ACKNOWLEGEMENTS

First of all, I want to thank my advisor Dr. Yuzhong Shen, for his guidance during my research and study at Old Dominion University. His energy and enthusiasm in research motivated all his advisees, including me. Throughout my research period, he provided encouragement, sound advice, and lots of good ideas. Without him this work would not have been possible. I would also like to thank Dr. Jiang Li and Dr. Lee A. Belfore II for serving on my thesis committee. I want to thank all the members of my research group for their valuable suggestions. I especially want to thank my twin brother Laxman Pedada, who has been my inspiration since the start of my college education. Lastly, and most importantly, I wish to thank my parents for their support and love.

This thesis is dedicated to my parents

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

. LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Medical imaging has become a vital component in many fields of bio-medical research and clinical practise that allows scientists and physicians to gain life-saving information without invading the human body [1]. Medical imaging applications occur throughout the track of clinical events ranging from clinical diagnostics to surgical procedures. With the discovery of x-rays by Wilhelm Roentgen in 1895, the first medical images were produced using x-rays. For over a century, x-ray projection radiography has been most widely used diagnostic imaging modality. In 1972, G.N. Hounsfield and A.M. Cormack invented computed tomography [2], which is considered to be the greatest innovation in the field of radiology since the discovery of x-rays. Today, computed tomography (CT) is one of the most important methods of radiological diagnosis. It delivers non-superimposed, cross-sectional images of the body that can show smaller contrast differences than conventional x-ray images. Magnetic resonance imaging (MRI) emerged in the 1980s and is in many ways the most informative medical imaging methodology. This imaging technology can provide information unavailable from other imaging methods.

Medical images are not only used by physicians for diagnostics but also as a tool for education. In 1989, the National Library of Medicine (NLM) started the Visible Human Project [3] to build a digital image library of volumetric data representing

This thesis follows IEEE journal style

complete, normal adult human male and female cadavers. The digital image library consists of computerized tomography (CT), magnetic resonance (MR), and cyrosection images of a representative. The visible human project was an effort to create a detailed data set of cross-sectional photographs of the human body in order to facilitate anatomy visualization applications. The visible human project data sets were designed to serve as a common reference point for the study of human anatomy and have been used by over 800 licenses in 27 countries. The data sets are being applied to a wide range of educational, diagnostic, treatment planning, virtual reality, artistic, mathematical and industrial uses. Medical images are used for training individuals before operating on a real patient. In 1999, the Institute of Medicine issued a report estimating that as many as 98,000 people die from errors in hospital settings each year. Many medical educators turn to simulation based training systems to provide meaningful learning experiences. In many high risk domains, simulators are often used to allow trainees to acquire and practise skills in a safe and controlled environment. Modeling and simulation of medical images is a key part of simulator based training methods. Most of the simulators use a wide range of simulated medical image modalities based on the type of simulator being used for training.

1.2 PROPOSED WORK

We propose the development of two algorithms for medical applications. The first proposed algorithm deals with wound debridement simulation, and it automatically generates the wound textures in the simulation. The second proposed algorithm deals with MRI image denoising, and it optimizes the parameter values used in an image

denoising algorithm. We briefly describe the two algorithms as follows.

1.2.1 Procedural Modeling of Wound Textures

Virtual reality based training systems have been developed for various surgical fields. Realistic simulation of scenes with anatomically correct organ models and textures is the main purpose of such systems. Simulator based training methods allow trainees to acquire and practice skills in a safe and controlled environment. One main advantage of surgical simulation is that trainees can acquire much of the fundamental knowledge needed to perform basic activities on their own without the need for constant supervision by an instructor. This also assures patient safety. The surgical wound debridement simulator was developed at Old Dominion University [4] to provide training of surgical wound debridement in a virtual environment. In the wound debridement simulator, wound cleanliness is modelled by a series of textured images, simulating a variety of debris around the contaminated wound. Previously, the wound texture images were created by artists manually with the help of image editing tools such as Photoshop, which takes substantial amount of time, and the created texture images once loaded could not be changed dynamically during the simulation. This limits the training outcome.

In this thesis, we propose a procedural method to automatically generate the wound texture images used in the wound debridement simulator. This is accomplished by texture mapping. Texture mapping is a powerful tool in the field of imaging that can add detail, surface texture, and color to a computer generated graphic model. Noise functions play a key role in procedural modeling. The proposed method is a two pass procedure. In the first pass of the proposed method, we use the Perlin

noise function to simulate noise at every pixel location of a clean wound image. In the second pass the noise is confined to the wound area by elliptical shape composition. By automatically generating the wound textures, each execution of the wound debridement simulation produces new wound conditions, thereby greatly enhancing the range of scenarios that the trainee can experience and improving the training outcome.

1.2.2 Parameter Optimization for Image Denoising Based on Block Matching and 3D Collaborative Filtering

MRI images play an important role in identifying various pathologies. The raw data sets collected during the acquisition process are generally contaminated by noise. Imperfect instruments, problems with the data acquisition process, and interfering natural phenomena can all degrade the data of interest. Thus, denoising is often a necessary first step to be taken before the images are analyzed. For quality analysis, the image has to be restored in the presence of noise. Various restoration techniques have been proposed ranging from basic mean filtering to wavelet domain filters. The restoration process is accomplished at the expense of blurred subtle features in the images. Therefore, there is always a trade-off between noise removal and structure information preservation for many denoising methods. In order to effectively remove noise and to preserve structure information, it is crucial to tune denoising methods to obtain the best trade-off. In this thesis, we propose an optimization procedure that can produce a set of optimal solutions which enable the designer to make the best trade-offs. The proposed algorithm is illustrated using image denoising based on block matching and 3D collaborative filtering.

Image denoising using block matching and 3D collaborative filtering is a recently developed denoising algorithm based on enhanced sparse representation in the transform domain. 3D collaborative filtering enhances the sparsity so that the noise can be well attenuated by shrinkage of transform coefficients [5]. Block matching is used to stack 2D blocks of a given image that are similar to a reference block to form a 3D array. The major steps involved in collaborative filtering are applying a 3D transform to the formed groups followed by shrinkage of transform coefficients and inverse wiener filtering. The main disadvantage of this denoising algorithm is that a lot of interesting structure information is lost during the process of denoising. Structure information should be preserved for effective image analysis.

In the proposed research, we define several cost functions to assess the quality of noise removal and the quantity of structure information preserved in the denoised image and utilize the Strength Pareto Evolutionary Algorithm (SPEA2) to simultaneously optimize those cost functions by modifying parameters associated with denoising methods. Experimental results showed that the proposed optimization algorithm can significantly improve denoising methods in terms of noise removal and structure information preservation.

1.3 THESIS ORGANIZATION

The remainder of this thesis is organized as follows. Chapter II presents the proposed algorithm to model the texture images for the wound debridement simulator using procedural modeling. The results of the proposed algorithm are compared with that of the results obtained with Photoshop. Chapter III describes the MRI denoising

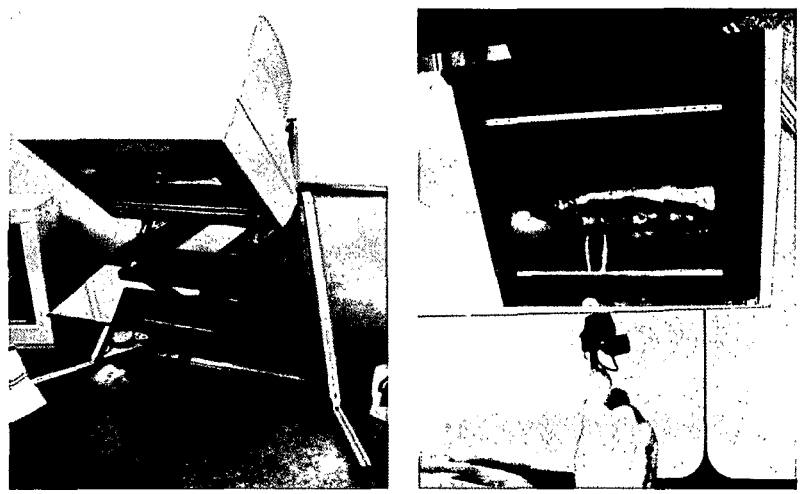
algorithm used in this thesis based on block matching and 3D collaborative filtering and presents the proposed algorithm for optimizing the results of image denoising methods. The proposed method is used to optimize the results of image denoising using block matching and 3D collaborative filtering. Chapter IV concludes the thesis and includes discussion of possible future work.

CHAPTER 2

PROCEDURAL MODELING OF WOUND TEXTURES

2.1 INTRODUCTION

Surgical simulation is a valuable addition to traditional teaching methods in the field of medical education. It helps trainees improve their skills prior to operating on a patient. The wound debridement simulator [4] was developed at Old Dominion University to provide training for surgical wound debridement in a virtual environment. Wound debridement refers to the process of removing necrotic, devitalized, or contaminated tissue and/or foreign material to promote healing. The simulator currently uses a shallow laceration wound to the thigh with embedded shards of glass, dirt, and debris, as shown in Fig. 1. Trainees are instructed to remove the glass shards with forceps, scrub with a brush, rinse with saline solution, and remove devitalized tissue. Wound cleanliness is modeled by a series of textured images, simulating a variety of debris on the contaminated wound. At any time, there is only one active image laid on the polygonal mesh that models the wound geometry by texture mapping. Texture mapping is a standard and powerful rendering technique used in modern computer graphics, and it enhances realism and details in a scene with only a modest increase in computational complexity. Previously, these textures were generated manually by artists using image editing tools such as Photoshop. The creation and editing of these texture images in Photoshop requires careful determination of parameters used by Photoshop manually and takes substantial time to achieve satisfactory results. In



(a) (b)

FIG. 1: Wound Debridement Simulator. (a) Configuration. (b) Reach-in display.

addition, once the textured images are created and loaded into the wound debridement simulation, they cannot be changed dynamically during the simulation, thus seriously limiting the training scenarios that the trainee can experience.

In this chapter, an algorithm for automatically generating the textured images used in the wound debridement simulator is presented. The proposed method is a two-pass procedure. In the first pass, noise is generated based on spectral synthesis to simulate the dirt and debris on the wound. In the second pass, the noise locations are computed based on elliptical shape composition. Automatic image generation significantly reduces the amount of work and time needed by the artist. Furthermore, each execution of automatic image generation produces a set of visually different but statistically identical images. In the wound debridement simulation, a new set of textured images amounts to a different wound condition, thus each execution of the wound debridement simulation produces a new wound instance, greatly enhancing

the range of scenarios that the trainee can experience and improving the training outcome.

2.2 PROCEDURAL MODELING

Procedural modeling has been widely used in the history of computer graphics. It has become a vital component in creating realistic computer-generated images, commercial rendering and animation systems. Procedural techniques are code segments or algorithms that specify some characteristics of a computer-generated model or effect. For example, a procedural texture for a marble surface does not use a scanned image to define the color of the marble. Instead, it uses algorithms and mathematical functions to determine the color values for the marble surface. Procedural modeling techniques can be used to create complex models of nature like water, smoke, steam, fire, etc. The three important features of procedural modeling are abstraction, parametric control, and flexibility [6]. These are briefly described below.

Abstraction: Instead of explicitly specifying and storing all the complex details of a scene, in procedural modeling, they are abstracted into a function or algorithm and that algorithm is evaluated when and where it is needed. This saves storage, as the details are no longer explicitly specified but are implicit in the procedure. This shifts the time requirements for detail specifications from programmer to computer which allows us to create inherent multiresolution models and textures that can be evaluated to the required resolution.

Parametric Control: This allows users to assign to a parameter a meaningful concept. This provides amplification of user efforts as a few parameters yield a large

amount of detail. Parametric control reduces the burden of low level control and specification on the user.

Flexibility: Procedural models are flexible. The designer of the procedure can capture the essence of the object, phenomenon, or motion without being constrained by the complex laws of Physics. Procedural techniques allow the user to include any desired amount of physical accuracy in the model. The designer may produce a wide range of effects, from accurately simulating natural laws to purely artistic effects.

2.3 NOISE MODELS

An important component in procedural modeling is the noise function, which generates pseudorandom numbers to be used as the base for more complicated patterns. Some of the most popular implementations of noise functions are Lattice Noise, Value Noise, Gradient Noise, Perlin Noise, Value Gradient Noise, etc. The properties of an ideal noise function are as follows:

1. Noise is a repeatable pseudorandom function of its inputs.
2. Noise has a known range depending on the required application.
3. Noise is band-limited with a maximum frequency of 1.
4. Noise does not exhibit regular patterns. Such pseudorandom functions are always periodic, but the period can be made very long; therefore, the periodicity is not conspicuous.
5. Noise is stationary, i.e. its statistical character should be translationally invariant.
6. Noise is isotropic, i.e. its statistical character should be rotationally invariant.

2.3.1 Perlin Noise

Among different noise functions, Perlin noise has significant importance as it can generate a wide variety of complex procedural textures with modest computational cost. Perlin noise is a lattice noise of the gradient variety. Perlin noise produces more realistic results by interpolating random noise. Perlin noise first generates a set of pseudorandom gradient vectors at all lattice points and then interpolates the gradient vectors to generate the desired noise. A rectangular lattice structure is used to generate the noise. The final Perlin noise function is then generated by adding all the smoothed interpolated functions with various frequencies and amplitudes. The values of persistence can be adjusted to specify the amplitude at each frequency of random noise function. Persistence is a constant multiplier that determines how quickly the amplitudes diminish for each successive iteration in a Perlin-noise function. A higher value of persistence produces "rougher" Perlin noise. The frequency and amplitude of a Perlin noise component are defined as follows,

$$\text{frequency} = 2^i; \text{amplitude} = \text{persistence}^i; \tag{1}$$

where i is the ith noise function being added. Each successive noise function that is being added is known as an octave. The number of octaves to be added depend on the user requirement. The step by step procedure for generating one dimensional Perlin noise is described below along with the pseudo code.

Step 1. Generating Random Noise:

The first step is to generate 1-dimensional random noise with floating point numbers in the form of -1 and $+1$. This can be accomplished by a pseudo random number

generator. The pseudorandom number generator functions [7] available in most standard computer math libraries generate random numbers between the range of 0 and 1, but Perlin noise matrix elements vary between -1 and +1. To generate random numbers within a desired range, the following mathematical relation is used.

$$R_N = ((R_H - R_L) \times R) + R_L, \tag{2}$$

where R_N is a floating point random number whose range is $R_L \leq R_N \leq R_H$. R is a random number varying between 0 and 1. R_L and R_H are the lower and upper limits of the desired range. For the Perlin noise matrix, R_L and R_H are -1 and +1 respectively.

Step 2. Interpolation:

The noise values between grid points are computed by interpolating the noise values at the grid points. Several interpolation techniques such as linear interpolation [8], cosine interpolation, and cubic interpolation can be used. Of all these, cubic interpolation generates smoother results when compared to cosine and linear interpolations [9].

Step 3. Smoothed Noise:

After interpolation, we also need to smooth the output of the noise function to make it bandlimited. Smoothing is done by averaging the current value with its neighbor. The pseudo code for 1-dimensional smoothing function is given below:

SmoothNoise(x)

return $Noise(x)/2 + Noise(x-1)/4 + Noise(x+1)/4$

end

where Noise(x) is the interpolated random noise function.

Step 4. Combining Octaves

To generate the 1- dimensional Perlin noise the interpolated noise functions have to be added together. Octaves define the number of iterations over the noise function for a particular point. For each iteration, the frequency is doubled, and the amplitude is given by Eq. (1). The pseudo code for 1-dimensional Perlin noise is given below:

PerlinNoise(float x)

total = 0;

loop i from 0 to $Number\_of\_Octaves$;

frequency = 2^i;

amplitude = $persistence^i$;

total = total + $SmoothNoise_i(x * frequency) * amplitude$;

end of i loop

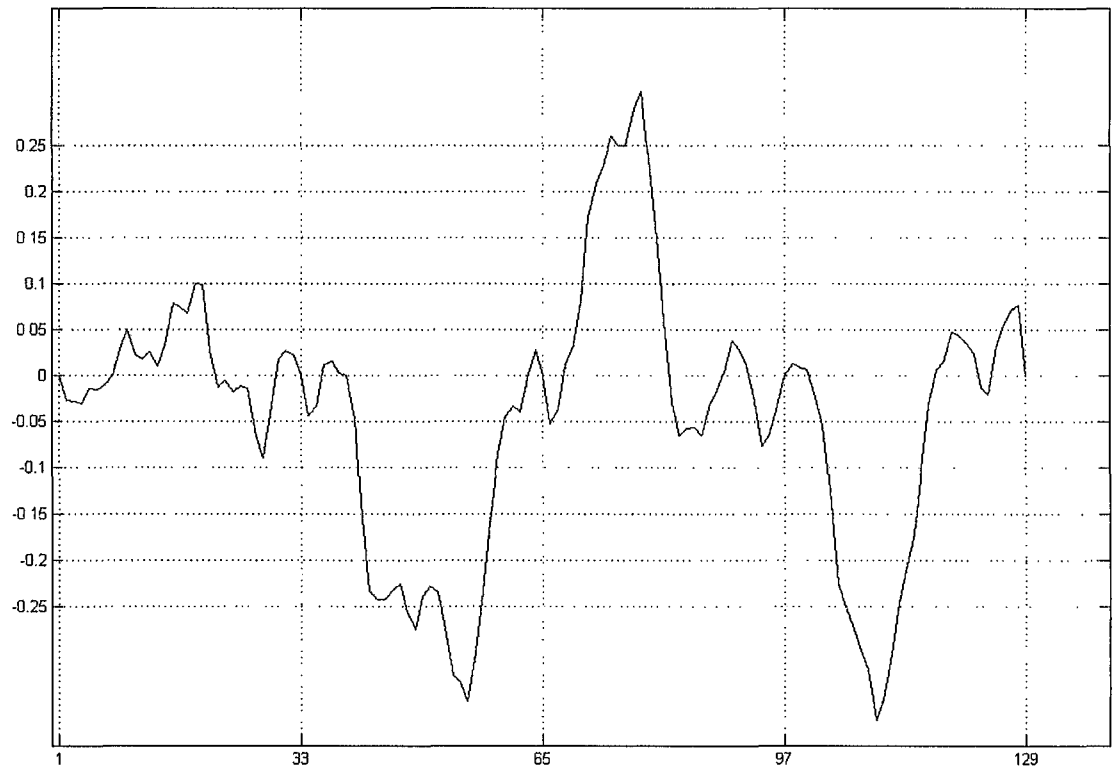
return total

end

Fig. 2 shows the 1D Perlin noise generated using the method described above. The noise is generated using 5 octaves with a persistence value of 1/2.

