3D image analysis with variational methods and wavelets : applications to medical image processing

Minh-Phuong Tran

PUCVL
Pôle Universités
Centre Val de Loire

UNIVERSITÉ D'ORLÉANS

UNIVERSITE D'ORLEANS

ÉCOLE DOCTORALE MATHÉMATIQUES, INFORMATIQUE, PHYSIQUE THÉORIQUE ET INGÉNIERIE DES SYSTÈMES

LABORATOIRE : Mathématiques - Analyse, Probabilités, Modélisation - Orléans

THÈSE PRÉSENTÉE PAR :

MINH-PHUONG TRAN

soutenue le : **28 Septembre 2012**

pour obtenir le grade de : **Docteur de l'université d'Orléans**

Discipline/ Spécialité : **MATHÉMATIQUES**

Analyse d'images 3D par méthodes variationnelles et ondelettes - Application à l'imagerie médicale

3D image analysis with variational methods and wavelets - Applications to medical image processing

THÈSE DIRIGÉE PAR :

Maïtine BERGOUNIOUX	Professeur, Université d'Orléans
Renaud PÉTERI	MCF, Université de La Rochelle

RAPPORTEURS :

Valérie PERRIER	Professeur, Université de Grenoble
Christian GERMAIN	Professeur, Université de Bordeaux

JURY :

Romain ABRAHAM	Professeur, Université d'Orléans, Président du jury
Maïtine BERGOUNIOUX	Professeur, Université d'Orléans
Jean-Claude BELOEIL	DR, CNRS Orléans
Christian GERMAIN	Professeur, Université de Bordeaux
Valérie PERRIER	Professeur, Université de Grenoble
Renaud PETERI	MCF, Université de La Rochelle
Vincent TORRI	MCF, Université d'Evry

Acknowledgements

It would not have been possible to write this PhD thesis without the help and support of the kind people around me. It is a true pleasure for me to thank the people who have made this thesis possible.

First and foremost of all, I would like to thank my director and supervisor, professor Maïtine Bergounioux, for giving me the chance to accomplish my PhD thesis at the University of Orléans and for supporting me in this long journey. Her extraordinary leadership and research expertise constantly pushed me to go beyond my limits. In addition, she was always accessible and willing to help, so the research life became smooth and rewarding for me.

I owe my deepest gratitude and respect to my co-director Renaud Péteri who has shown his support in many different ways. Thanks for all our passionate discussions about my thesis, for all the time dedicated to explain me the smallest details, for all the meticulous effort to improve every small, single paper and presentation. Thank you forever for giving me your generous support, invaluable guidance and inspiration all along the time it took to accomplish this thesis.

I am also truly thankful for professor Vincent Torri, who has been patience to help me to understand problems step by step, whose letters gave me important guidance during my first steps into the thesis subject. I appreciated his guidance and helpful discussions. His remarks in the reaction of this thesis and several reports in the past, his fruitful comments and suggestions are very meaningful for me, too.

I would like to express my sincere appreciation to the financial support from the University of Orléans, this support enabled myself to do extensive research abroad, without which this thesis would have never been possible.

I owe big gratitude to my thesis reviewers that sent feedback with most interesting suggestions about the original manuscript. A special thank to their precious time dedicated to reviewing the manuscript despite their numerous constraints.

I am highly grateful for all the priceless time of Dr. Loïc Piffet in the laboratory MAPMO, who provided me in order to be able to accomplish the second-order variational model, without which our methods and codes would have been meaningless.

It has been an honor for me to have the special MRI medical images from members of laboratory CBM: Sandra Même, J.C. Belœil and F. Szeremeta. The presence of these images has been made my thesis works are more meaningful.

In the summer of 2010 and 2011, I had a very inspiring time staying at the laboratory MIA in the university of La Rochelle, France. I mainly thank to all the studious and friendly atmosphere there, a warm thank to Renaud Péteri, Sloven Dubois, Ghina El Mir, José Mennesson and all the others! It has been an honor for me to be able to spend several weeks here.

Many thanks to all members of laboratory MAPMO, who have accompanied with me during a long three years of working and studying here. A special thank to secretaries, Mrs Anne Liger, Marie-France Grespier, Marie-Laurence Poncet and

Marine Cizeau, who always present when I need help. I am also grateful Romain Theron for his understanding and help concerning all the informatic and administrative issues.

In my daily work I have been blessed with a friendly and cheerful group of fellow students. I was very fortunate to have Nhan Nguyen, Thuy Nguyen, Nga Nguyen, Ly Tran, Tam Phung, Lan Tran, Hai Ong, Liem Nguyen and all other Vietnamese students, who have made my PhD incredibly interesting and fun. We have shared a lot of good memories starting with our running sessions to overcome the most difficult stages, when I was sick and depressed. I have learnt a lot from them, and moreover I am forever grateful for every moment that felt so hard to live through student life.

Spending 3 years in Orléans, it has been a honor to know the other PhD students, Loïc Piffet, Olivier Vitry, Tamara El Bouti, Sebastien Dutercq and all my officemates. Their valuable appearance and friendship helped me a lot, we have shared and discussed together about works, life, future... It was a real gift for me, even after they have left the laboratory, their helps in the past are still really memorable to me.

I also would like to thank my family for all their love and encouragement. I owe a tremendous amount of gratitude to my family, who never understood what I am doing, but was always proud of me! I would like to express my graduate to my father, who raised me with a love of science and supported me in all my pursuits. For my beloved mother and little sister, who always listened to me everytime I need advices.