2.4 PROPOSED METHODS

The proposed method is a procedural technique since the algorithm possesses all the characteristics described in section II.2. The texture images for wound debridement are generated in two passes. In the first pass, Perlin noise is generated using the procedure described in section II.3.1. In the second pass, the noise is confined around the wound area using elliptical shape composition.



(a)

FIG. 2: 1D Perlin noise generated with 5 octaves.

2.4.1 2-D Noise Generation Based on Spectral Synthesis

Since we deal with textured images, the noise function should be 2-dimensional. The method used to generate 1-D Perlin noise can be extended to generate 2-D Perlin noise. A basic 2-D Perlin noise function is generated by defining a noise matrix $N(x, y)$ and assigning pseudorandom numbers used in Eq. (2) to each element of $N(x, y)$, with values varying from -1 to $+1$, where x and y represent rows and columns of the noise matrix. To generate band-limited noise, low-pass filtering is applied to $N(x, y)$ to

yield a stochastic function $Ns(x, y)$ that is smoother than white noise as follows,

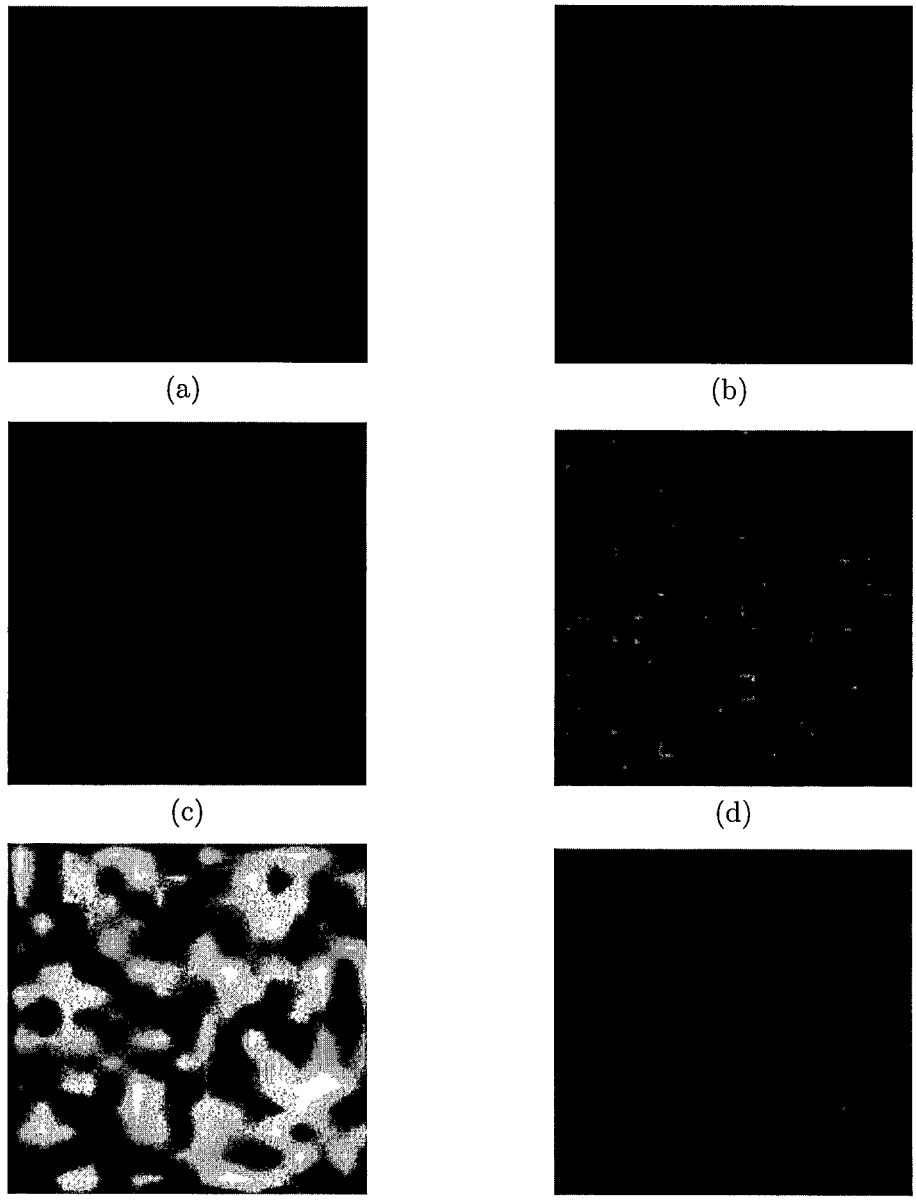
$$\begin{aligned} N_s(x,y) &= \frac{1}{16}N(x-1,y-1) + \frac{1}{16}N(x+1,y-1) + \\ &\frac{1}{16}N(x-1,y+1) + \frac{1}{16}N(x+1,y+1) + \frac{1}{8}N(x-1,y) + \\ &\frac{1}{8}N(x+1,y) + \frac{1}{8}N(x,y-1) + \frac{1}{8}N(x,y+1) + \frac{1}{4}N(x,y). \end{aligned} \tag{3}$$

To create even richer stochastic content, spectral synthesis [6] is utilized. Analogous to Fourier synthesis, spectral synthesis can generate complex textures by using a combination of sinusoidal component functions of differing frequencies, amplitudes, and phases. In spectral synthesis, the so-called turbulence function blends multiple layers of smoothed noise function with different frequencies. It creates natural looking features out of smoothed noise. The commonly used Perlin's turbulence noise function [10] is a spectral synthesis method that uses the noise function defined in Eq. (3) as the primitive.

$$N_T(x,y) = \sum_{f=1}^{N} \frac{1}{f} N_s(2^f x, 2^f y), \tag{4}$$

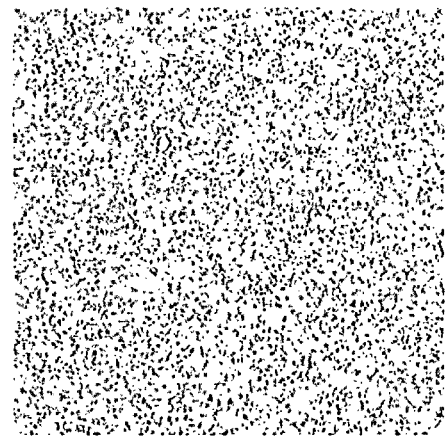
where 2^f is the component frequency, N is the number of frequency components, and N_T denotes Perlin's turbulence function. It can be seen that the amplitude of each frequency component 2^f is proportional to $1/f$. Equations (3) and (4) apply to grayscale images. The Perlin's turbulence noise component functions with different frequencies and amplitudes are shown in Fig. 3(a) to Fig. 3(e). Eq. (4) is used to combine Figs. 3(a) to 3(e), and the resultant image is shown in Fig. 3(f).

To create color noise, noise matrix N is replaced with a randomly distributed color pattern. Fig. 4 shows some textured images generated using spectral synthesis for the wound debridement simulator. In Fig. 4(a) and Fig. 4(b), the color noise density can be controlled by varying the number of frequency components in Eq. (4).



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

FIG. 3: Spectral summation. (a), (b), (c), (d), and (e) are various textures generated using varied amplitude and frequency. (f) Result of spectral summation.



(a) (b)

FIG. 4: Color texture images generated by spectral synthesis

The texture images created in Fig. 4 are then combined with a clean wound image to form a dirty wound textured image using an alpha-transparency model [11] as follows

$$I_D = \alpha I_C + (1 - \alpha) I_N, \tag{5}$$

where I_D is the dirty wound, I_C is the clean wound image, and I_N is the noise image. The value of α is chosen empirically; for the current textures, a value of 0.5 is used. Fig. 5(a) shows the clean wound image, and Fig. 5(b) shows the combined image.

2.4.2 Elliptical Noise Shape Composition

The noise generated in the previous section produces a new value for each pixel in the texture map, which is not desired since the dirt and debris are usually located in the wound area. To limit the noise to the wound area, ellipses are used as the fundamental element to define the noise shape in the proposed method because the ellipse's shape can be easily changed by adjusting its center location, major axis,

(a) (b)

FIG. 5: Synthesized image. (a) Clean image and (b) Synthesized noisy image.

minor axis, and orientation. The standard ellipse with its center at the origin and its axes aligned with the coordinate axes is shown in Fig. 6(a). A point P with coordinates (x, y) on the ellipse can be represented as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{6}$$

where a and b are the major and minor axes of the ellipse. Fig. 6(b) shows an arbitrary ellipse with its center at (C_x, C_y) and an intersection angle of θ between its major axis and x axis. A point $Q(x, y)$ on this arbitrary ellipse can be transformed to a point $Q'(x', y')$ on the standard ellipse with a rotation of $-\theta$ followed by a translation of $(-C_x, -C_y)$. Expressed in homogeneous coordinates [12], the transformation can be written as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -C_x \\ 0 & 1 & -C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{7}$$

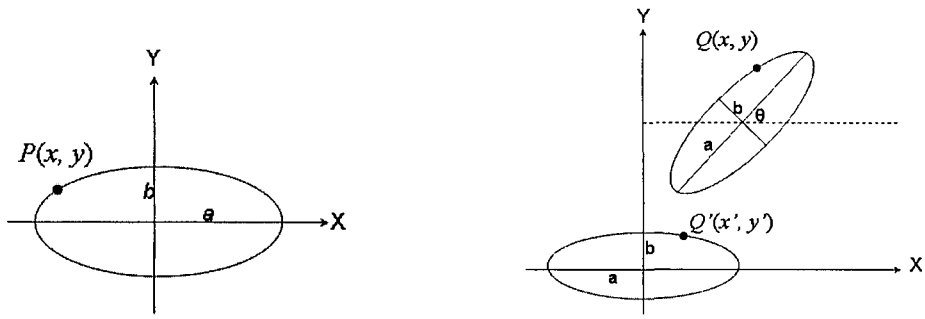


FIG. 6: Ellipse representations. (a) The basic ellipse with it center at origin and its major and minor axes aligned with coordinate axes. (b) An arbitrary ellipse with its center at (C_x, C_y) and an intersection angle of θ between its major axis and x axis.

This results in

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} (x - C_x)\cos\theta + (y - C_y)\sin\theta \\ -(x - C_x)\sin\theta + (y - C_y)\cos\theta \end{bmatrix}. \tag{8}$$

Substituting (8) into the standard ellipse from Eq. (6), we obtain the equation for the arbitrary ellipse as follows,

$$\begin{aligned} &\frac{1}{a^2}((x - C_x)\cos\theta + (y - C_y)\sin\theta)^2 + \\ &\frac{1}{b^2}(-(x - C_x)\sin\theta + (y - C_y)\cos\theta)^2 = 1. \end{aligned} \tag{9}$$

The noise shape definition consists of two steps. A set of large ellipses is first generated, yielding a rough approximation of the wound shape. Then a new set of fine (small) ellipses around each large ellipse created in the first step are generated with the restriction that their centers are located close to the perimeter of the large ellipse. Correlated pseudorandom numbers are used to control the location, size, shape, and density of the ellipses generated. Fig. 7 shows an irregular shape formed by a set of correlated ellipses. As is common in procedural approaches, a large set of parameters are needed to achieve realistic effects.

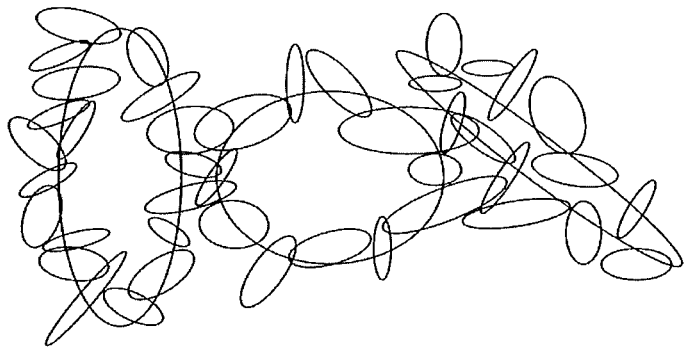


FIG. 7: Irregular shape composed of correlated varying ellipses

To reduce computational complexity, the bounding boxes for all ellipses are computed. For the general ellipse described by Eq. (9), the maximum axial distances from the ellipse is center to a point inside the ellipse can be computed as follows.

$$\begin{aligned} x_B &= \max(\text{abs}(a\cos\theta - b\sin\theta), \text{abs}(a\cos\theta + b\sin\theta)) \\ y_B &= \max(\text{abs}(a\sin\theta + b\cos\theta), \text{abs}(a\sin\theta - b\cos\theta)), \end{aligned} \tag{10}$$

where abs$(\cdot)$ is the absolute value operator and max$(\cdot)$ is the maximum operator. Thus, the ellipse bounding box can be computed as follows,

$$C_x - x_B \leq x \leq C_x + x_B, C_y - y_B \leq y \leq C_y + y_B. \tag{11}$$

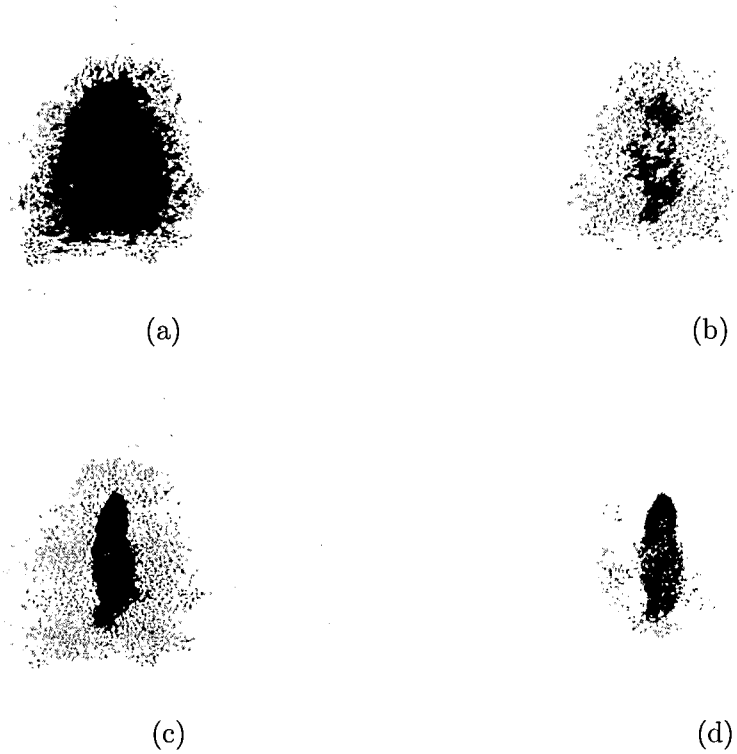
Based on Eq. (11), only pixels within the bounding box of an ellipse rather than all the pixels in the textured image are checked if they are inside that ellipse, reducing computational cost significantly.

The entire set of ellipses is used as a stencil mask to apply the noise generated in Section II.3.1 to the base clean wound image to form various dirty image textures. To obtain even better visual effects, low-pass filtering is performed in order to eliminate high frequency noise components. The noise components are the artifacts formed at

the edges due to the mask applied. Due to the pseudorandom numbers utilized in noise generation and elliptical noise shape composition, each execution of the wound debridement simulation produces new wound conditions, greatly enhancing the range of scenarios that the trainee can experience.

2.5 EXPERIMENTAL RESULTS

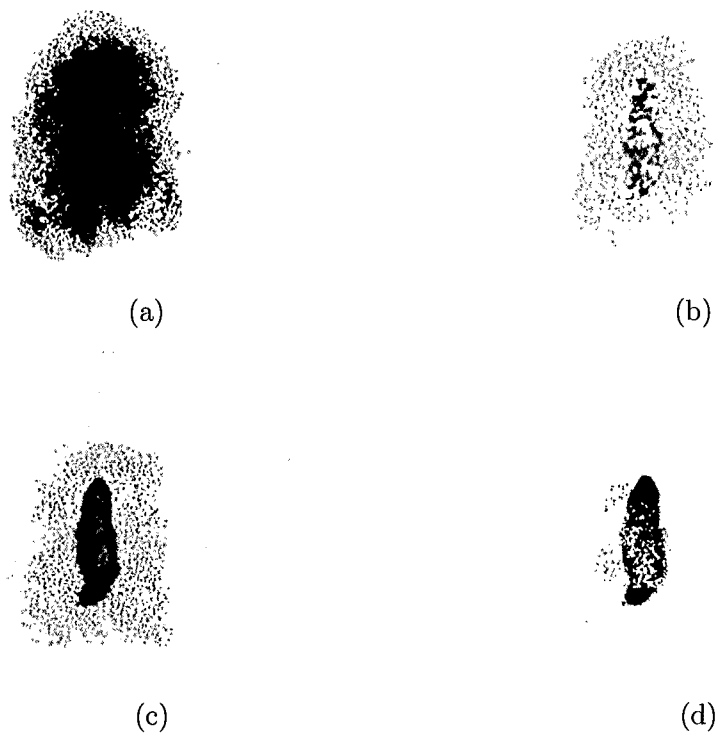
To test the effectiveness of the proposed approach, the algorithm is used to generate many different wound texture images, and the results are compared with the wound textures generated manually using Photoshop. Fig. 8 shows the texture images generated using Photoshop. Textures in Fig. 9 are synthesized using the proposed algorithm and are similar to the manually generated ones in Fig. 8. A set of different synthesized textured images with varied geometry, noise color, and noise density is shown in Fig. 10, enhancing the range of scenarios that the wound debridement simulator can depict. Figures 11, 12, and 13 show the 3-dimensional, wounded thigh created using the textured images in Figures 8, 9, and 10, respectively. From these results, it can be seen that the proposed method can produce results similar to ones previously generated manually through a tedious process. In addition, the proposed method can easily produce results that are different from the ones generated manually, thus increasing the variety of conditions the wound debridement simulator can model.



(a) (b)

(c) (d)

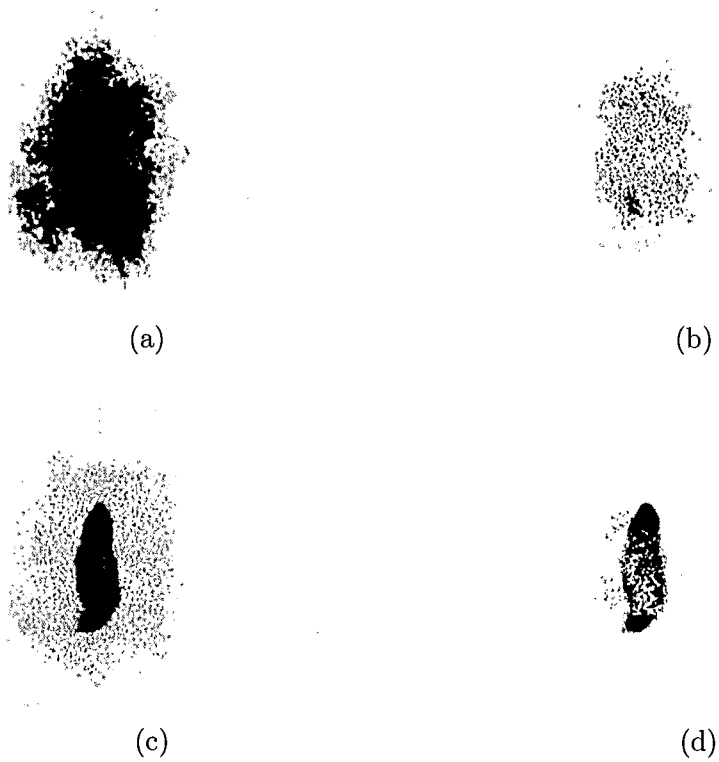
FIG. 8: Texture images generated by Photoshop.