Finally, I would like to dedicate this thesis to my fiance, is also a mathematician, who was always by my side, helped and gave me advices, on both mathematics and life. I send all my grateful thanks for his patience, support and continued enthusiasm throughout the work. Without him, the achievement of this thesis would have never been possible.

I am looking forward to the new coming challenge.

— Minh-Phuong TRAN
University of Orléans
February 7, 2012.

Abstract

Medical procedures have become a critical application area that makes substantial use of image processing. Medical image processing tasks mainly deal with image restoration, image segmentation that bring out medical image details, measure quantitatively medical conditions etc. The diagnosis of a health problem is now highly dependent on the quality and the credibility of the image analysis. The practical contributions of this thesis can be considered in many directions for medical domain.

This manuscript addresses a 3D image analysis with variational methods and wavelet transform in the context of medical image processing. We first survey the second-order variational minimization model, which was proved that better than the classical Rudin-Osher-Fatemi model. This method is considered in problems associated to image denoising, image segmentation, that makes a short state of the art on medical imaging processing techniques. Then we introduce the concept of wavelet transform and present some algorithms that also used in this domain. Experimental results show that these tools are very useful and competitive. The core of this research is the development of new 3D representations, which are well adapted to representing complicated medical data, and filament structures in 3D volumes: the cerebellum and mice vessels network.

Each of these two based methods has advantages and disadvantages, we then propose a new modified model that combines these schemes in the rest of the thesis. In this situation we propose a new modified model that combines these schemes. With the new decomposition model, in the reconstructed image, noise can be removed successfully and contours, textures are well preserved. This leads to further improvements in denoising performance.

Finally, the further part of the thesis is devoted to the description of contribution to extend some classical contour closing methods, namely hysteresis thresholding and contour closing based on chamfer distance transform, in the 3D context. The thesis concludes with a review of our main results and with a discussion of a few of many open problems and promising directions for further research and application.

Résumé

L'imagerie médicale joue un rôle de plus en plus important avec le développement de nombreuses techniques d'acquisition. Il faut principalement pouvoir restaurer (débruiter) les images et en faire une segmentation. Ainsi toute l'information qualitative et quantitative sera disponible pour affiner les diagnostics. Dans cette thèse nous proposons une contribution à cette analyse dans un contexte 3D.

Nous étudions deux grands types de méthodes : les méthodes variationnelles et les méthodes par ondelettes. Nous commençons par présenter les modèles variationnels du second ordre, qui s'avèrent plus performants que la classique méthode du premier ordre de Rudin-Osher-Fatemi. Nous l'utilisons pour débruiter et segmenter après avoir donné un bref état de l'art des procédés d'acquisition des images en médecine. Nous introduisons ensuite la transformée en ondelettes et présentons des algorithmes basés sur cette méthode. Les résultats numériques montrent que ces méthodes sont performantes et compétitives.

Le cœur de notre travail est de développer des rerésentations 3D qui sont bien adaptées à des données médicales complexes comme des images IRM sous échantillonnées, peu contrastées (cervelets de souris) ou des images IRM d'angiographie (cerveaux de souris).

Chaque technique a ses avantages et ses inconvénients. Aussi nous proposons un modèle variationnel mixte second ordre / seuillage par ondelettes. Ce modèle se comporte particulièrement bien : le bruit est correctement éliminé et les contours et textures préservés.

Pour finir, nous adaptons plusieurs méthodes de fermeture de contours (hystérésis et distance de chanfrein) dans un contexte 3D. Le mémoire se termine par une synthèses des résultats et une présentation de futures directions de recherche.

Contents

Chapter 1

Introduction

In five senses that consist of sight, hearing, touch, smell and taste, which humans use to perceive their environment, the vision is the most powerful. In fact, more than 99% of the human brain is involved in processing images from the visual cortex, [22]. A visual image is rich in information. Therefore, images play a very important role in handling, storage data and they are applied in almost fields of life.

Nowadays, the medical industry, astronomy, physics, chemistry... are just some of the many fields that rely upon images to store, to display, and provide information about the world around us. In addition, the use of images is necessary for communication, transmitting information, and also for creation ... The challenge to scientists (biomedical researchers, physicians, mathematicians), engineers and business people is to quickly extract valuable information from raw image data. This is the primary purpose of image processing - converting images to information.

Image enhancement is especially important in medical imaging because it allows physicians to obtain a better visual interpretation, especially when viewing small structures (e.g. thin vessels), and in looking for other numerous clinical applications. Medical imaging is the technique and process used to create images of the human body for seeking to diagnose or examine disease or the study of anatomy. Such procedures are not usually referred to as medical imaging, but rather are a part of pathology.

Medical images have a number of different dimensionalities (two, three, four - dimensional), where three-dimensional, can be considered as an image stack of 2D slices or directly spatial three-dimensional volume data. On the other hand, 3D can be seen as a video which is composed from a lot of 2D slices that change every time unit Δt. We could compose a video as 2D+T model. Video refers to recording, manipulating, and displaying moving images, especially in a format that can be presented on a camera or device.