(a) (b)

(c) (d)

FIG. 9: Texture images generated by proposed method.



(a) (b)

(c) (d)

FIG. 10: Additional texture images generated by proposed method.

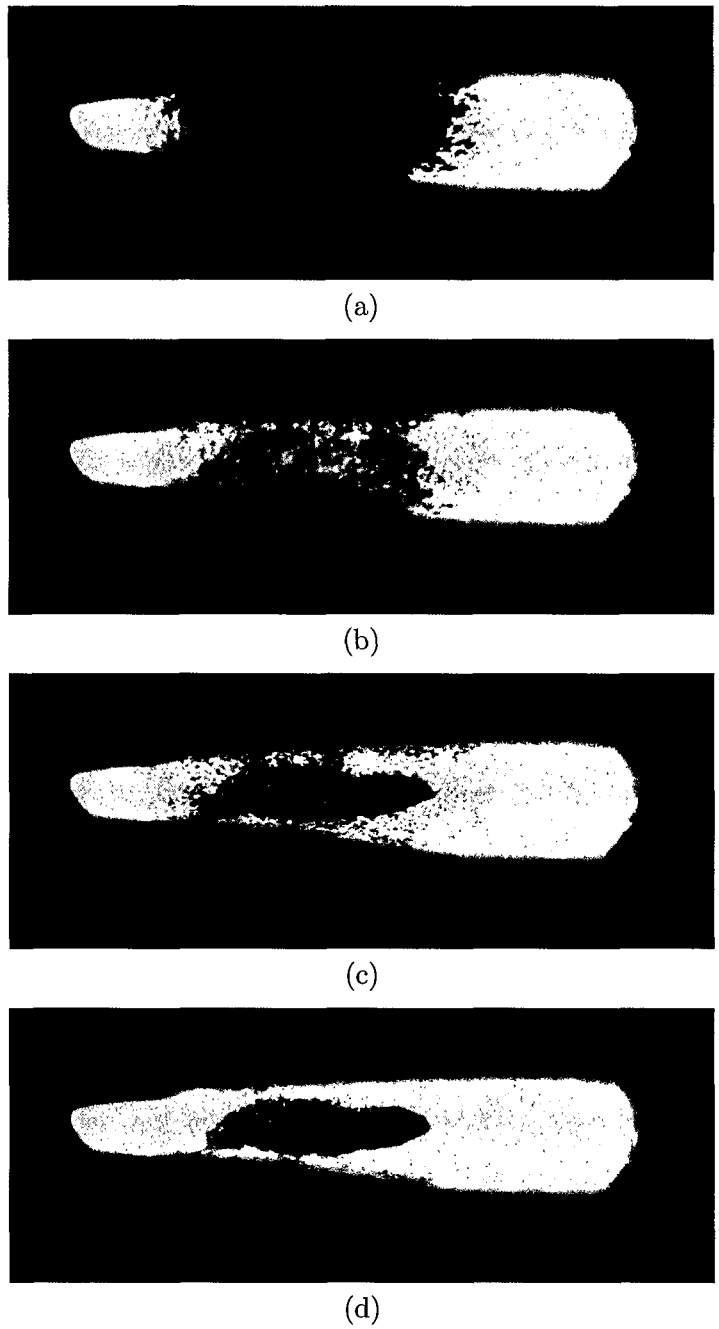
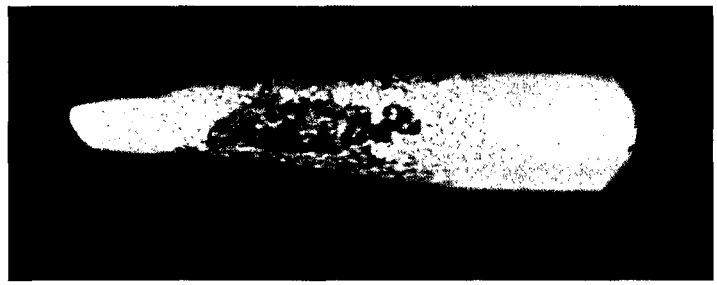
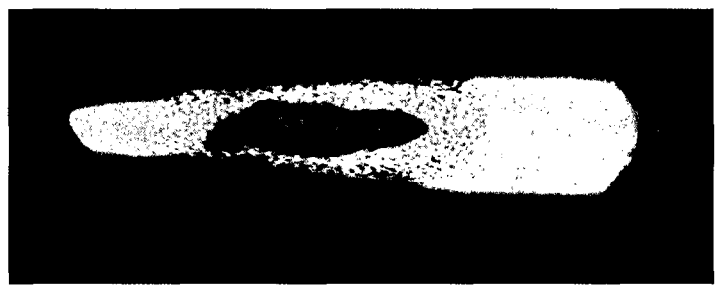


FIG. 11: 3-dimensional wounded thigh generated with texture images in Fig. 8, which are manually generated using Photoshop.

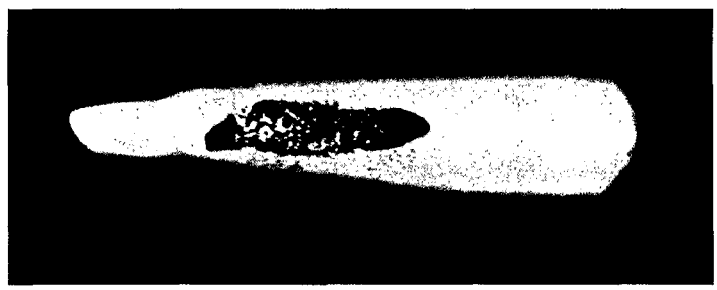
(a)



(b)



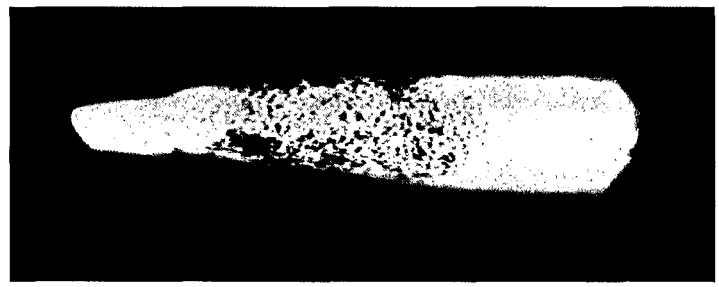
(c)



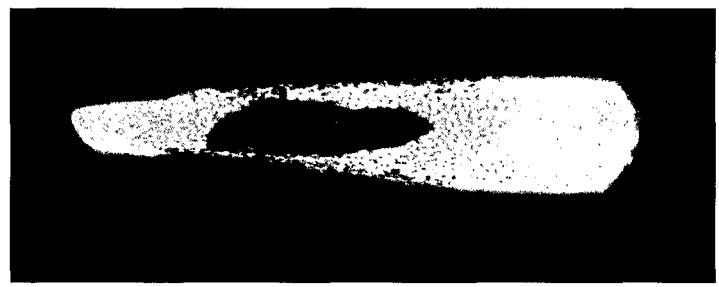
(d)

FIG. 12: 3-dimensional wounded thigh generated with texture images of Fig. 9, which are generated automatically by the proposed algorithm.

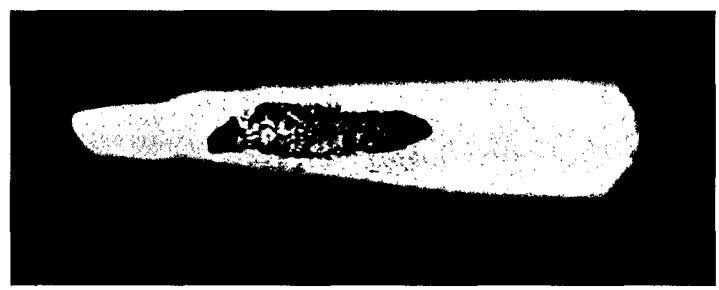
(a)



(b)



(c)



(d)

FIG. 13: 3-dimensional wounded thigh generated with texture images of Fig. 10, which are generated automatically by the proposed algorithm.

2.6 CONCLUSION

In this chapter, an algorithm to automatically generate the wound texture images used in the wound debridement simulator is proposed. The proposed method is a two-pass procedure. In the first pass, noise is generated based on spectral synthesis to simulate the dirt and debris on the wound. In the second pass, the noise locations are computed based on elliptical shape composition. Due to the pseudorandom numbers utilized in noise generation and elliptical noise shape composition, each execution of the wound debridement simulation produces new wound conditions, greatly enhancing the range of scenarios that the trainee can experience.

CHAPTER 3

PARAMETER OPTIMIZATION FOR IMAGE DENOISING BASED ON BLOCK MATCHING AND 3D COLLABORATIVE FILTERING

3.1 INTRODUCTION

Magnetic resonance imaging is a medical imaging technique primarily used in radiology to visualize the structure of a body. In clinical practice, MRI is used to distinguish pathologic tissue from normal tissue. Clinical MRI images are generally corrupted by random noise during acquisition with blurred subtle structure features [13]. Many denoising methods have been proposed to remove noise from corrupted images at the expense of distorted structure features [14]. Therefore, there is always a trade-off between noise removal and structure information preservation for many denoising methods. In order to effectively remove noise and to preserve structure information, it is crucial to tune denoising methods that can obtain the best trade-off. In this thesis, we propose an optimization procedure that can produce a set of optimal solutions that enable the designer to make the best trade-offs. The proposed algorithm is illustrated by applying the proposed optimization procedure to enhance the results of image denoising based on block matching and 3D collaborative filtering.

Image denoising using block matching and 3D collaborative filtering is a recently developed denoising algorithm based on an enhanced sparse representation in the transform domain. Block matching is used to stack 2D blocks of a given image that are similar to a reference block. The matched blocks are stacked together to form a

3D array. Collaborative filtering is a special procedure used to deal with these 3D groups. The major steps involved in collaborative filtering are applying a 3D transform to the formed groups followed by shrinkage of transform coefficients and inverse wiener filtering. The reason for using 3D filtering is that 2D transforms cannot achieve good sparsity for all kinds of images. Using 3D arrays instead of 2D blocks enhances the sparsity so that the noise can be well attenuated by shrinkage of transform coefficients. The main disadvantage of this denoising algorithm is that a lot of interesting structure information is lost during the process of denoising. Structure information should be preserved for effective image analysis. In the proposed research, we define several cost functions to assess the quality of noise removal and the quantity of structure information preserved in the denoised image and utilize the Strength Pareto Evolutionary Algorithm (SPEA2) to simultaneously optimize those cost functions by modifying parameters associated with denoising methods. Experimental results showed that the proposed optimization algorithm can significantly improve denoising methods in terms of noise removal and structure information preservation.

3.2 OVERVIEW OF IMAGE DENOISING

3.2.1 Noise Sources in MRI Images

The magnetic resonance signal is an electromotive force induced in a coil by a rotating magnetic moment of nuclear spins. The signal level must dominate the level of noise to produce clinically useful MR images. Noise in the MR image originates from the patient or object to be imaged or at the receiver section where the signal is processed [15]. In the receiver, equipment noise is generated in the preamplifier

and the connection between the preamplifier and the receiver RF coil. Thermal noise is produced by the stochastic motion of free electrons in the RF coil, which is a conductor. The stochastic motion is caused by the ohmic loss in the RF coil. The eddy current losses in the patient are inductively coupled to the RF coil, which is the another cause for the stochastic motion. The resistance induced in the receiver section due to eddy currents in the patient contributes a significant amount of noise.

3.2.2 Restoration Model

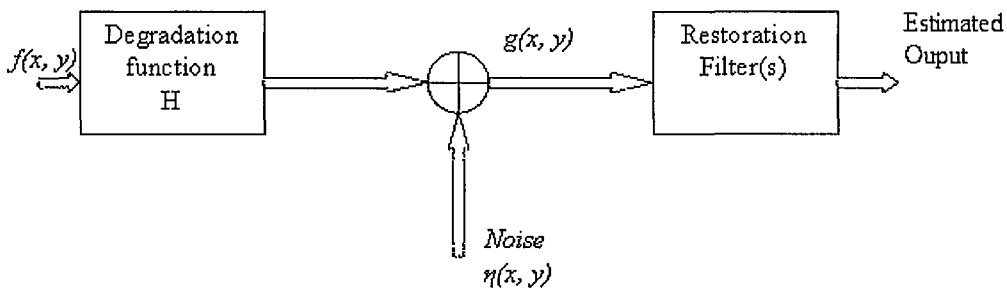
Image restoration is an essential preprocessing step for many image analysis applications. It is concerned with the removal or reduction of degradations in an image that are introduced due to several factors during image acquisition. Denoising refers to the process of recovering an image that has been degraded by noise with the help of prior knowledge of the degrading phenomenon. Consider an image, $f(x, y)$, that is degraded by the presence of noise, $\eta(x, y)$. Let $g(x, y)$ be the degraded image. Using this information, a denoising filter should obtain an estimate, $\hat{f}(x, y)$, which has to be as close as possible to $f(x, y)$. The degraded image in the spatial domain is given by

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y), \tag{12}$$

where $h(x, y)$ is the spatial representation of the degradation function, and $*$ represents the convolution process. The frequency domain representation of the above equation is given as follows.

$$G(u, v) = H(u, v)F(u, v) + N(u, v), \tag{13}$$

where the capitalized terms are the Fourier transforms of the corresponding terms in Eq. (12). The block diagram of the image restoration process is shown in Fig. 14. The prime objective of the restoration process is to make the estimated output as close as possible to the input.



(a)

FIG. 14: Block diagram of image restoration process.

3.2.3 Types of Noise Functions

Noise in an image is randomly distributed over the entire image. Therefore, noise can be treated as a random variable characterized by a probability density function (PDF). For most denoising algorithms the PDF serves as a priori information about the type of noise [16]. In this section, a brief overview of most common noise PDFs is presented.

Gaussian Noise

The Gaussian noise model is used frequently in many denoising algorithms, because of its mathematical tractability. The PDF of a Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}, \tag{14}$$

where z represents gray level, μ is the mean of z, σ is the standard deviation, and σ^2 is the variance.

Rayleigh Noise

The PDF of Rayleigh noise is characterized by the following equation.

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & for \quad z \geq a \\ 0 \quad for \quad z < a. \end{cases} \tag{15}$$

The expressions for μ and σ^2 are given as

$$\mu = a + \sqrt{\frac{\pi b}{4}}, \sigma^2 = \frac{b(4-\pi)}{4}. \tag{16}$$

Exponential Noise

The PDF of exponential noise is given as

$$p(z) = \begin{cases} ae^{-az} & for \quad z \geq 0 \\ 0 \quad for \quad z < 0. \end{cases} \tag{17}$$

where $a > 0$, $\mu = \frac{1}{a}$, and $\sigma^2 = \frac{1}{a^2}$.

Impulse Noise

The PDF of impulse noise is given as

$$p(z) = \begin{cases} P_a & for \quad z = a \\ P_b & for \quad z = b \\ 0 & otherwise \end{cases} \tag{18}$$

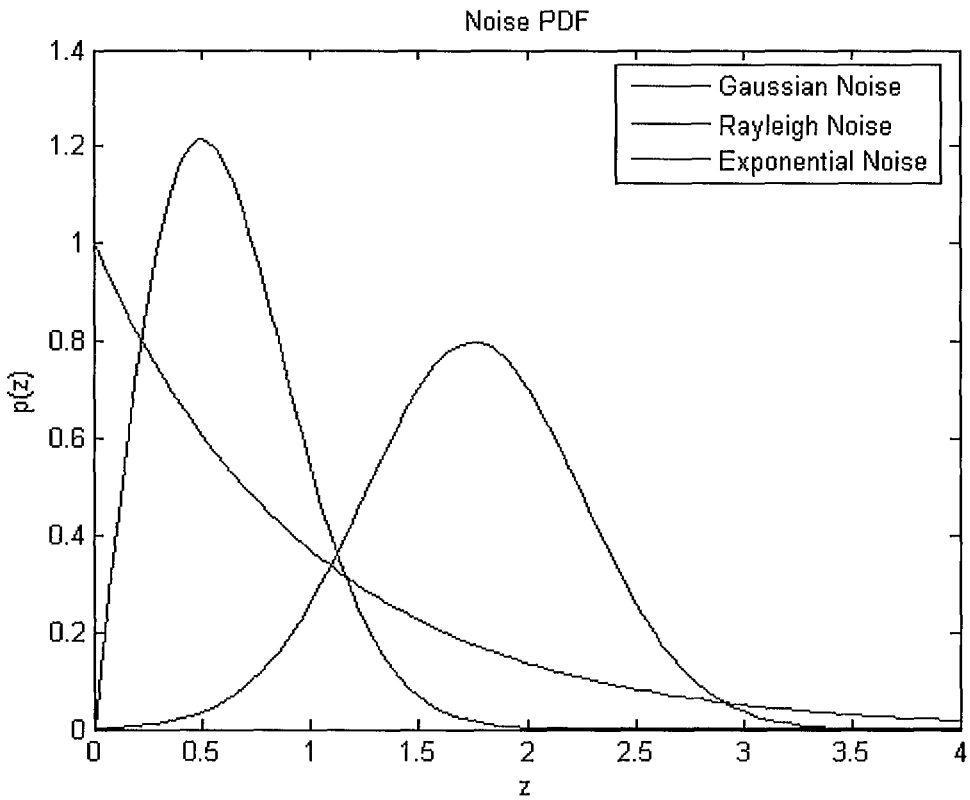
If either P_a or P_b is zero, the impulse noise is called unipolar. Salt and pepper noise is a special case of impulse noise. If neither of the probabilities are zero and are approximately equal, i.e. $P_a \simeq P_b$, then the impulse noise is referred to as salt and pepper noise. The type of denoising filter can be chosen based on the type of PDF associated with the noise in the concerned image. The plot of various noise probability density functions is shown in Fig. 15.

3.3 SPATIAL AND TRANSFORM DOMAIN DENOISING METHODS

Denoising methods can be classified as spatial and frequency domain methods. Denoising in the spatial domain is useful in situations where noise is additive. Spatial filters can be further divided as linear and nonlinear filters. In this section some common spatial denoising filters are discussed.