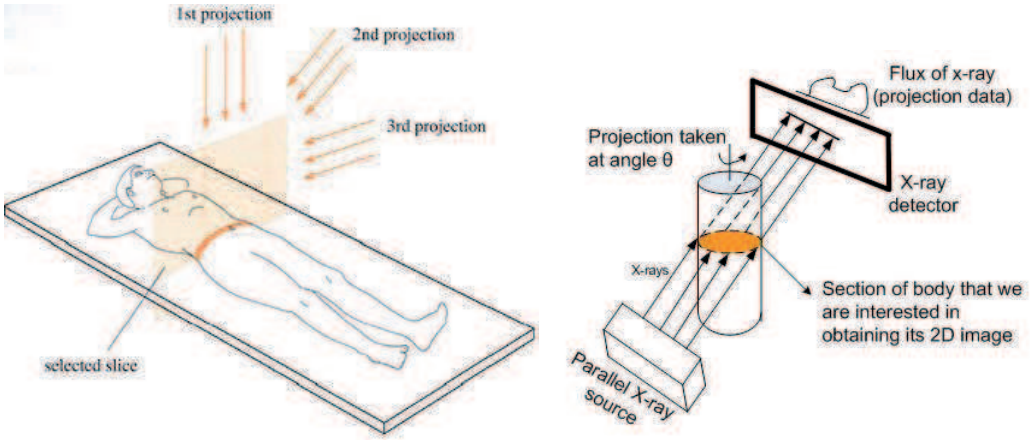
Here we pay attention that the 2D+T video model is different from the whole directly spatial 3D volume data. Indeed, 2D+T video presents a visualization that a sequence of images representing scenes in motion every time step Δt, for the video viewing a lot of images move very quickly so we can see that the objects in images change during video playing. Contrary to the video model, the 3D data viewer presents objects in reality at every direction. In short, 2D+T video is the spatio-temporal viewing and direct 3D data can be considered in spatial viewing. And in this thesis, we focus on the 3D viewer of data, the typical 3D data set is a group of 2D slice images acquired by a computed tomography, magnetic resonance imaging or micro computed tomography scanner.

With the growth of computing power and image technology, medical imaging has greatly influenced the medical field. The diagnosis of a health problem is now highly dependent on the quality and the credibility of the image analysis. Medical imaging is considered as a part of biological imaging, which have been developed since 19^{th} century onwards. In 20^{th} century, the mathematical principles behind tomographic reconstruction have been understood, X-ray computed tomography (CT) have been developed. Nuclear magnetic resonance has been using for imaging in magnetic resonance imaging (MRI).

CT and/or MRI scans are specialized types of X-rays and/or magnetic field. The scanners of both look like a large doughnut. Patient lies down on a moveable bed that slides into the center of magnet. The bed moves slowly backwards and forwards to allow the scanner to take pictures of the body, it does not touch the patient. Inside the tunnel, computer collects the results and these results are translated into images that look like a "slice" of a person.

Computed Tomography (CT)

Computed Tomography (CT) scans are used to image a wide variety of body structures and internal organs, it is great to look at the bones. CT scans uses multiple X-rays to create cross-sectional pictures of the body. While traditional X-rays image organs in two dimensions, with the possibility that organs in the front of the body are superimposed over those in the back, CT scans allow for a more three-dimensional effect , [62], [13].



(a) X-rays projections of CT on the body. Thanks for the image from http://labspace.open.ac.uk.

(b) Parallel X-ray CT system.

Figure 1.1: CT systems working on a body. Many projections are obtained, then they are combined in software using filtered backprojection to obtain an accurate 2D image of the section of the body.

The camera in CT machine moves around inside and takes pictures, it may make some noise because of the movement during the scan.

From the mathematical point of view, CT can be considered as an inverse problem which are related to Radon transform (RT), Discrete Fourier transform (DFT) and inverse DFT [47].

In CT, thin X-ray parallel beams are transmitted across a section of the body at a specific angle θ. When the beams end at the other side of body, we have the representation of projection of the cross section at the angle θ, as in figure 1.1(b). Next, the angle θ increases and another projection is obtained. Repeating this process we will get the sequence of projections. This sequence of projections is then used to reconstruct a 3D image of that section of body.

Magnetic Resonance Imaging (MRI)

Magnetic Resonance Imaging scans are used for looking at soft tissue. The MRI scan uses magnetic and radio waves, there is no exposure to X-rays or any other damaging forms of radiation. In the magnetic field, the effects of radio waves through the body cause body's tissue to resonate and take high-resolution pictures of your bones and soft tissues. One of the great advantages of MRI is the ability to change the contrast of the images. Moreover, MRI is able to change the imaging plane without moving the patient, MRI machines can produce images in any planes [32], but the results may be noisy.

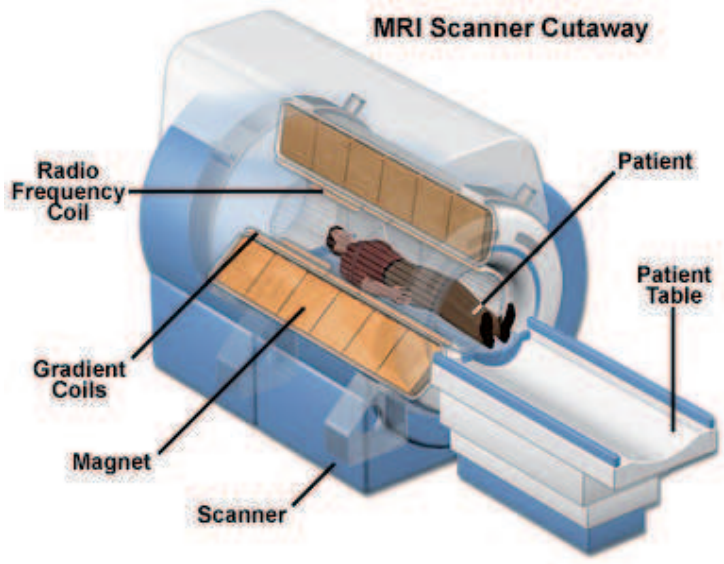


Figure 1.2: MRI scanner cutaway, thanks for the image from [34].