3.3.1 Linear Filters

For images corrupted with Gaussian noise, the mean filter is the optimal filter [16] in the sense of mean square error. However, linear filters tend to blur edges and other fine details in the image. Their performance is very poor in the presence of signal



(a)

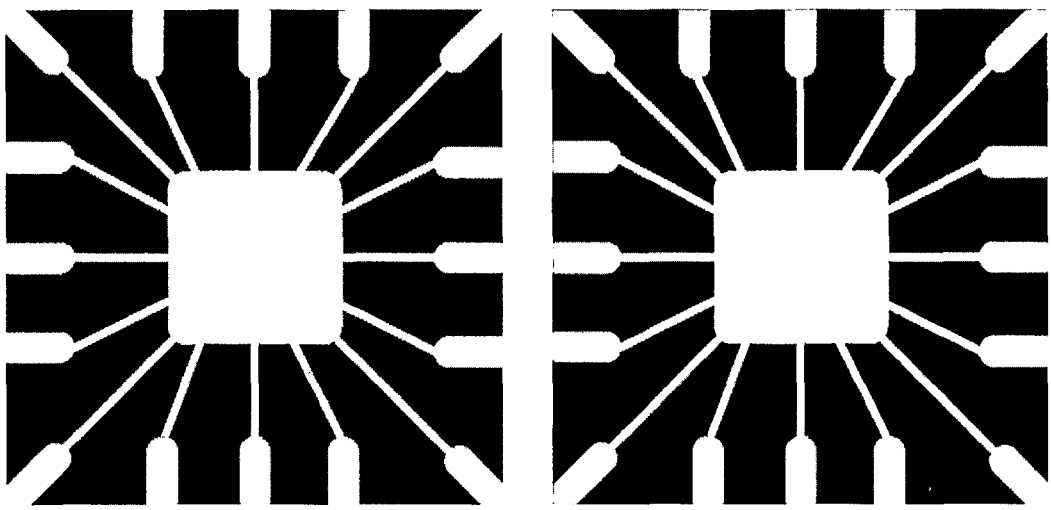
FIG. 15: Plot of noise PDFs.

dependent noise. Let S_{xy} be a set of coordinates used to represent a rectangular block of size m x n, centered at (x, y). The mean filter computes the average of the noisy image $g(x, y)$ in the area defined by S_{xy}. Noise is reduced as a result of smoothing. This is given as

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t). \tag{19}$$

Fig. 16(a) is a gray scale image corrupted by Gaussian noise, and the resulting image after mean filtering is shown in Fig. 16(b). The Geometric mean filter, harmonic mean filter, and contraharmonic mean filters are extended versions of the basic mean

filter. The Geometric mean filter achieves smoothing comparable to the arithmetic mean filter and, in the process of denoising less image details are lost. Harmonic mean filtering works well for Gaussian noise and salt noise, but it fails for pepper noise. Contraharmonic mean filters work well for eliminating the effects of salt-and-pepper noise.



(a) (b)

FIG. 16: Result of mean filter: (a) Image corrupted by Gaussian noise, (b) Output of mean filter.

3.3.2 Nonlinear Filters

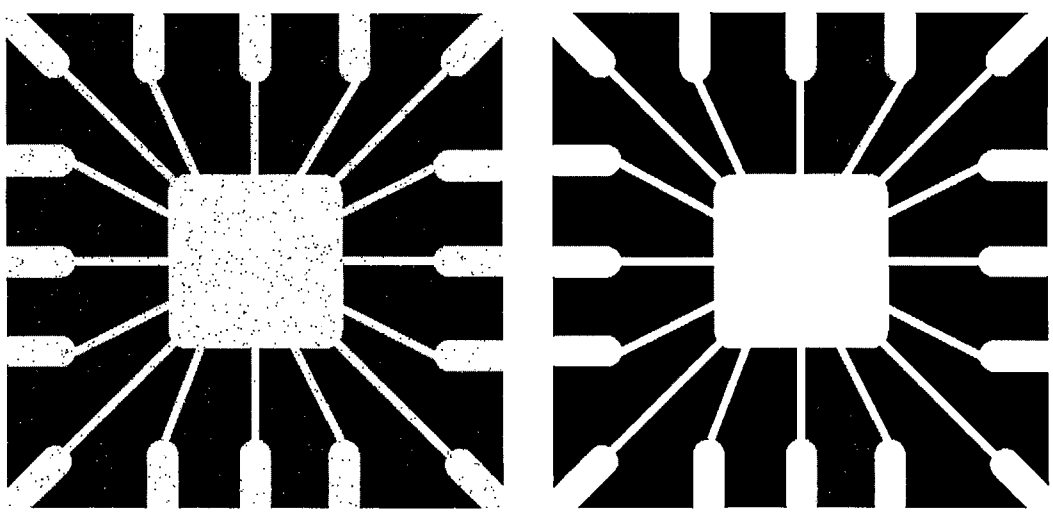
Linear filters often tend to blur meaningful high frequency components such as edges of an image. To overcome this nonlinear filters are used [17]. These are generally used to remove impulse noise from images. In this section, two important non linear filters are discussed.

Median Filters

For an observation window $\Omega = \{X_1, X_2, \ldots, X_N \in \mathbb{R}\}$, where N is the window size, the output of the median filter is defined as

$$X_{MED} = \arg\min_{\beta} \sum_{i=1}^{N} |X_i - \beta|. \tag{20}$$

From this definition it is easy to show that the median filter is selective when N is odd, i.e., it selects the input sample that minimizes the sum of the distances to all other samples in the window. If we order the input samples $X_1, X_2, \ldots, X_n$ such that $X_{(1)} < X_{(2)} < \cdots < X_{(N)}$, where $X_{(i)}$ denotes the ith smallest value in Ω, then $X_{MED} = X_{(\frac{N+1}{2})}$ when N is odd; when N is even, X_{MED} can be any value in the interval $[X_{(\frac{N}{2})}, X_{(\frac{N}{2}+1)}]$. The median is a robust estimator extensively used in statistics. Fig. 17(a) is a gray scale image corrupted by impulse noise, and the resulting image after median filtering is shown in Fig. 17(b).



(a) (b)

FIG. 17: Result of median filter: (a) Image corrupted by impulse noise, (b) Output of median filter.

Weighted Median Filters

Given a set of N non-negative real valued weights $\{W_1, W_2, \ldots, W_N\}$ and an observation window consisting of N samples, $\{X_1, X_2, \ldots, X_N\}$, the output of the weighted median filter [18] is defined as

$$X_{WM} = \arg\min_{\beta} \sum_{i=1}^{N} W_i |X_i - \beta|. \tag{21}$$

If the weights $W_1, W_2, \ldots, W_N$ are non-negative integers, Equation (21) is equivalent to the following:

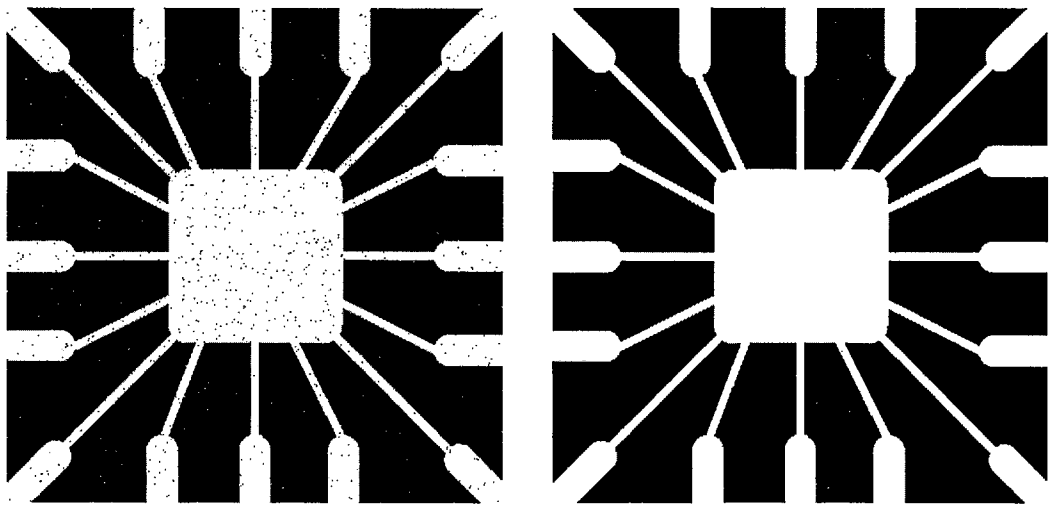
$$X_{WM} = \text{MEDIAN}(W_1 \diamond X_1, \ldots, W_N \diamond X_N), \tag{22}$$

where $\diamond$ is the replication operator. The positive weight constraint restricts the class of filters to smoothers. Recently developed procedures, however, take advantage of both positive and negative real weights to yield filters with a wide range of frequency responses [19]. The WM filter admitting negative weights is defined as

$$X_{WM} = \text{MEDIAN}(|W_1| \diamond sgn(W_1)X_1, \ldots, |W_N| \diamond sgn(W_n)X_N). \tag{23}$$

Fig. 18(a) is a gray scale image corrupted by impulse noise, and the resulting image after weighted median filtering is shown in Fig. 18(b).

Apart from a median and a weighted median filter there are other nonlinear filters such as a min-max filter, midpoint filter, and alpha-trimmed mean filter. Min-max filters are used in finding the brightest and darkest points in an image. The max filter is very useful in reducing pepper noise since pepper noise has very low values. The min filter is useful for reducing the effect of salt noise. The midpoint filter is best



(a) (b)

FIG. 18: Result of weighted median filter: (a) Image corrupted by impulse noise, (b) Output of weighted median filter.

for randomly distributed noise, like Gaussian or uniform noise. The alpha-trimmed mean filter is used when an image is corrupted by multiple types of noise such as a combination of salt-and-pepper and Gaussian noise. However, all these types of filters are applied to an image without regard how image characteristics vary from one point to another. Unlike these filters, adaptive nonlinear filters are capable of performing superior to that of the nonlinear filters discussed so far. However, improved filtering power increases filter complexity and computational time.

3.3.3 Denoising in Transform Domain

In transform domain denoising methods, a true signal is approximated by a linear combination of basis functions. By preserving high magnitude transform domain coefficients that convey most of the true signal and discarding the coefficients of

noise, the true signal can be estimated. 2D discrete cosine transform (DCT), discrete fourier transform (DFT), and discrete wavelet transform (DWT) are some of the basic transforms used to approximate the original signal in the form of basis functions. DCT is not effective in representing sharp transitions, whereas wavelets perform poorly for textures and smooth transitions. 2D transforms generally cannot approximate all kinds of images.

3.4 IMAGE DENOISING USING BLOCK MATCHING AND 3D COLLABORATIVE FILTERING

Before presenting the proposed optimization procedure for denoising methods, a brief review of a recently developed image denoising method is discussed in this section. The denoising algorithm is based on block matching and 3D collaborative filtering, and it was proposed by Kostadin Dabov [20]. In later sections, the proposed optimization procedure is illustrated by applying the optimization procedure to this denoising algorithm to enhance the results.

3.4.1 Algorithm Background

In the previous sections, some basic image denoising methods in the spatial and transform domains were discussed. All the methods described rely on some explicit or implicit assumptions about the true image in order to separate it properly from the random noise. Significantly, the transform domain methods assume that the true signal can be well approximated by a linear combination of basis functions. That is, the signal is sparsely represented in the transform domain, but the shrinkage of transform

domain coefficients is ineffective with sparse representation of an image. For example, 2D DCT cannot sparsely represent sharp transitions such as edges of an image. Generally, a 2D transform cannot achieve good sparsity because of the great variety of natural images. 2D orthogonal transforms can achieve sparse representation only for particular image patterns. To achieve good sparsity for any kind of natural image, 3D data arrays can be used instead of 2D fragments. In this novel image denoising strategy, the enhanced sparse representation in the transform domain is achieved by grouping similar 2D fragments of the image into 3D arrays called "groups." Collaborative filtering is a special procedure developed to deal with these 3D groups. 3D transformations are applied to the 3D groups followed by shrinkage of transform coefficients; then the inverse 3D transform is applied. Due to the similarity between the grouped blocks, the 3D transform can achieve a highly sparse representation of the true signal so that the noise can be well separated by shrinkage.

3.4.2 Grouping and Collaborative Filtering

Grouping is the concept of collecting similar d-dimensional fragments of a given signal to form a $d+1$ dimensional data structure. For an image, a 3D array is formed by stacking together similar 2D image fragments. A simpler and effective grouping strategy for images is by matching. Matching is a method for finding fragments similar to a given reference one [21]. This can be accomplished by pairwise testing the similarity between the reference fragment and other fragments located at other locations. A fragment is said to be matched with the reference fragment if the distance between these two is smaller than a specified threshold. The fragments satisfying this

condition are considered members of the group, and the reference fragment is considered a "centroid" for the group. The block-matching approach is used extensively for motion estimation in video compression techniques. An illustrative example of grouping by block-matching for images is given in Fig. 19, where a reference block is denoted by "R" and the ones matched to it are denoted in rectangular boxes.

After grouping the 2D image fragments to form a 3D array, collaborative filtering is applied to 3D arrays. Given a group of n estimates, the collaborative filtering of the group produces n estimates, one for each of the grouped fragments. Effective collaborative filtering can be realized as shrinkage in the transform domain. Given $d+1$ dimensional groups of similar fragments, the first step in collaborative shrinkage is to apply a $(d+1)$-dimensional linear transform to the group. Then soft- and hard-thresholding or Wiener filtering is used to shrink the transform coefficients to attenuate the noise. The final step in collaborative filtering is to apply an inverse transform to produce estimates of all grouped fragments. Generally, groups formed from natural images as shown in Fig. 19 are characterized by intra-fragment correlation(appears between pixels of each grouped fragment) and inter-fragment correlation(appears between the corresponding pixels of different fragments). The 3D transform can take advantage of both kinds of correlation and thus can produce a sparse representation of the true signal in the group. This makes the shrinkage very effective in attenuating the noise.

3.4.3 Outline of the Denoising Algorithm

In the denoising algorithm grouping is realized by block matching, and collaborative filtering is accomplished by shrinkage in a 3D transform domain. All the image

(a) (b)

FIG. 19: Illustration of grouping blocks from noisy MR images corrupted by Gaussian noise. Each fragment shows a reference block "R" and a few of the blocks matched to it.

fragments used in matching are square blocks of fixed size. The general procedure of denoising is implemented in two steps [5]. This is described as follows:

Step 1. *Basic Estimate.*

a) *Block-Wise Estimate.* For each block in the noisy image

i) *Grouping.* Apply the block matching technique to find similar blocks and stack them to form a 3D array.

ii) *Collaborative hard-thresholding.* Apply a 3D transform to the groups formed and apply hard thresholding to attenuate the noise and invert the 3D transform to produce estimates of all grouped blocks to their original positions.

b) *Aggregation.* Compute the weighted average of all the overlapping block wise estimates, this results in the basic estimate.

Step 2. *Final estimate.* After obtaining the basic estimate from step 1, perform the following steps.

a) *Block-wise estimates.* For each block,

i) *Grouping.* Apply block matching for the basic estimate to find similar blocks and form two groups, one from the noisy image and one from the basic estimate.

ii) *Collaborative Wiener filtering.* Apply a 3D transform on both groups. Perform Wiener filtering on the noisy one to produce estimates of all grouped blocks by applying the inverse 3D transform on the filtered coefficients, and return the estimates of the blocks to their original positions.

b) *Aggregation.* The final estimate of the true image is computed by aggregating all of the obtained local estimates using a weighted average.

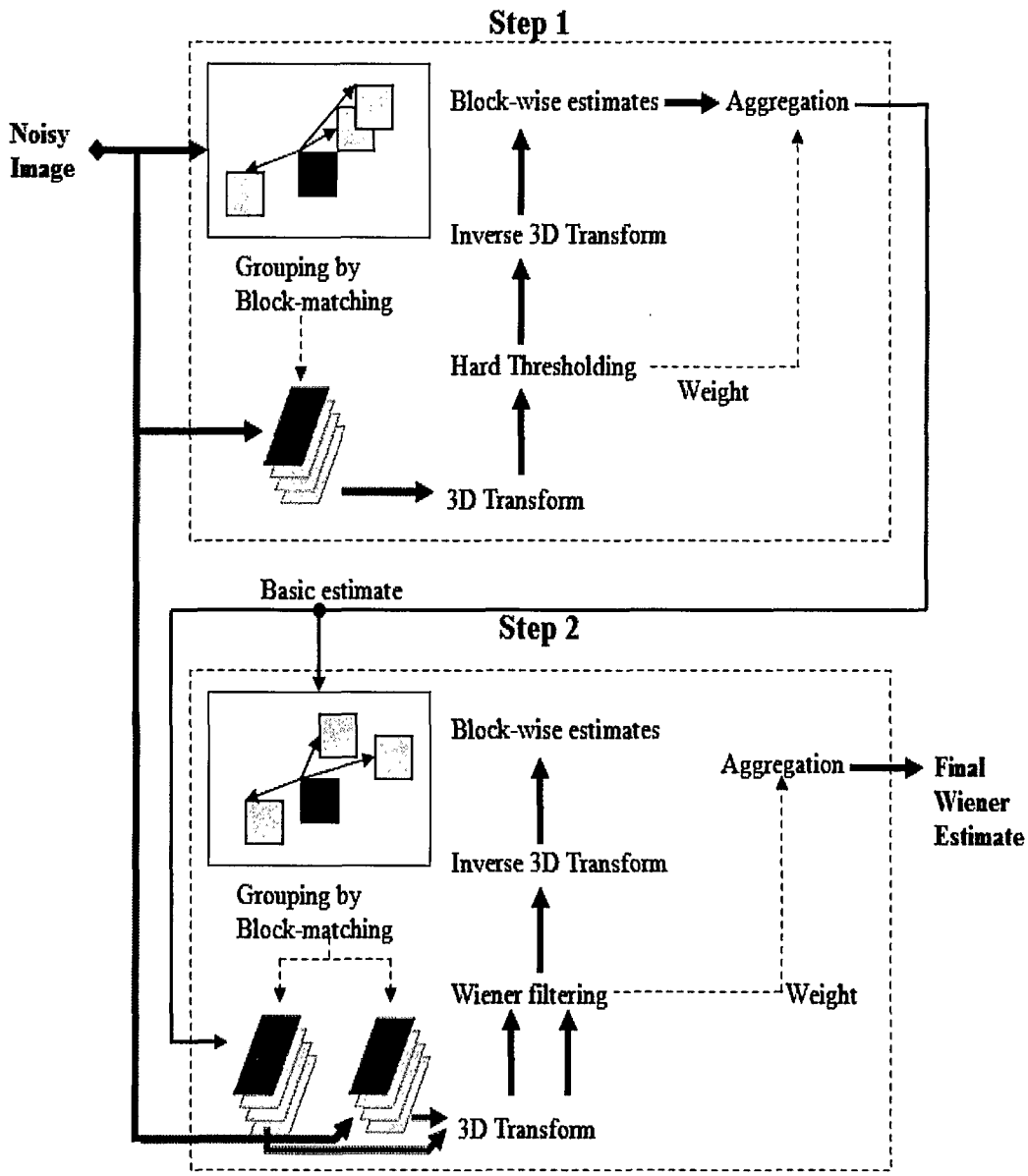
The reason for the second step in the above procedure is that, using the basic estimate instead of a noisy image improves the grouping. The flow chart of the algorithm is illustrated in Fig. 20. In the figure, operations surrounded by dashed lines are repeated for each processed block marked with "R".