One of the strengths of MRI compared to other imaging techniques (optical, PET, CT,...) is the ability to obtain not only anatomical images in three dimensions but also functional information with the implementation of new method of imaging (fMRI, diffusion, perfusion,...), [19]. For this reason, MR images will be applied in our studied.

How does a MRI scan differ from a CT scan?

Because the MRI creates a magnetic field around person and the pulses radio waves to the area of body to be pictured, MRI scan makes possible to take pictures from almost every angles, whereas CT scans only shows pictures horizontally. While CT scan is obtained using X-rays, the radiation is passed through the body and received by a detector then integrated by a computer to obtain a cross sectional image that is displayed on the screen, there is no X-rays involved in producing an MRI scan. MRI most commonly used in radiology in magnetic field to visualize detailed internal structure and limited function of the body, so MRI scans are generally more detailed than CT scanner.

CT can outline bone inside the body very accurately, and MRI is more versatile than X-rays, it is used to examine a large variety of medical conditions. MRI is much higher detail in the soft tissues comparing with CT while CT provides good details about bony structures more than MRI. Moreover, the time taken for total testing and the cost range of CT scans are less than MRI scans.

CT and MRI scans are very important tools in visualizing body parts, especially in clinical evaluation and treatment planning, CT and MRI scanners can be used to create a volume by reconstructing a series of slicing images. CT and MRI scans depend on what need to be visualized and the reason we need for the test. For instance, in figure 1.3 below the normal and abnormal pericardium seen on CT and MRI studies:

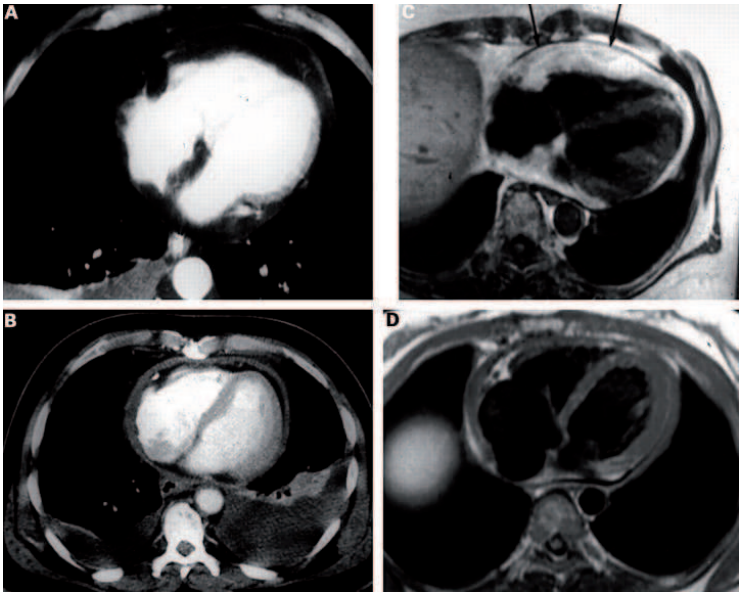


Figure 1.3: Normal and abnormal pericardium seen on computed tomography (CT) and magnetic resonance imaging (MRI) studies. (A) CT scan of normal pericardium. (B) CT scan of thickened pericardium. (C) MRI scan of normal thick pericardium (arrows). (D) MRI scan of a thickened pericardium. Images are reproduced from Breen on [29].

The 3D MRI scans of Mouse Brain and Vessels of Mouse Brain

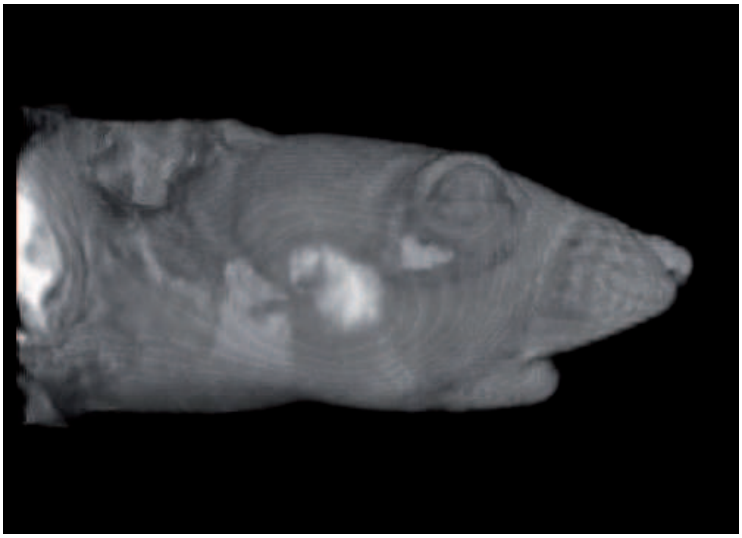
Mouse is one of the major animal models for biomedical research. Mouse models are helpful to link genes, since 80% of the genome is common with that of human being, [19]. The development of magnetic resonance imaging (MRI) on mouse brain helps to study human diseases. Here we investigate small animals which are completely embedded in the magnetic field, namely, mice whose teeth are tied hanging up and put on the magnetic field. The MRI camera takes pictures of the whole mouse's body. Biologists here want to consider MRI scans of the mouse brain without killing the animal. Human intervention will be limited to minimum and the manual handling will be greatly reduced.

The focus of the thesis is based on three-dimensional visualization of MRI scans, specially the brain and vessels of transgenic mice brain. In Figure 1.4 we give these two MRI examples, and moreover, we present arbitrary slices along xy, yz and xz cross directions on each volume data in Figure 1.5.

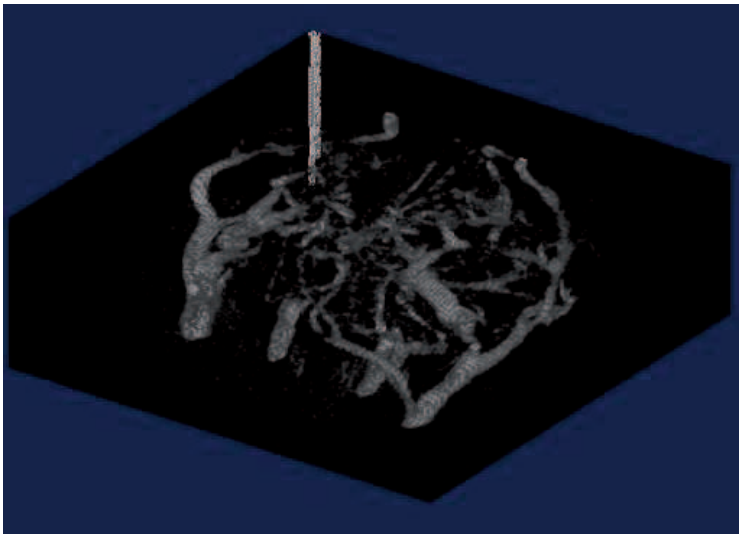
The mouse brain volume is the stack of 110 MR images which size is 341×110 pixels. The vessels of mouse brain volume is the stack of 51 MR images which size is 256×256. We thank J.C. Belœil, S. Même and F. Szeremeta, from CBM Laboratory in Orléans, for the use of these two 3D volumes. Segmenting the mouse brain 3D data is a difficult issue since the contrast between different objects is low. For the vessels of mouse brain data, biologists want to recover the network of filament structures, specially the small ones inside the noisy volume. In detail, we want to know how to recover the real blood-vessels network (without noise). The problem is to identify the noise, since thin structures can be considered as noise. Images have to be positioned and oriented relative to one another and aligned exactly so that vessels are continuous through slices. For such volume data, we want to consider medical image processing from the mathematical point of view.

A very large part of image processing is devoted to image restoration. Image restoration is the removal or reduction of unexpected elements that are involved while the image is obtained. One of the main objectives in image restoration is denoising and/or texture extraction. It is very difficult to diagnose a particular disease for biomedical researchers because of noise that pollutes images. The noise that degradate our image cannot be easily removed here. With noise statistical property and frequency spectrum distribution rule actually calculated according to image's characterization, there are many methods to removing noises. Let us mention, for example, PDE smoothing filters, total variation filter, adaptive filters (high or low pass filters). We also have frequency domain filters or those using Wavelets, Ridgelets, Beamlets and so on. For the denoising and/or texture identification tasks, the goal is to remove noise and/or spurious details from a given corrupted image but still maintain its essential features.

The general idea behind variational denoising methods is to regard a noisy image f as being obtained by corrupting a noiseless image u. The desired image u is then a solution of the corresponding inverse problem: which u could f be obtained from



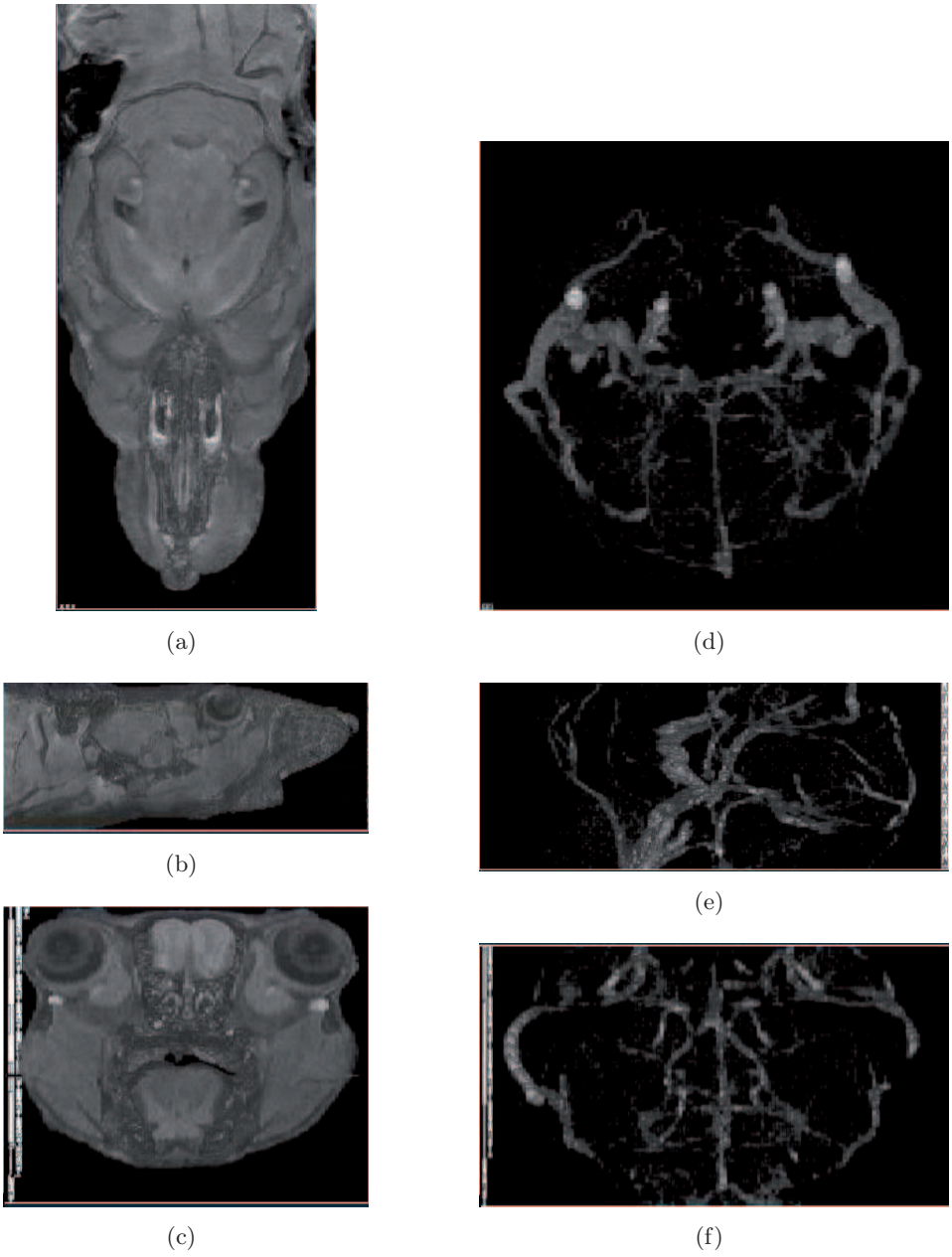
(a)