3.4.4 Algorithm Implementation

Implementing the above denoising method directly is computationally demanding. In order to realize a practical and efficient algorithm, the authors of this denoising algorithm have imposed some constraints by defining some parameters. The following are the list of parameters and their descriptions.

1. N2: Maximum number of similar blocks (maximum size of the 3rd dimension of a 3D array).

2. Ns: Size of the search neighborhood for full-search block-matching(BM). It must be odd.



(a)

FIG. 20: Flowchart of the image denoising algorithm.

3. τ_match: Threshold for the block distance (d-distance)

4. λ_thr2D: Threshold parameter for the coarse initial denoising used in the d-distance (distance between reference block and the matched block) measure.

5. λ_thr3D: Threshold parameter for the hard thresholding in 3D transform domain.

6. β: Parameter of the 2D kaiser window used in the reconstruction. Kaiser window is used to reduce the border effects that can appear due to the usage of certain 2D transforms.

7. N2_Wiener: Maximum number of similar blocks for step 2.

8. Ns_Wiener: Length of side of search neighborhood in step 2.

9. τ_match_Wiener: Threshold for block distance in step 2.

10. β_Wiener: 2D kaiser window used in step 2.

11. σ: Standard deviation of noise.

12. N1: N1 x N1 is the block size used for the hard-thresholding (HT) filtering. This is the block size of the kaiser window used during the shrinkage of transform coefficients using hard thresholding.

13. Nstep: Sliding step to process every next reference block. Rather than sliding by one pixel to every next reference block, use a step of Nstep pixels to move to the next reference block.

14. N1_Wiener: N1 x N1 is the block size used for the HT filtering in step2

15. Nstep_Wiener: Sliding step to process every next reference block in step2

The values of above parameters can be varied between certain thresholds. For a given noisy image, a different set of parameters will result in a different denoised image. Based on several experiments, the authors have proposed a set of parameters

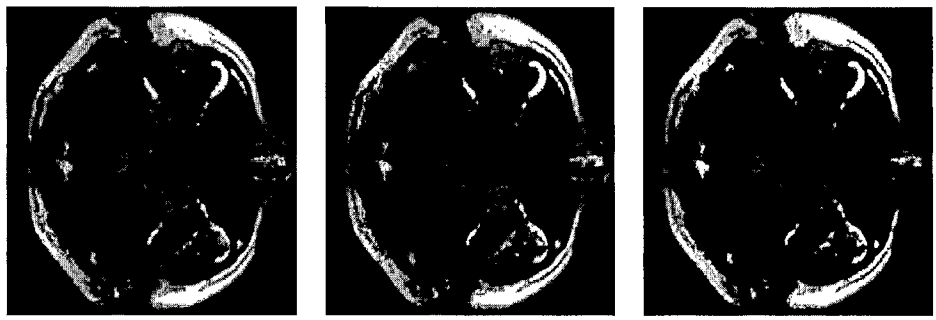
(a) (b) (c)

FIG. 21: Denoising results. (a) Noise free image. (b) Noisy image with sigma value 25. (c) Result after denoising.

(a) (b) (c)

FIG. 22: Denoising results. (a) Noise free image. (b) Noisy image with sigma value 15. (c) Result after denoising.

that produce best results, and these are used as default parameters. In this section, the results obtained by using the default parameters of the algorithm are presented. Fig. 21(a) shows a noise free MR image obtained from brain web [22]. Additive white Gaussian noise with zero mean and standard deviation of 25 is added to the noise free image shown in Fig. 21(b). The result of the denoising algorithm and the default parameter values is shown in Fig. 21(c). Similarly, results of the algorithm for noise standard deviations of 15 and 6 are shown in Fig. 22 and Fig. 23 respectively.



(a) (b) (c)

FIG. 23: Denoising results. (a) Noise free image. (b) Noisy image with sigma value 6. (c) Result after denoising.

3.4.5 Discussion

Consider the results shown in Fig. 21. After applying the denoising algorithm to the noisy image shown in Fig. 21(b), many interesting features in the image are lost in the process of denoising. From Fig. 21(c) it is clear that many edges in the images are lost due to extra smoothing. This is the same for the results shown in Fig. 22 and Fig. 23. For effective image analysis, edge information of an image should be preserved along with a good amount of denoising. The main drawback of the current denoising algorithm is that the interesting structures in the image are not preserved well. This problem exists not only with the current denoising algorithm but also with many other denoising algorithms. Therefore, there should always be a tradeoff between structure information preservation and the amount of denoising. Optimum results can be obtained by selecting parameters that lead to optimal results. This is described in the following section.

3.5 PROPOSED OPTIMIZATION PROCEDURE

The most common problem with the existing denoising methods is that some interesting structures in the image will be removed from the image during the process of denoising. Such interesting structures in an image often correspond to discontinuities in the image that provide important information. Especially in MRI analysis, discontinuities are very important as they are used to differentiate the pathological tissue from the normal tissue. Therefore, there is always a tradeoff between noise removal and structure information preservation. Optimizing the outcome of the denoising method is crucial for effective image analysis. In the proposed research, multiobjective evolutionary techniques are used for the optimization procedure. In this section, the proposed optimization algorithm is discussed. The optimization procedure is applied to the denoising algorithm presented in section III.5. The results of the denoising algorithm are compared with the results obtained after optimization.

3.5.1 Multiobjective Optimization

Many real world applications have several multiple conflicting objectives. Multiobjective optimization problems occur whenever optimal decisions need to be taken in the presence of trade-offs between two conflicting objectives. Maximizing profit and minimizing the cost of a product, maximizing performance and minimizing fuel consumption of a vehicle are examples of multi-objective optimization problems. It is rare that there is a single solution that simultaneously optimizes all the objectives. Therefore, when dealing with multiobjective optimization problems, rather than a

single solution we normally look for a set of solutions, and a designer can make trade-offs within this set. The resulting solution set is said to be Pareto optimal and the solutions are said to be Pareto efficient [23]. The plot of the objective functions whose vectors are in the Pareto optimal set is called the Pareto front.

In the proposed research work, the concept of multiobjective optimization is used for optimizing the results of denoising methods. Many existing denoising methods are associated with a set of variable parameters. Optimization can be achieved by selecting optimal parameters. Selecting optimal parameters for denoising algorithms is characterized by the presence of two conflicting objectives, the amount of noise removal and structure preservence. The objective functions are difficult to express in a closed form. A practical solution is to look at this parameter setting problem as a multiobjective evolutionary problem [24]. The Pareto optimal solutions for this case are obtained by using the SPEA2 algorithm [25]. This optimal set provides more freedom for a decision maker when making trade-offs between the two objectives.

3.5.2 Cost Functions

As mentioned earlier, selecting the optimal parameter set for any denoising algorithm has two conflicting objectives, which are noise removal and structure preservence. The fitness values of objective functions are calculated by evaluating the cost functions. Cost functions are used to assess the outcome of the denoising algorithm by quantifying the amount of denoising and the amount of structure information being preserved. The selection of cost functions play a crucial role in the optimization procedure. For instance, mean square error (MSE) is one means to quantify the amount by which the denoised image differs from the original noise free image. For a

denoising method, a low value of MSE is preferable since the denoised image will be as close as possible to the noise free image. In this section, the cost functions defined for the optimization procedure are discussed. Later these are minimized using the SEPA2 algorithm.

Mean Square Error

As mentioned earlier, MSE value indicates the amount by which an estimated value differs from a true value. In the current context, MSE is calculated between the denoised image and the ground truth(noise free) image. Let $f(x, y)$ be the noise free image of size M x N and $\hat{f}(x, y)$ be the output of the denoising algorithm, then MSE is calculated as follows:

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (f(i, j) - \hat{f}(i, j))^2 \tag{24}$$

Entropy

Entropy in image processing is a statistical measure of randomness or information content of an image. This can be used to characterize the texture of a given image. Let $f(x, y)$ be the noise free image and $\hat{f}(x, y)$ be the output of the denoising algorithm and $D(x, y)$ be the difference between $f(x, y)$ and $\hat{f}(x, y)$. If the power spectrum of $D(x, y)$ is white, i.e. the information content in the power spectrum is randomly distributed over the entire frequency range in equal proportions, this indicates that the amount of denoising is uniform for all frequency components. If the information content of the power spectrum is concentrated only in particular frequencies, this indicates that the features corresponding to those frequency components are lost during denoising, i.e. the amount of denoising is not uniform. In other words, if

the information content of the power spectrum of $D(x,y)$ is less and entropy is less, we can say that the structural information is preserved by the denoising algorithm. However, if $D(x,y)$ has a regular pattern, which implies more information content, structural information is not preserved. Therefore, a lower value of entropy is an indicator of good structure preservence. If the information content in an image can be represented with N gray level values and x occurs with a probability of $p(x)$, then entropy is given as

$$Entropy = -\sum_{i=1}^{N} p(x_i) log(p(x_i)) \tag{25}$$

Entropy is typically measured in bits per symbol (gray level).

Second Derivative

Let $f(x,y)$ be the noise free, $\overline{f}(x,y)$ be the output of the denoising algorithm and $\overline{f}''(x,y)$ be the second derivative of $\overline{f}(x,y)$. The second derivative of an image gives the edge information. The inverse of the standard deviation of $\overline{f}''(x,y)$ can be used as other cost function, which can assess the structural information of a given image. A low value of $1/SD$(standard deviation) is preferred, since edges in the estimated image must be preserved. The second derivative of an image can be approximated by convoluting the image with a Laplacian mask, which is given below.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}. \tag{26}$$

3.5.3 SPEA2 Algorithm

Strength Pareto Evolutionary Algorithm 2 (SPEA2) [26] is a relatively recent technique for finding or approximating the Pareto optimal set for multiobjective optimization problems. The reason for using SPEA2 for optimization of cost functions is because of its fast convergence rate. In this section the main details of the optimization procedure are discussed. For any general genetic algorithm, there are four steps:

1. Randomly initialize the solution population.

2. Evaluate and assign a fitness value for each individual in the population according to its performance.

3. Select individuals based on their performance so that better individuals are more likely to be selected for producing the next generation.

4. Use crossover and mutation to produce the next generation from the selected individuals.

In the current optimization procedure, lower fitness values of cost functions indicate better performance. Steps 2 through 4 are repeated until the generation number is reached. SPEA2 differs from a standard genetic algorithm in the following aspects.

Environmental Selection: Apart from regular population, which is used in the genetic algorithm, in SPEA2 an archive is maintained that contains all the non-dominated solutions from the previous generation. A member of the archive is removed if the following conditions are satisfied.

1. A dominant solution is found in the current generation or

2. the archive maximum size is reached and the member's performance is worse than

that of others.

Fitness Evaluation: Let P_t and $\overline{P}_t$ denote the population and archive respectively; each individual i in P_t and $\overline{P}_t$ is assigned a strength value $S(i)$ that denotes the number of dominant solutions,

$$S(i) = ||\{j | j \in P_t + \overline{P}_t \wedge i \succ j\}||, \tag{27}$$

where $||\cdot||$ represents the cardinality of a set, $+$ represents the multiset union and $\succ$ represents the Pareto dominance relation. The Pareto dominant relation is described in later sections. The raw fitness value $R(i)$ is given as follows

$$R(i) = \sum_{j \in P_t + \overline{P}_t, j \succ i} S(j) \tag{28}$$

The final fitness values for ith individual is given as

$$F(i) = R(i) + D(i), \tag{29}$$

where $D(i)$ are the density values estimated as follows

$$D(i) = \frac{1}{{\delta_i}^k + 2}, \tag{30}$$

where ${\delta_i}^k$ denotes the kth nearest distance for the ith individual among P_t and $\overline{P}_t$ in objective space. k is usually set as $\sqrt{N + \overline{N}}$, where N is the population size and $\overline{N}$ is the archive size.

Mating Selection: In SPEA2 all candidates are chosen from the archive using a binary tournament selection procedure scheme unlike the standard genetic algorithm where the probability of choosing an individual depends on the performance. In the binary tournament selection, first we randomly select two individuals and only the better one survives.

3.5.4 Problem Formulation

Let $f_1(\overrightarrow{x})$ and $f_2(\overrightarrow{x})$ denote the cost functions defined, where $\overrightarrow{x}$ is an n-dimensional threshold vector. The values of cost functions are obtained by modifying the parameters associated with the denoising algorithm. The value of n depends on the denoising algorithm being used. The elements of the threshold vector are the algorithm parameters that can be varied between certain thresholds. The range for each threshold is experimentally determined such that the range is wide enough to include all solutions of interest. The cost functions obtained by running the denoising algorithm using the range of parameters in the threshold vector are to be minimized, and this can be treated as a multiobjective optimization problem given mathematically as follows [27],

$$min\overrightarrow{f}(\overrightarrow{x}) = \{f_1(\overrightarrow{x}), f_2(\overrightarrow{x})\} \tag{31}$$

For the current denoising algorithm, we have $n = 15$ parameters. The threshold vector is given as follows:

$$\overrightarrow{x} = \{x_1, \cdots, x_{15} | x_1, \cdots, x_{11} \in \mathbb{R}, x_{12}, \cdots, x_{15} \in I\} \tag{32}$$

The multiobjective minimization problem (MOP) for the current denoising algorithm can be stated as follows,

$$min\overrightarrow{f}(\overrightarrow{x}) = \{f_1(\overrightarrow{x}), f_2(\overrightarrow{x})\} \tag{33}$$

subject to

$$\begin{cases} x_1 \in (0, 36], \\ x_{2\in}(0, 89], \\ x_{3\in}(0, 10000], \\ x_{4\in}(0, 1], \\ x_{5\in}(0, 5], \\ x_{6\in}(0, 4], \\ x_{7\in}(0, 36], \\ x_{8\in}(0, 89], \\ x_{9\in}(0, 10000], \\ x_{10\in}(0, 5], \\ x_{11\in}(1, 15], \\ x_{12\in}(2, 32], \\ x_{13\in}(4, 32], \\ x_{14\in}(2, 32], \\ x_{15\in}(4, 32]. \end{cases} \tag{34}$$

x_1 to x_{15} correspond to parameters 1 to 15 defined in Section III.4.4 respectively. The range of parameter values are chosen such that the final optimal set contains a wide range of solutions. The global optima of the multiobjective optimization problem is the Pareto front [28] determined by evaluating each member of the Pareto optimal set. The Pareto optimal set consists of solutions that are not dominated by any other solutions. A solution, $\overrightarrow{x_1}$, is said to dominate, $(\succ)$, $\overrightarrow{x_2}$, if the objective vector, $\overrightarrow{f}(\overrightarrow{x_1})$, is less than or equal to $\overrightarrow{f}(\overrightarrow{x_2})$ in all attributes and strictly less than at least one

attribute,

$$\begin{cases} \overrightarrow{x_1} \succ \overrightarrow{x_2}, \textit{iff} \\ \forall i \in \{1,2\} : f_i(\overrightarrow{x_1}) \leq f_i(\overrightarrow{x_2}) \wedge \exists j \in \{1,2\} : f_j(\overrightarrow{x_1}) < f_j(\overrightarrow{x_2}). \end{cases} \tag{35}$$

The plot of the objective functions whose vectors are in the Pareto optimal set is called the Pareto front. From the Pareto optimal set a decision maker can select the solution that has the best trade-off, satisfying all the objectives [29].

3.5.5 Algorithm Outline

SPEA2 algorithm is used to minimize the cost functions defined in Section III.5.2. The optimization procedure is described as follows.

Step 1. **Initialization:**

Initialize the population size, (N), archive size, $(\overline{N})$, and generation number, (T). The typical values are $N = 100$, $\overline{N} = 100$, and $T = 200$. Along with these values, we have to initialize the parameters of the denoising algorithm P_t randomly and generate the empty archive $\overline{P}_t$.

Step 2. **Fitness Evaluation:**

In this step, the denoising algorithm that is to be optimized takes the parameters from P_t and $\overline{P}_t$ and runs the denoising algorithm on a given image. This step returns the values of the cost functions that are selected for optimization. The fitness values of each cost function are then calculated using the procedure mentioned in Section III.5.3.

Step 3. **Termination:**

Nondominated parameters in $\overline{P}_t$ are returned as a final result if $t > T$.

Step 4. **Environmental Selection:**

Copy all individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. Delete the worst solutions in $\overline{P}_{t+1}$ if the size of $\overline{P}_{t+1}$ exceeds $\overline{N}$. If the size is less than $\overline{N}$, copy the dominant solutions in P_t that have smaller fitness values.

Step 5. **Mating Selection:**

Using the binary tournament procedure, select 100 individuals from $\overline{P}_t$ with a replacement.

Step 6. **Reproduce:**

The next generation is reproduced using mutation and standard crossover procedures. The probability values of crossover and mutation are chosen as 0.9 and 0.001 respectively. Store the results in $\overline{P}_{t+1}$ and increment t by 1 and go to step 2.

3.6 RESULTS

The proposed optimization procedure is to optimize the results of image denoising using block matching and 3D collaborative filtering. Images to run the denoising algorithm are obtained from brainweb [22]. Using the proposed optimization procedure two pairs of cost functions are minimized for the denoising algorithm being tested. From the obtained Pareto fronts, we choose several points on the curve, where each point on the curve corresponds to one parameter set of the denoising algorithm. The results of the denoising algorithm with the selected parameters from the Pareto front are compared with that of the results obtained using the default parameters of the denoising algorithm.

3.6.1 Simulated Brain Database

All the images used for the testing optimization procedure are obtained from brainweb, which contains a simulated brain database(SBD) [22]. SBD contains simulated brain MRI data based on two anatomical models: normal and multiple sclerosis (MS). This contains full 3-dimensional data volumes simulated using three sequences (T1-, T2-, and proton-density- (PD-) weighted) and a variety of slice thicknesses, noise levels, and levels of intensity non-uniformity. These data are available for viewing in three orthogonal views (transversal, sagittal, and coronal), and for downloading. In this pre-computed simulated brain database (SBD), the parameter settings are fixed to 3 modalities, 5 slice thicknesses, 6 levels of noise, and 3 levels of intensity non-uniformity. Fig. 24 shows some of the simulated images from brainweb.