(b)

Figure 1.4: (a) Mouse brain, (b) Vessels of mouse brain.

by corruption? Moreover, because the classical filtering methods, such as high/low pass filters, heat equation convolution ... do not preserve the contour shapes in outcome image, another variational model may be considered. A classical approach of image restoration consists in considering that image f can be decomposed into two components $u+v$, where the first component u is well structured and represents a simple geometrical information, the second component v containing the oscillating patterns (which usually both textures and noise). The regularization term involves only the homogeneous component u, while the remaining term $v := f - u$ models the noise that need to be minimized. These models were studied extensively the



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

Figure 1.5: Left: Brain of Mice volume, Right: Vessels of Mice Brain volume, (a-d) xy cross direction, (b-e) yz cross direction and (c-f) xz cross direction.

past years. The general form is to minimize a functional energy:

$$\mathcal{F}(u) = \|f - u\|_X + \mathcal{R}(u), u \in Y \subset X \tag{1.1}$$

where X, Y are (real) Banach spaces and $\mathcal{R}$ is a regularization operator.

One of the most successful algorithms is the Rudin-Osher-Fatemi (ROF) model, [4], [50], [5] which uses total-variation regularization. Regard to this model, the

observed image f contains two parts:

$$f = u + v$$

where v represents the noisy component ($v \in L^2(\Omega)$) and u is the geometrical component which models the homogeneous objects presented in the image. It usually belongs to $BV(\Omega)$ - the space of bounded variation function.

In the ROF model we consider the functional energy:

$$F(u) = \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV(u,\Omega), u \in BV(\Omega); \tag{1.2}$$

where $TV(u)$ stands for the total variation of $u \in BV(\Omega)$. We look for a solution to:

$$\inf_{u\in BV(\Omega)} F(u). \tag{1.3}$$

This problem has a unique solution in $BV(\Omega)$. However, the ROF model has some inconveniences: the use of BV norm implies perturbations. The computed solution turns to be piecewise constant which is called the *staircasing effect* ([11]). Artificial contours that do not exist in the original image appear. This *staircasing effect* is presented in detail of Appendix B. Therefore, though noise can be successfully removed the solution is not satisfactory. This variational model can be improved using different functional spaces. In [41] it is proposed to used the functional space of second order bounded variation - the BV^2 space.

$$\inf_{u\in BV^2(\Omega)} F(u); \tag{1.4}$$

where

$$F(u) = \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV^2(u,\Omega), u \in BV^2(\Omega). \tag{1.5}$$

In this thesis, we generalize the ROF model to the functional space BV^2 in three-dimensional framework. Mouse brain MRI images segmentation (Figure 1.4) will be considered as an application. Besides variational methods, there also exist many other techniques, for instance wavelet analysis or multifractal analysis ...

Each technique has its advantages and disadvantages, and wavelet technique is one of recent approaches. In the thesis, this second-order variational method and wavelet transform scheme are studied and compared.

The outline of this thesis is organized as follows:

Chapter 2

This chapter discusses variational models from the classical Rudin-Osher-Fatemi (ROF) model to the second order model (ROF2), that deals with all disadvantages of problem ROF. We first present theoretical definitions of the first TV and second-order total variation TV^2, and describe the functional framework (BV and BV^2

spaces). In [59], Bredies K. et al. gave another functional concept of *total generalized variation* TGV and its second-order (TGV^2): The new regularization term was then replaced to consider the other variational problems, that also overcome some inconveniences problem (P_{ROF}) brings us efficiently. In this chapter, the comparison between theoretical definitions of TV^2 and TGV^2, between different variational problems are also studied in the finite dimensional case.

Chapter 3

In this chapter, we especially deal with the three-dimensional models that have been generically described in the previous chapter. The so-called isotropic model is performed in details with the fixed point algorithm proposed by Chambolle A. [4] and we also investigate the Nesterov type algorithm to speed up the method, see Weiss P. [23], Piffet L. [41].

As previously mentioned, the observed image f can be divided into a regular part $u \in BV^2(\Omega)$ and an oscillating part $v := f - u$. Observing solutions of problem ROF2, we note that the model keeps geometrical information and removes almost noise as well. However, a lot of geometrical information remains together with the texture and noise in the oscillating part v.