3.6.2 Results of Default Parameters

Before presenting the optimized results, in this section the results of the denoising algorithm using previous default parameters are presented. Fig. 25(a) is a simulated noise free image from brain web. To this, noise with standard deviation of 6.0 is added as shown in Fig. 25(b). The result of the denoising algorithm using the default parameters of the algorithm is shown in Fig. 25(c). The difference between the denoised image and the noise free image is shown in Fig. 25(d). For this result, the value of entropy obtained is 3.1955, MSE is 1.33 x 10^{-4}, and 1/SD value is 0.9149. These points are shown in Fig. 29 and Fig. 30, denoted by $*$. In a later section it will be shown that these solutions are dominant solutions whereas the results obtained from the Pareto front are non-dominated solutions. From the difference image it can

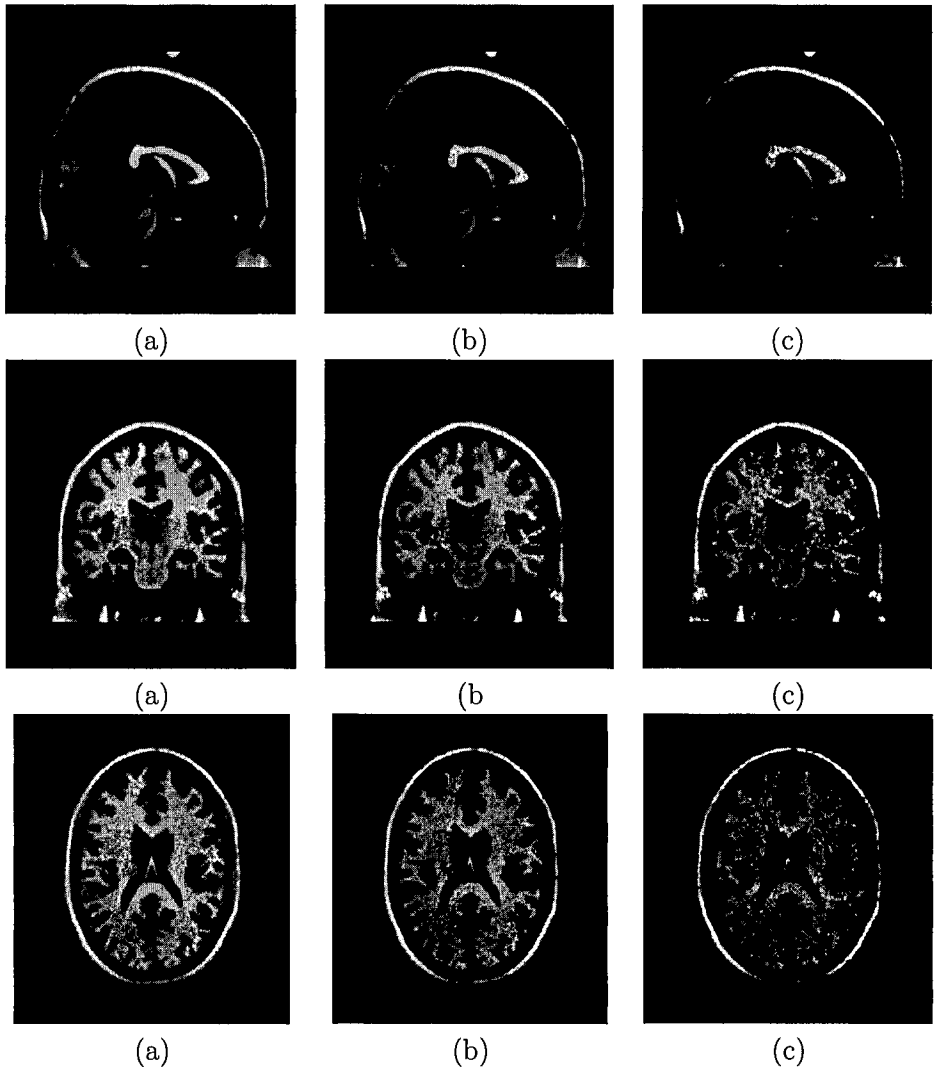


FIG. 24: Simulated brain MRI images from brain web. (a) Noise free image, (b) Image with 3 percent noise, and (c) Image with 9 percent noise.

be clearly seen that the edges in the denoised image are removed during the process of denoising. The power spectrum of the denoised image is shown in Fig. 25(e). It has more information content, i.e. the entropy value is high which is not desirable.

3.6.3 Results of Proposed Algorithm

In step 2 (Fitness Evaluation) of the optimization procedure, the cost function values are calculated from the results of the denoising algorithm. The Pareto front is calculated for two pairs of cost functions. One pair of cost functions is MSE and Entropy. 1/SD and MSE is the other pair of cost functions, where SD is the standard deviation of the second derivative of the denoised image. The Pareto front for MSE Vs. Entropy is shown in Fig. 26. Fig. 27 shows the Pareto front for 1/SD Vs. MSE. The number of points on each Pareto front is 100. Not all the points are of interest, but points corresponding to the region on the curve where the cost function are minimized will yield optimized results. To check the optimized denoising results, 11 points on each Pareto front are chosen, and the corresponding values of cost functions are shown in Table I and Table II. The selected points are marked on the Pareto curve of MSE Vs. Entropy, shown in Fig. 28, where Pset is a short form for parameter set. Similarly, another set of 11 points is chosen from the Pareto front of 1/SD and MSE. Each value of cost the function in the table corresponds to one parameter set of the denoising algorithm.

3.6.4 Comparison

The 11 points on each Pareto front correspond to 11 parameter sets. Each parameter set of the Pareto front of MSE Vs. Entropy is applied to the denoising

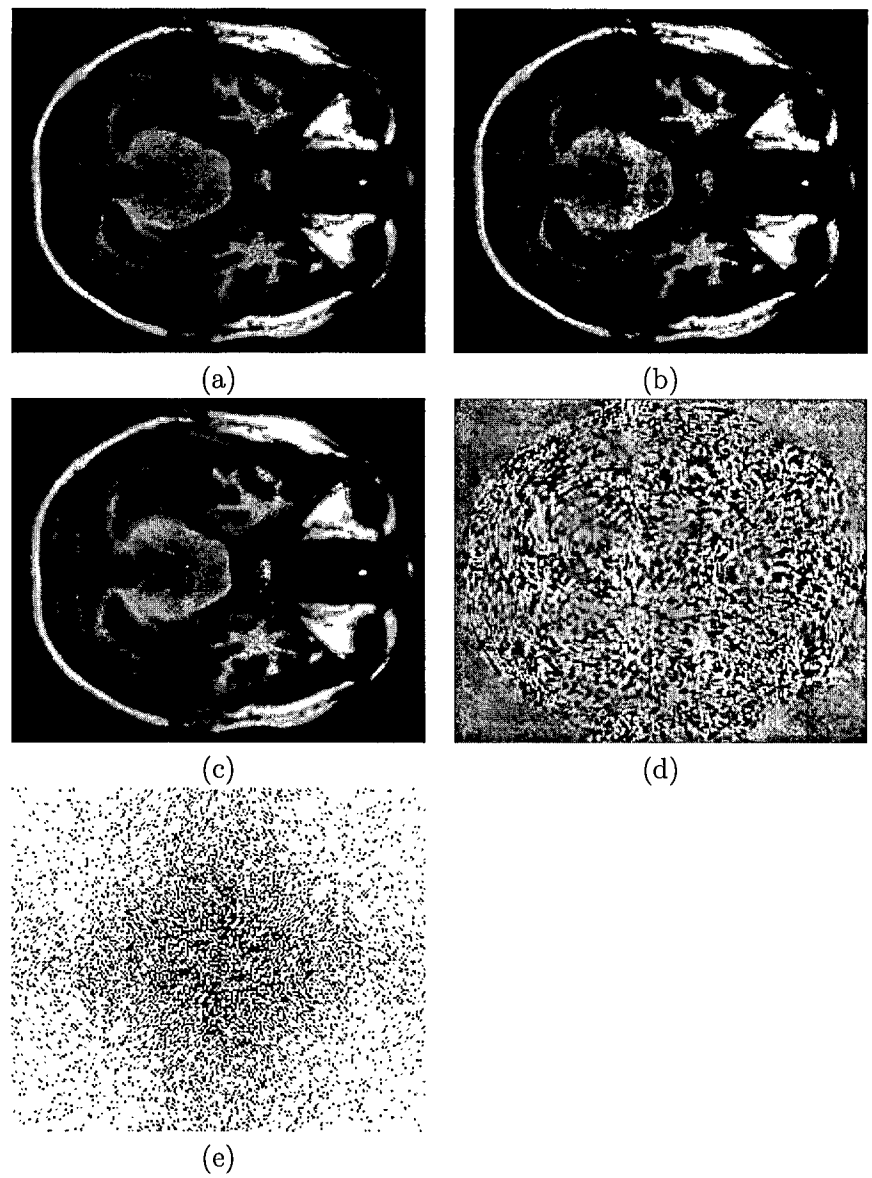
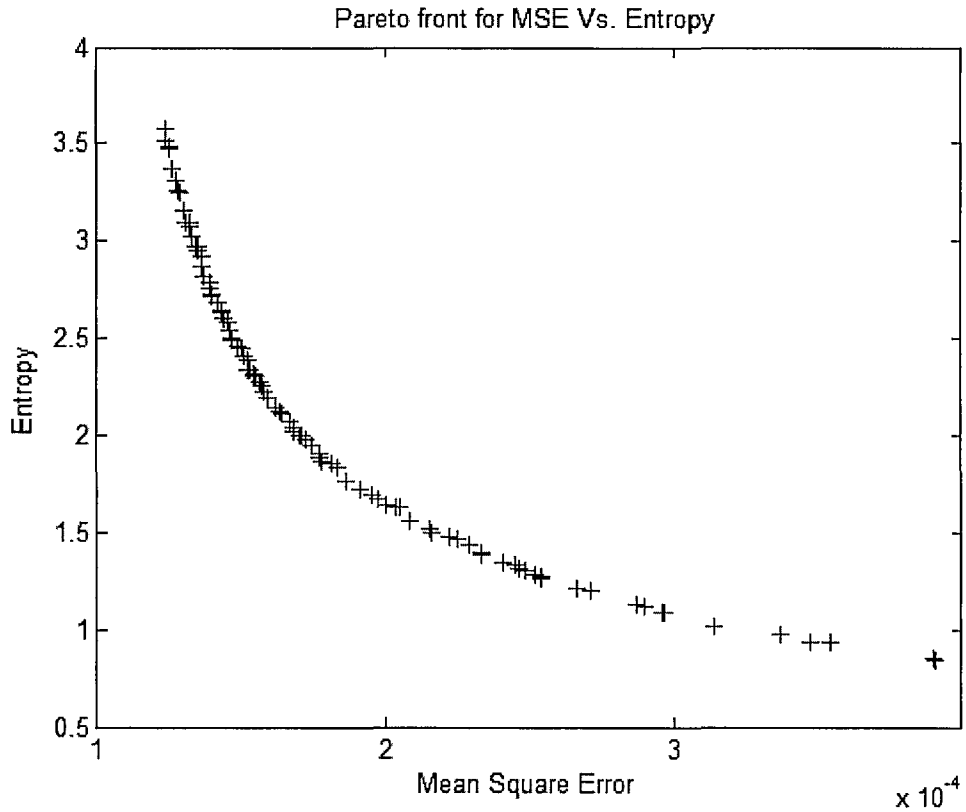


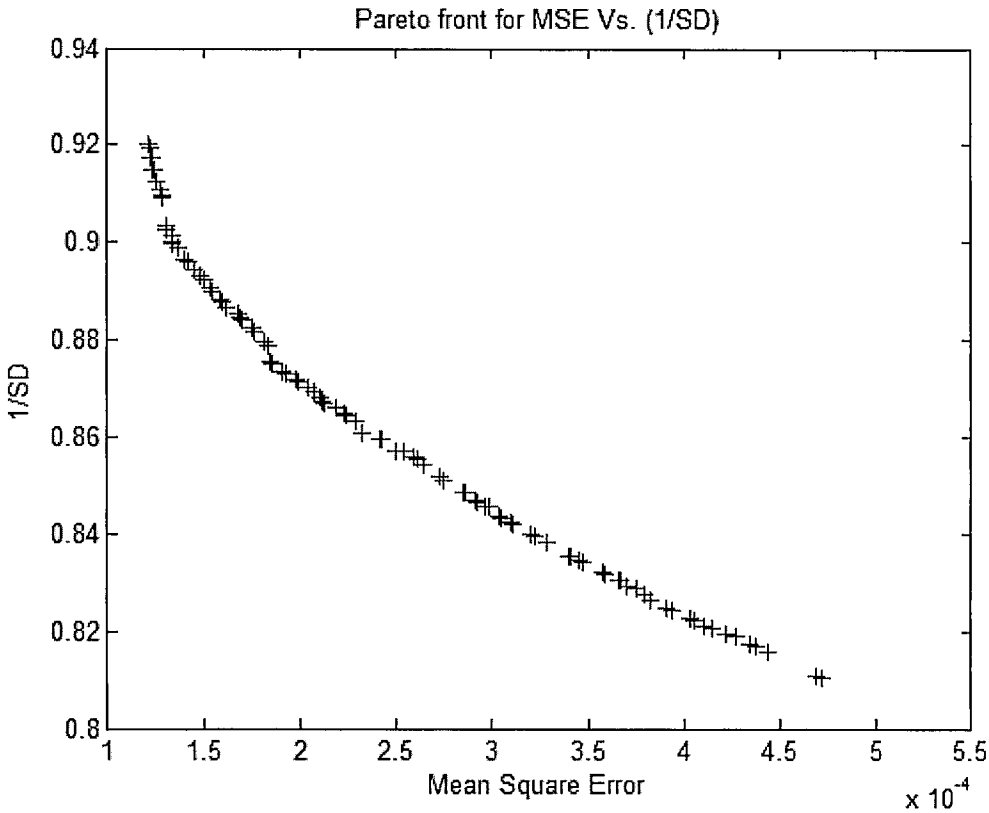
FIG. 25: (a) Simulated Input image. (b) Noisy image, (c) Denoised result of (b), (d)Difference of (a) and (c) (b), and (e) Power Spectrum of (d).



(a)

FIG. 26: Pareto front for Entropy vs. MSE.

algorithm, and the corresponding results are shown in Fig. 31, Fig. 32, and Fig. 33. Images in Fig. 31(a), Fig. 32(a), and Fig. 33(a) are the results of Parameter set 1 through Parameter set 11 of the Pareto front for the noisy image shown in Fig. 25(b). Fig. 31(b), Fig. 32(b), and Fig. 33(b) show the difference image (denoised - noise free) and Fig. 31(c), Fig. 32(c), and Fig. 33(c) show the power spectrum of the difference image. If these results are compared with results obtained by using the default parameters of the algorithm shown in Fig. 25, it is clear that the results obtained by the parameters using the proposed algorithm have a better amount of noise removal along



(a)

FIG. 27: Pareto front for 1/SD vs. MSE.

with good structure preservence. From the results it is clear that the Pareto front provided a set of solutions for denoising whereas the denoising algorithm with default parameters resulted in only one solution, which is a dominant solution whereas the solutions obtained from the optimized Pareto front are non-dominant. This is shown in Fig. 29 where points denoted by + are the solutions obtained by the proposed optimization procedure, and the point denoted by $*$ is the result obtained by the default parameters of the denoising algorithm. Results corresponding to parameter set 5, 6, 7, and 8 denoted by Pset5, PSet6, Pset7, and Pset8 in Fig. 28 have low entropy and

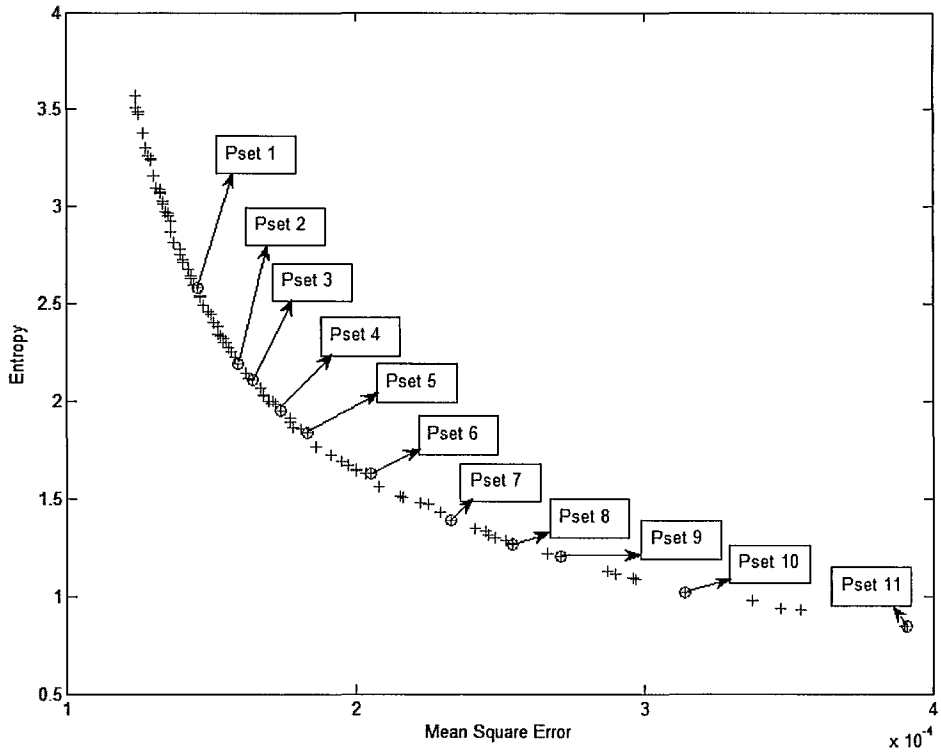
TABLE 1: Values of MSE and Entropy obtained from Pareto front.

Parameter Set	Entropy of Power Spectrum	Mean Square error$\times 10^{-3}$
1	2.5844	0.1450
2	2.1943	0.1590
3	1.9527	0.1640
4	1.8323	0.1740
5	1.6268	0.1830
6	1.4774	0.2050
7	1.6268	0.2220
8	1.2644	0.2540
9	1.2020	0.2710
10	1.0176	0.3140
11	0.8474	0.3910

TABLE 2: Values of 1/SD and MSE obtained from Pareto font.

Parameter Set	1/SD	Mean Square error$\times 10^{-3}$
1	0.8825	0.1750
2	0.8645	0.2240
3	0.8610	0.2320
4	0.8544	0.2640
5	0.8465	0.2920
6	0.8395	0.3220
7	0.8345	0.3450
8	0.8247	0.3910
9	0.8225	0.4050
10	0.8174	0.4340
11	0.8103	0.4720

MSE values. This region of the Pareto front has minimized cost functions, and the corresponding denoised results have a good amount of noise removal with structure information being preserved. This is very important for effective preprocessing such as segmentation. The results of the Pareto front corresponding to 1/SD Vs. MSE is shown in Fig. 34, Fig. 35, and Fig. 36. From Fig. 30 it is clear that the result obtained using the default parameters of the denoising algorithm are dominant, while



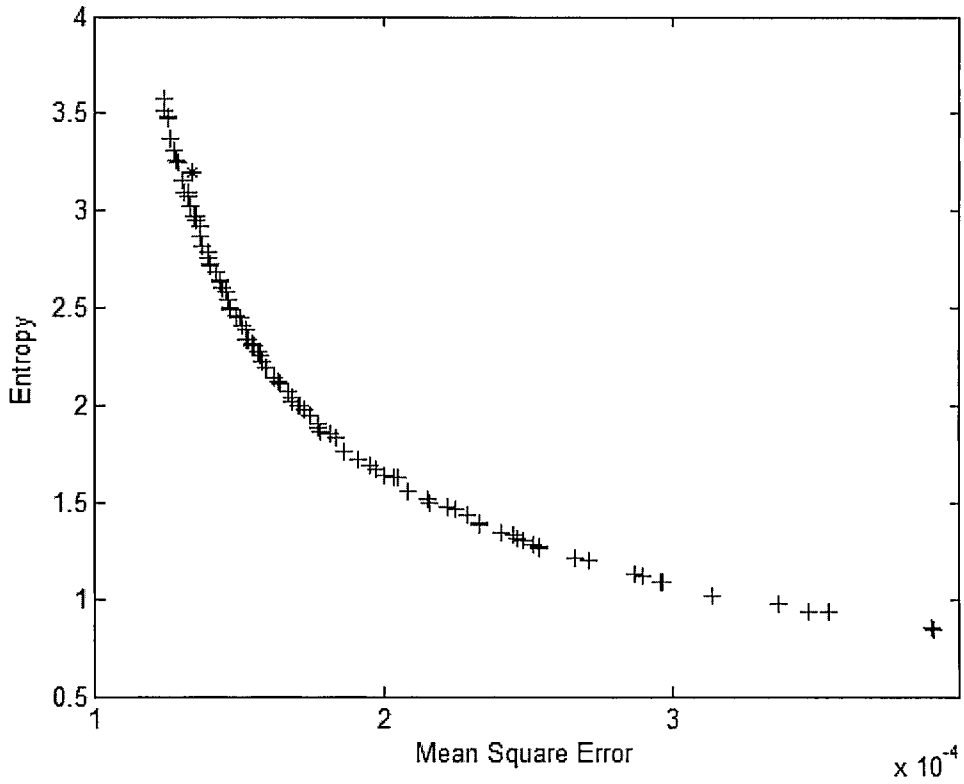
(a)

FIG. 28: Sampled points on the Pareto front of MSE vs. Entropy.

the solutions on the Pareto front are non-dominant.

3.7 CONCLUSION

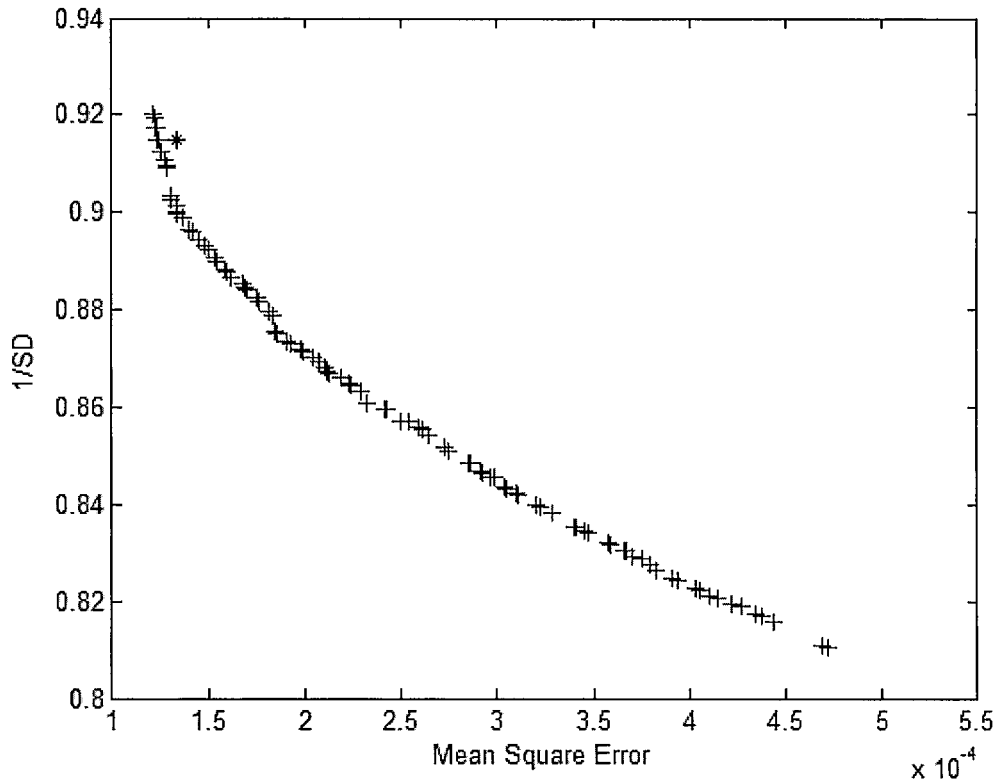
In this chapter, multiobjective optimization techniques are used to optimize the results of image denoising. The optimization is achieved by minimizing the defined cost functions using the SPEA2 algorithm. The SPEA2 algorithm is used to find the Pareto optimal set for the current multiobjective optimization problem. Instead of a single solution, using the proposed optimization algorithm we can have a set of



(a)

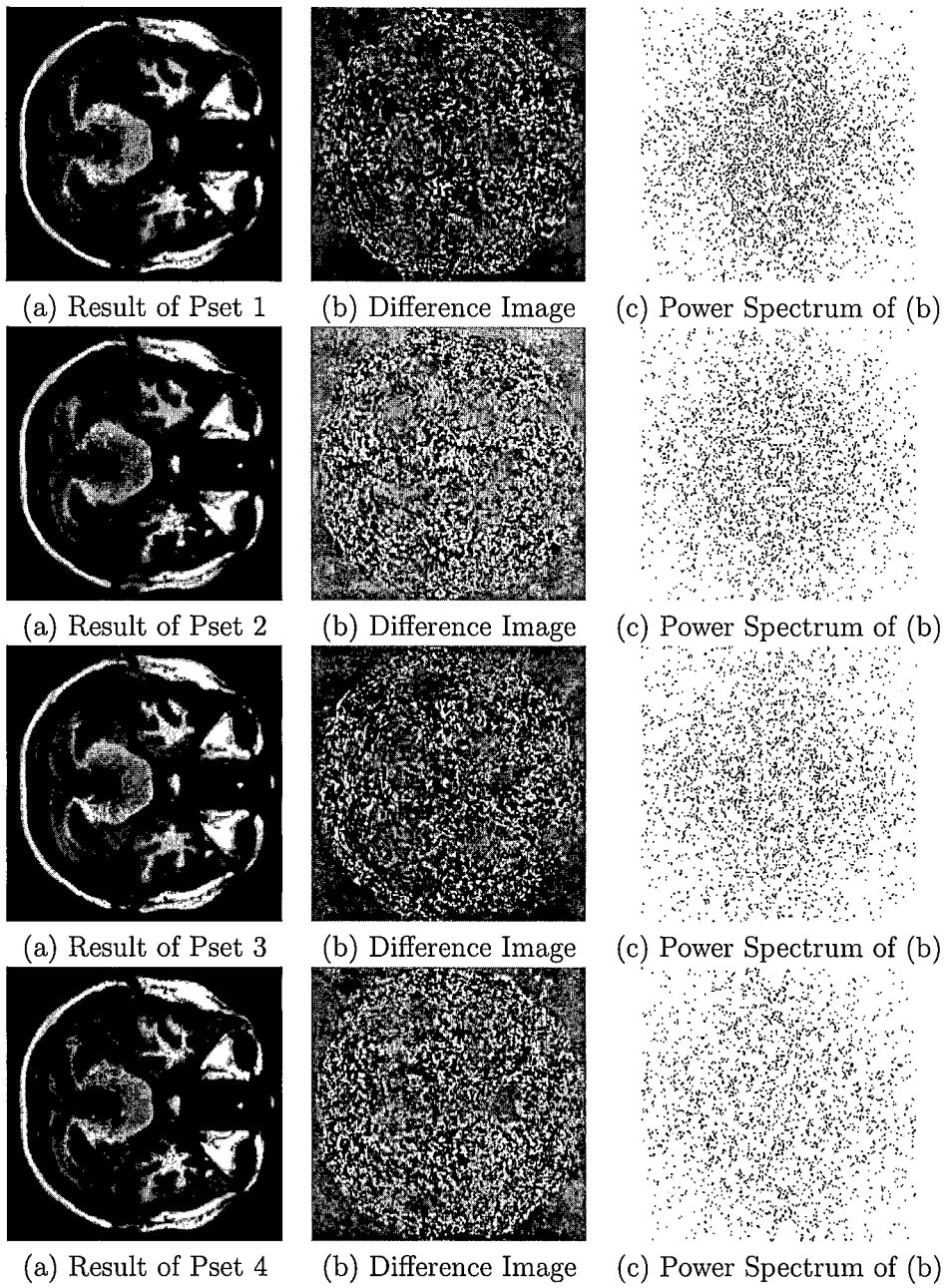
FIG. 29: Dominant solution marked as $*$ represents the MSE and Entropy values obtained by using the default parameters of the denoising algorithm.

optimal solutions that allows the designer to make trade-offs. Experimental results showed that the proposed optimization algorithm greatly enhances the results of the denoising algorithm in terms of noise removal and feature preservation.



(a)

FIG. 30: Dominant solution marked as $*$ represents the MSE and (1/SD) values obtained by using the default parameters of the denoising algorithm.



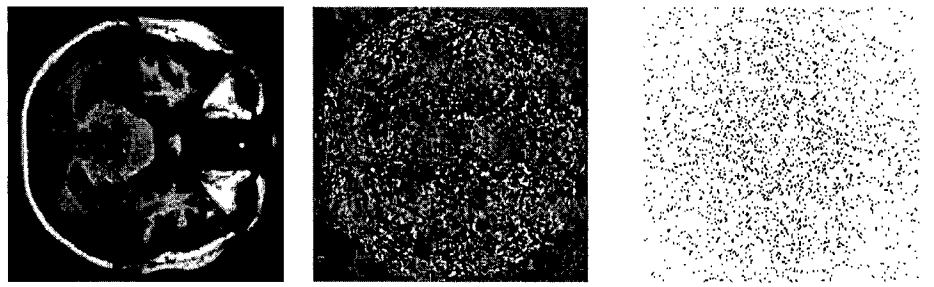
(a) Result of Pset 1 (b) Difference Image (c) Power Spectrum of (b)

(a) Result of Pset 2 (b) Difference Image (c) Power Spectrum of (b)

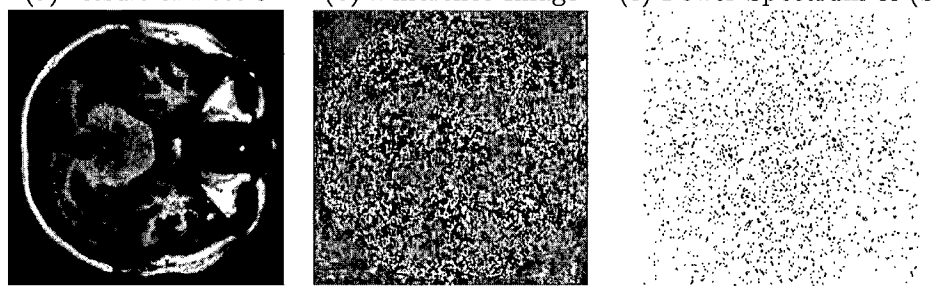
(a) Result of Pset 3 (b) Difference Image (c) Power Spectrum of (b)

(a) Result of Pset 4 (b) Difference Image (c) Power Spectrum of (b)

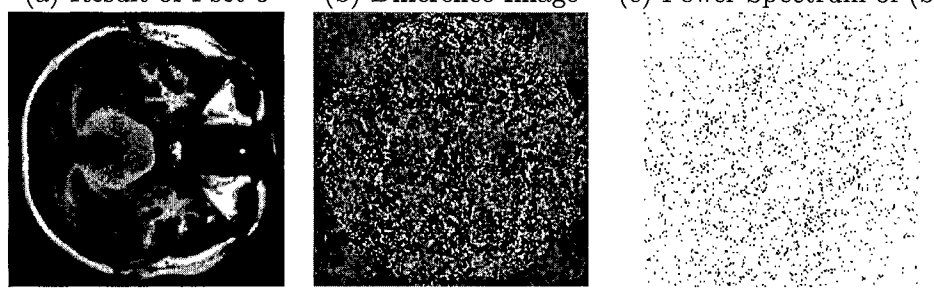
FIG. 31: Results obtained by using the parameters obtained from the Pareto front of MSE Vs. Entropy.



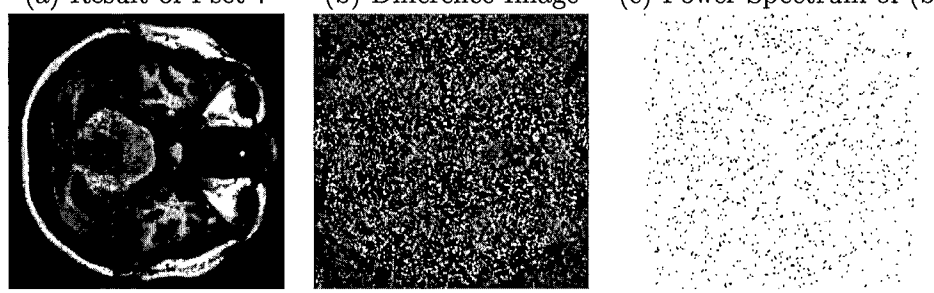
(a) Result of Pset 5 (b) Difference Image (c) Power Spectrum of (b)



(a) Result of Pset 6 (b) Difference Image (c) Power Spectrum of (b)

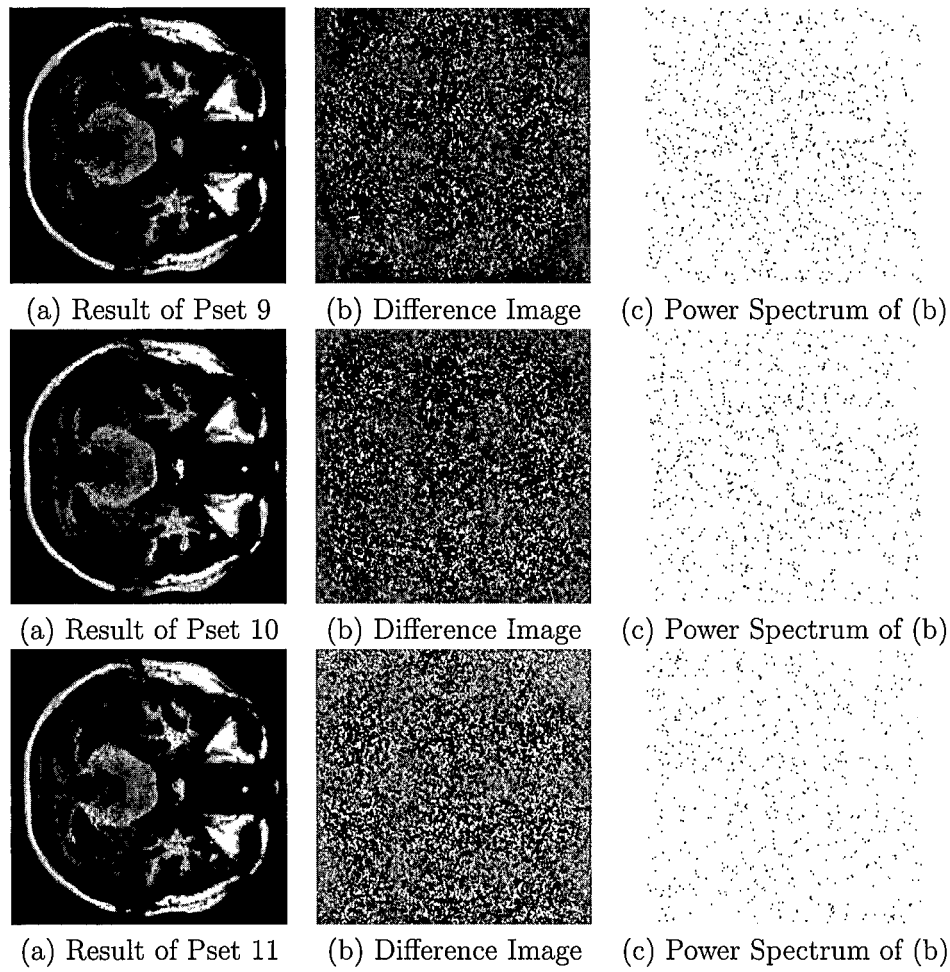


(a) Result of Pset 7 (b) Difference Image (c) Power Spectrum of (b)



(a) Result of Pset 8 (b) Difference Image (c) Power Spectrum of (b)

FIG. 32: Results obtained by using the parameters obtained from the Pareto front of MSE Vs. Entropy.



(a) Result of Pset 9 (b) Difference Image (c) Power Spectrum of (b)

(a) Result of Pset 10 (b) Difference Image (c) Power Spectrum of (b)

(a) Result of Pset 11 (b) Difference Image (c) Power Spectrum of (b)

FIG. 33: Results obtained by using the parameters obtained from the Pareto front of MSE Vs. Entropy.