In particular, the remainder term $v := f - u$ contains together texture and/or noise and a part of contour shapes. In this sequel, our goal is to locate texture and/or noise and we do not need to work with the cartoon part u any longer. To perform a full texture extraction, we hope to keep most of texture and noise in the oscillating part while many contour shapes disappear. Based on Piffet L. [40], a local anisotropic model is then performed. Moreover, we also sketch out in this chapter two algorithms from Chambolle A. [4] and Nesterov [67], [23] and present some MRI numerical tests in 3D.

Chapter 4

Wavelet analysis has been developed during the two past decades. Wavelets are used successfully in many applications such as signal analysis, image analysis, communications systems, and other signal processing applications. The flexibility of wavelets makes them appropriate for many special purposes. Wavelets are widely described by Daubechies I. [26], Mallat S. [53], [54], [3], [8] and so on.

Among various denoising techniques, wavelet denoising is one of the most popular ones, using multiresolution representation and noise separation features, [51]. Roughly speaking, wavelet coefficients are calculated via a wavelet transform, and it is possible to denoise by thresholding their coefficients. Image is subjected to the wavelet transform, then the wavelet coefficients are found, the components with coefficients below a threshold are replaced with zeros, and the image is then reconstructed (inverse discrete wavelet transform).

One and two dimensions image texture extraction procedure can be further generalized to three dimensions. In 3D wavelet transform decomposition, a dataset vol-

ume is decomposed into eight subbands occupying eight octants in the 3D wavelet space. Inside the wavelet filter banks, the high-pass subbands collect the noise and irregular features while the low-pass subbands contain the trend or approximation of the original data, which removes or reduces noise. Therefore, throughout this chapter, we are studying 3D wavelet transform in image restoration by using filter banks for medical imaging applications. Besides the study of subsampled wavelet transform (Mallat's algorithm), we also give an understanding to undecimated wavelet transform (called "à trous" algorithm) therein.

In this chapter, we study three-dimensional discretized wavelet transform that includes wavelet decomposition and reconstruction strategies. In addition, it gives some hints for 3D image denoising and contour detection applications that will be considered in chapters 5 and 6.

Chapter 5

This chapter 5 is devoted to numerical tests for 3D image denoising and texture extraction by both mathematical points of view: variational model and wavelet technique discussed previously. In this chapter we consider how these methods behave on 3D examples of video and MR images. Besides, the chapter also discusses about texture extraction. It is an important task in many computer applications of image detection. During denoising process, we do not want to loose some useful information, that's why the local anisotropic algorithm is performed to determine the texture and/or contour information.

In the second part of the chapter, we shall discuss about the wavelet transform. As previously studied in chapter 4, we have two different types of wavelet transform: Mallat's subsampled wavelet and the "à trous" unsubsampled wavelet transform. Chapter 5 deals with detailed numerical performances for 3D volume examples. In addition, we thank to [51] for the 3D wavelet proposed method that combined 2D and 1D wavelet transform together in the later work on 3D image denoising. This chapter also gives us the 3D denoising representation with fully automated at a high speed of this 2D+1D wavelet technique.

To illustrate each method, we present some numerical results tested on the experimental examples arising in biomedical imaging. It is also provided a comparison between considered methods, analysis of some experimental results and some evaluations, some conclusions about our approaches.

Chapter 6

Chapter 6 addresses the application to 3D image **segmentation** process that is performed on a 3D image. In particular we locate the contour information of image with both variational model and wavelet pyramid method. From a different point of view, the contour detection is another application to the denoising application in chapter 5. Here, we deal with contour detection simulations and compare some experimental results by variational method and wavelet scheme. In particular, the

difference between isotropic and local anisotropic results represent contour shapes in variational method, on the other hand, the contour detection methods using wavelet transform scheme is also studied to isolate the contour shapes of image, [55], [36], [17].

Visual perception evolved in a world of objects many of which are bounded by smooth closed contours. Especially with the three-dimensional dataset that need to be marked the locations of filaments to get the region of interest. There also exist other structures with high values (noise) or we may loose some information within the vessels through contour detection process. That is the reason why we should perform contour closing algorithms. Once the contour shapes are detected, we next provide methods of contour closing implementation. This chapter also presents two classical methods, which are applied to three-dimensional contour closing, the hysteresis thresholding [48] and a local operator based on chamfer distances methods [35], [9], [6].

Chapter 7

Throughout previous chapters, we have studied two based methods and their applications in image implementation, the second-order variational method and wavelet transform simulated on three dimensional medical images. Because each method has the advantages and disadvantages, BV^2-variational method gives good denoised results meanwhile the wavelet shrinkage technique reduces well edge/contour artifacts. In this situation we propose a new modified model that combines these schemes, we then have a reconstructed image that has fewer oscillations near edges and noise is smoothed. A lot of references, for instance in [24], [25] etc, gave ideas of denoising algorithm based on a combination of these frameworks several years ago. It permits us to improve another approach that employs variational framework, in particular the minimization model (P_{ROF2}) plus the wavelet thresholding to reduce oscillations, remove noise while maintain the sharpness of image.

The ideas introduced here can be considered to perform in this chapter and use as a post-processing technique for image denoising application. Taking advantages of both methods (variational method and wavelet based scheme) mentioned previously, we apply them in a new model that combines two based methods; the numerical results are well denoised and almost geometrical details are well preserved. Therefore, here we propose a new small variational model for image denoising and decomposition (so-called the "merged-problem"), which combines the second-order total variational model of Rudin-Osher-Fatemi (P_{ROF2}) and the Wavelet transform.