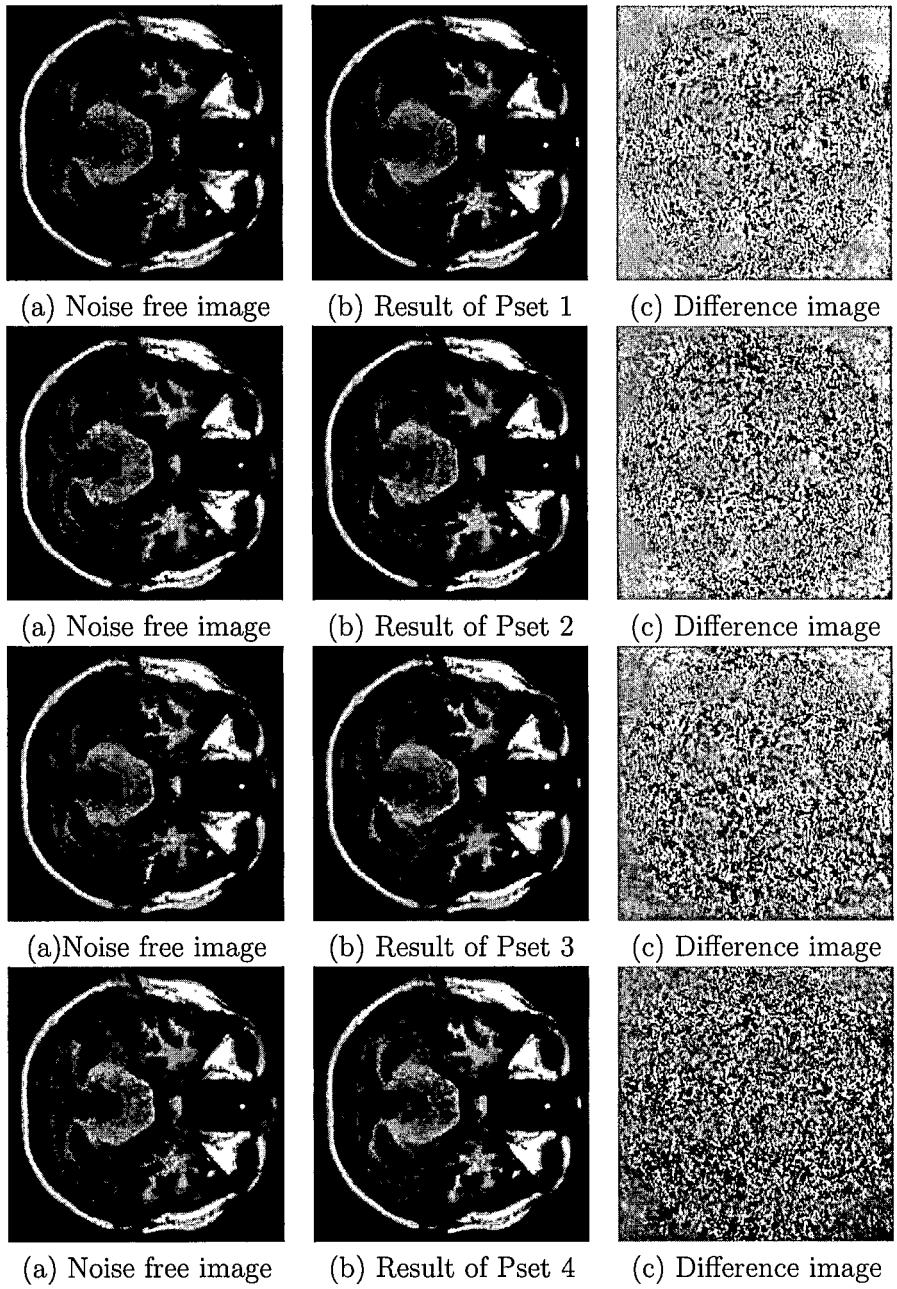
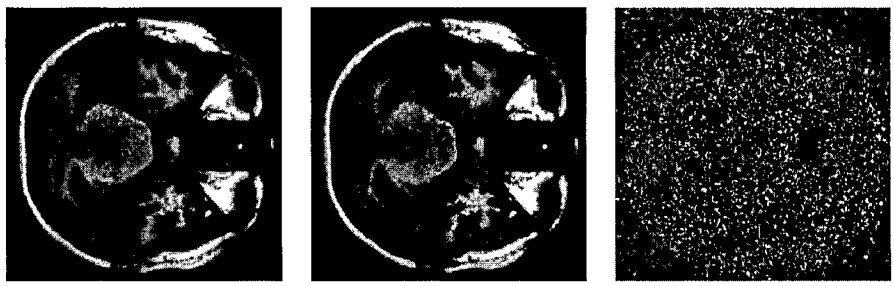
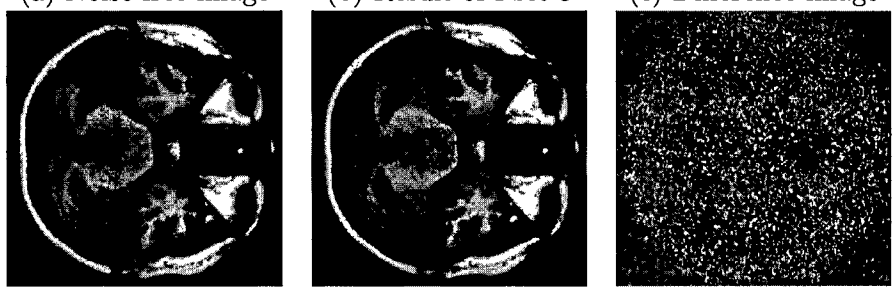


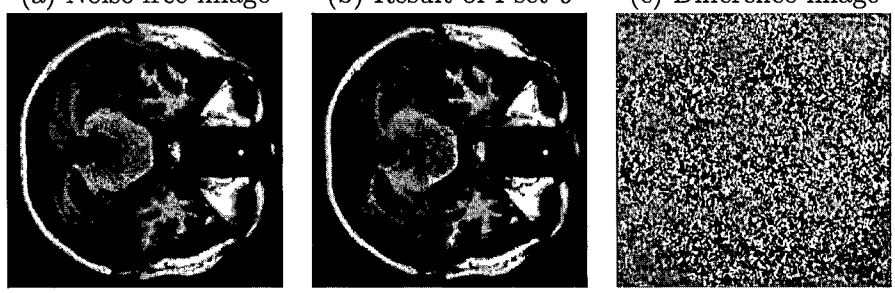
FIG. 34: Results obtained by using the parameters obtained from the Pareto front of 1/SD Vs. MSE.



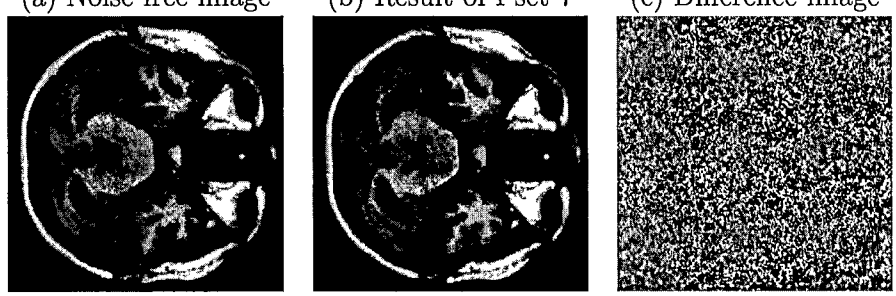
(a) Noise free image (b) Result of Pset 5 (c) Difference image



(a) Noise free image (b) Result of Pset 6 (c) Difference image

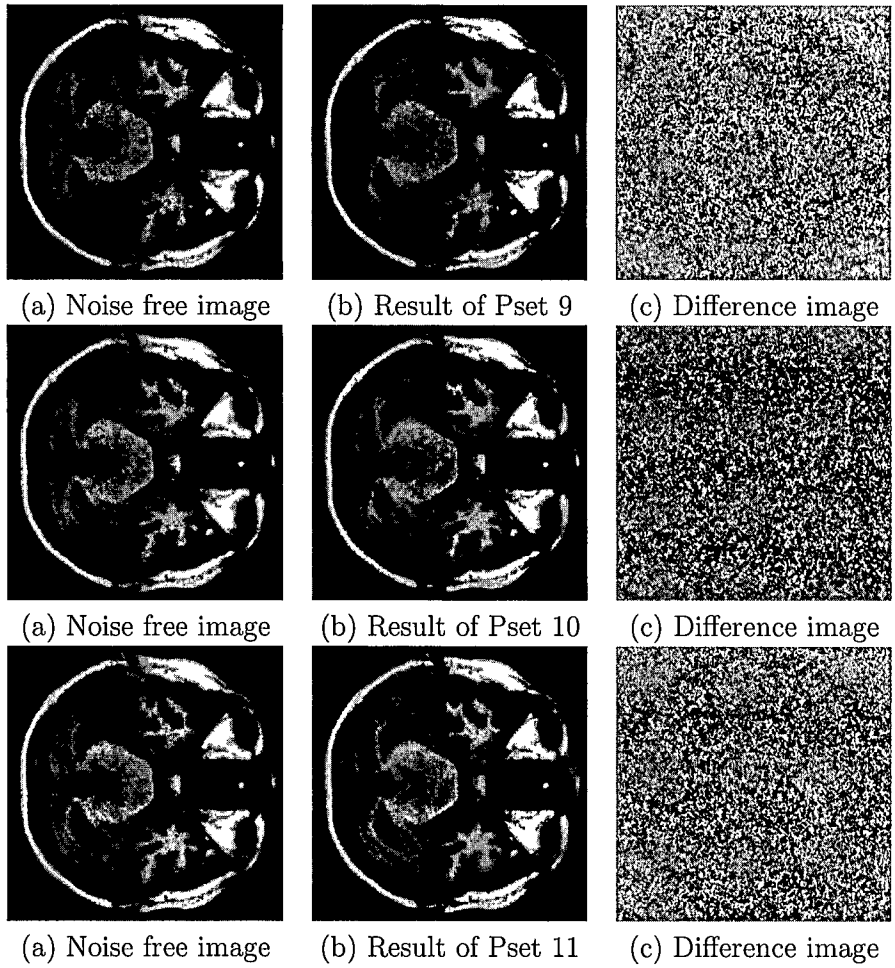


(a) Noise free image (b) Result of Pset 7 (c) Difference image



(a) Noise free image (b) Result of Pset 8 (c) Difference image)

FIG. 35: Results obtained by using the parameters obtained from the Pareto front of 1/SD Vs. MSE.



(a) Noise free image (b) Result of Pset 9 (c) Difference image

(a) Noise free image (b) Result of Pset 10 (c) Difference image

(a) Noise free image (b) Result of Pset 11 (c) Difference image

FIG. 36: Results obtained by using the parameters obtained from the Pareto front of 1/SD Vs. MSE.

CHAPTER 4

CONCLUSION

In this thesis, two algorithms for medical imaging applications are proposed. In the first part of the thesis, we proposed an algorithm to automatically generate textured images for wound debridement simulation. Previously these images were manually generated using various image editing tools such as Photoshop. Textured images generated manually, once loaded into the simulator, cannot be changed dynamically. Also, the method is computationally demanding. In the proposed method, we developed a two pass procedural method that can generate the textured images automatically. In the first pass of the algorithm, noise is generated for the entire wound image. In the second pass, noise is confined to the wound area by using elliptical shape composition. The results obtained by the proposed method are similar to results obtained by Photoshop. By automatically generating the textured images, we can dynamically change the images during simulation representing a variety of wound conditions. This greatly enhances the training outcome. The proposed method can be extended to model complex wound scenarios such as generating multiple wounds in the same image.

In the second part of the thesis, we proposed an algorithm to optimize the parameters associated with MRI image denoising methods. Medical images are generally corrupted by random noise during acquisition. Many existing methods proposed for denoising do not preserve the structure information well. They tend to blur the subtle features in the image such as discontinuities that are informative. Especially in MRI image analysis, discontinuities such as edges should be preserved because they are

used to differentiate the pathological tissue from the noise tissue. Therefore, there is always a tradeoff between noise removal and structure information. In the proposed optimization procedure, we have defined several cost functions to assess the quality of noise removal and structure information of an image. These cost functions are minimized using the SPEA2 algorithm. The proposed optimization procedure is applied to image denoising based on block matching and 3D collaborative filtering. The proposed optimization procedure provides a set of optimal solution rather than a single solution. This enables the designer to choose the best trade-offs. Experimental results showed that the proposed method greatly enhances the outcome of denoising methods. This method can be applied to many other denoising methods to enhance results. This method can be extended further by defining more cost functions and optimizing all the cost functions simultaneously.

BIBLIOGRAPHY

[1] M. Sonka and M. J. Fitzpatrick, *Hand Book of Medical Imaging.* SPIE Press, 2004, vol. 2.

[2] J. Beutel, H. L. Kundel, and R. L. V. Metter, *Hand Book of Medical Imaging.* SPIE Press, 2000.

[3] M. J. Ackerman, "The visible human project," *Proceeding of the IEEE*, vol. 86, no. 3, pp. 504–511, March 1998.

[4] Y. Shen, J. Seevinck, and E. Baydogan, "Realistic irrigation visualization in a surgical wound debridement simulator," in *Proceedings of Medicine Meets Virtual Reality 14*, vol. 119/2005, Long Beach, CA, 2006, pp. 512–514.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[6] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley, *Texturing and Modeling, A Procedural Approach*, Third ed. Morgan Kaufman, 2002.

[7] D. Knuth, *The Art of Computer Programming*, Third ed., ser. Seminumerical Algorithms. Addison-Wesley, 1997, vol. 2.

[8] T. Blu, P. Thevenaz, and M. Unser, "Linear interpolation revitalized," *IEEE Transactions on Image Processing*, vol. 13, no. 5, pp. 710–719, May 2004.

[9] E. Meijering, "A chronolgy of interpolation: From ancient astronomy to modern signal and image processing," *Proceedings of IEEE*, vol. 90, no. 3, pp. 319–342, March 2002.

[10] J. Schpok, J. Simons, D. S. Ebert, and C. Hansen, "A real time cloud modeling, rendering and animation system," in *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, San Diego, California, 2003, pp. 160–166.

[11] A. Cadussi, T. Hollerer, and N. Candussi, "Real time rendering of realistic trees in mixed reality," in *Proceedings of the International Symposium of mixed and Augmented Reality*, Vienna, Austria, 2005, pp. 204–205.

[12] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics, Principles and Practice*, Second ed. Addison-Wesley, 1997.

[13] R. A. Zoroofi, Y. Sato, S. Tamura, and H. Naito, "Mri artifact cancellation due to rigid motion in the imaging plane," *IEEE Transactions on Medical Imaging*, vol. 15, no. 6, pp. 768–784, December 1996.

[14] Q. Lu and T. Jiang, "Pixon-based image denoising with markov random fields," *The Journal of Pattern Recognition Society*, vol. 34, no. 10, pp. 2029–2039, October 2001.

[15] G. Gerig, O. Kubler, and R. Kikinis, "Nonlinear anisotropic filtering of mri data," *IEEE Transactions on Medical Imaging*, vol. 11, no. 2, pp. 221–232, 1992.

[16] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Second ed. Pearson Education, 2005.

[17] G. R. Arce, *Nonlinear Signal Processing, A Statistical Approach.* Wiley-Interscience, 2004.

[18] K. E. Barner and G. R. Arce, *Nonlinear Signal and Image Processing, Theory, Methods, and Applications*, ser. Electrical Engineering and Applied Signal Processing Series. CRC Press, 2003.

[19] G. R. ARce, "A general weighted median filter structure admitting negative weights," *IEEE Transactions on Signal Processing*, vol. 46, pp. 3195–3205, December 1998.

[20] K. Dabov, A. Foi, and K. Egiazarian, "Image restoration by sparse 3d transform-domain collaborative filtering," in *Proceedings of SPIE Electronic Imaging*, vol. 6812, San Jose, California, 2008, pp. 681 207–681 212.

[21] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 264–323, 1999.

[22] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, "Design and construction of a realistic digital brain phantom," *IEEE Transactions on Mecial Imaging*, vol. 17, no. 3, pp. 463–468, 1998.

[23] M. A. Anastasio, M. A. Kupinski, and R. M. Nishikawa, "Multiobjective genetic optimization of diagnostic classifiers with implications for generating reveiver

operating characteristic curves," *IEEE Transactions on Mecial Imaging*, vol. 18, no. 8, pp. 675–685, 1999.

[24] D. A. V. Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm test suites," in *Proceedings of ACM Symposium on Applied Computing*, San Antonio, Texas, 1999, pp. 351–357.

[25] M. A. Anastasio, M. A. Kupinski, and R. M. Nishikawa, "Optimization and froc analysis of rule based detection schemes using a multiobjective approach," *IEEE Transactions on Mecial Imaging*, vol. 17, no. 6, pp. 1089–1093, 1998.

[26] C. A. C. Coello, "An updated survey of evolutionary optmimization techniques: State of the art and future needs," in *congress on Evolutionary Computation*, vol. 1, Washington, D.C, 1999, pp. 3–13.

[27] J. Li, A. Huang, J. Yao, I. Bitter, R. Summers, P. Pickhardt, and J. Choi, "Automatic colonic polyp detection using multiobjective evolutionary technique," in *in Proceedings of SPIE*, California, USA, 2006, pp. 1742–1750.

[28] J. D. Schaffer, "Multiobjective optimization with vector evaluated genetic algorithms," in *Gentics Algortihms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Lawrence, Erlbaum, 1985, pp. 93–100.

[29] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multi-objective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 117–132, April 2003.

Ramu Pedada
1015 W, 50th Street, Apt # 8,
Norfolk, VA – 23508.
Phone: 757-581-6994
Email: ramupedada@gmail.com

EDUCATION

B.Tech, Jawaharlal Technological University ***Aug 2002 – May 2006***
Electronics and Communication Engineer

Master of Science, Old Dominion University ***Aug 2006 – Aug 2008***
Electrical and Computer Engineering
Thesis*: Medical Image Modeling and Processing,* Proposed two algorithms for biomedical imaging applications.

SKILL SET

Programming Languages: MATLAB, C, C++, VHDL, Latex, Assembly Language.
Simulation Tools: PSPICE, Multisim, Simulink.
Packages: MS Office, Adobe Photoshop, CorelDraw

PUBLICATIONS

- R. Pedada and Y. Shen, "Procedural Modeling of Wound Textures," *MODSIM World Conference & Expo*, (Virginia Beach, VA), September 2007.
- R. Pedada, Y. Shen, J. Li, and K. E. Barner, "Binary-Output Weighted Median Filter for Binary Signals," *2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV),* (Las Vegas, NV), July 2008
- S. Jakkula, V. Mantena, R. Pedada, Y. Shen, and J. Li, "Seasonal Adaptation of Vegetation Color in Satellite Images," *2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, (Las Vegas, NV), July 2008.
- V. Mantena, R. Pedada, S. Jakkula, Y. Shen, and J. Li, "Vegetation Identification Based on Satellite Imagery," *2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, (Las Vegas, NV), July 2008.
- Ramu Pedada, Yuzhong Shen "Automatic Image Generation for Wound debridement Simulation," ***ODU-NSU-EVMS-VTC Research Exposition Day: Research Expo 2007*** - 400 Years of Discovery, Ted Constant Hall, Norfolk, VA, (Poster), p. 38, April 05, 2007.
- Ramu Pedada, Yuzhong Shen "Fast Weighted Median Filter for Binary Signals," ***ODU-NSU-EVMS-VTC Research Exposition Day: Research Expo 2008*** - 400 Years of Discovery, Ted Constant Hall, Norfolk, VA, (Poster), p. 17, April 09, 2007. Won **Honorable Mention Award** for best paper.