In [66], Y. Meyer has introduced an image decomposition model to split an image into two components: a geometrical components and texture (oscillatory) component. Inspired by his work, numerical models have been developed to carry out the decomposition of gray scale images. Other recent and related image decomposition models combined from variational model and wavelet transform are well proposed. In our approach, the considered model will decompose an image into several components, where each of them is characterized by a special space of function. More

precisely, the aim of work in this chapter is to construct a model which decompose image into two components: the first one represents all geometrical information of the image, the second one containing the oscillating structures of image. In the rest of this chapter, we present some numerical results applied both (P_{ROF2}) and an undecimated wavelet transform (the "à trous" algorithm) in order to deal with the case of 3D noisy images.

Chapter 8

As previous description throughout this introduction part, we have given a small introduction to the three-dimensional MRI model of vessel with the important role in diagnosing diseases via trisomic mouse brain (cerebellum). Indeed, the quantitative measurement of three dimensional vessel attributes as defined from high resolution, MR images thus provides a new method of evaluating many types of disease. In the rest of this thesis, we dedicate a special chapter 8 to describe and give a number of issues surrounding this model. It can be said that this is a special image with the difficulty on the difference in the contrast between the model objects, for instance the contrast between vessels and the background is low, or inside vessel regions can vary from region to region. The vessel's attributes include not only vessel number but also vessel morphological measures such as filament structure, tortuosity and branching pattern etc. Moreover, the obtained MRI model captured during scanning process contains a lot of noise, therefore it is necessary to apply some methods of denoising while retain almost filament structures of vessel.

This chapter aims to justify our mathematical application and performance only on the Vessels of mouse brain. This volume is a difficult issue because the original datum we received from biologists is very noisy during MRI scanning. Biologists want to recover the network of filament structures, specially the small filaments inside a noisy volume, which namely, we just want to know how to differentiate the real network of this model without noise, the problem is to distinguish where the noise is, where the real filament structures are. The noise of our image does not belong to any known classification noise (Gaussian, Salt and Pepper etc), it is a challenge to evaluate the results after noise reduction by our applied methods. In addition, it is the difficulty when we did not have any standard compare to the original quality image (for example, we already have such as Lena, Barbara or Cameraman image in two-dimensions). The presence of noise not only produces undesirable visual quality but also lowers the visibility of low contrast objects. In such situation it is very difficult for biomedical researchers to diagnosis the particular disease. Noise removal is essential in medical imaging applications in order to enhance and recover fine details that may be hidden in the data, that is reason why reducing random noise is a very active research area in medical image processing. Additionally, noise reduction methods developed in other research fields find their usage in biomedical applications, noise reduction must be carried out with extreme care to avoid suppression of the important image content. For the volume of vessel data, it is truly a challenge to highlight structures such as mouse brain vessels that otherwise would

be difficult to delineate from their surroundings, the results of biomedical image denoising should be consulted with medical experts.

The chapter sketches out the order of work as follows: image denoising performance can be considered, we then applied the segmentation process to the vessels based on some known methods that have been considered in previous chapters. Since the given data of vessel contain noise, we firstly perform variational or wavelet based methods to reduce noise without the appearance of staircase effect and contour shapes are well retained in the numerical results. In our experience, some proposed schemes introduced in chapter 6 could be applied to detect and close contours to get the full systems of vessel fibres. They aim to facilitate the visualization of vessel model by recovering volume prior to 3D reconstruction. Indeed, segmentation of vessels is one of essential medical computing tools for clinical assessment of human diseases, therefore it is challenging to perform this kind of image segmentation in angiography.

Finally, the image decomposition model ($\mathcal{P}$) is also performed to the vessel volume data from the theoretical studying with some examples are implemented and compared in the chapter 7. In the rest of this chapter, we present some experimental results applied to the vessels image, in which the method gives very satisfactory results of separation into geometrical objects and the oscillating component. This process also confirms that our approach is well adapted to the 3D image of vessels data, one of the most difficult issues in bio-medical research. The better numerical results will be obtained with a more expensive computational work in the promising future.

Conclusions and Further Works

Finally, we end the thesis with Conclusions and some of further works, where we discuss some open questions related to subject matter. Further work is also analyzed suggesting various ways through which the research may continue.

Chapter 2

Variational models

Contents

This chapter discusses variational models from the classical Rudin-Osher-Fatemi (ROF) model to the second order model (ROF2), that deals with all disadvantages of problem ROF. We first present theoretical definitions of the first TV and second-order total variation TV^2, and describe the functional framework (BV and BV^2 spaces). In [59], Bredies K. et al. gave another functional concept of *total generalized variation* TGV and its second-order (TGV^2): The new regularization term was then replaced to consider the other variational problems, that also overcome some inconveniences problem (P_{ROF}) brings us efficiently. In this chapter, the comparison between theoretical definitions of TV^2 and TGV^2, between different variational problems are also studied in the finite dimensional case.

2.1 Functional framework

2.1.1 The first order bounded variation space

Definition 1. [12] *Let $\Omega \subset \mathbb{R}^n$ be a generic open set and bounded, for $(n \geq 2)$, $u \in L^1(\Omega)$. We say that u is the function of bounded variation in Ω if the weak derivative in sense of distribution $D_i u$ of u a finite Radon measure, which namely, if:*

$$\int_\Omega u \frac{\partial \varphi}{\partial x_i} \mathrm{d}x = -\int_\Omega \varphi \mathrm{d}D_i u, \quad \forall \varphi \in C_0^\infty(\Omega; \mathbb{R}^n); i = 1, 2, ..., n \tag{2.1}$$

for some $\mathbb{R}^n$-valued measure $D_i u = (D_1 u, D_2 u, ..., D_n u)$ in Ω. The vector space of all functions of bounded variation is denoted by $BV(\Omega)$.

Definition 2. [12] **Total variation**
Let $u \in L^1(\Omega)$. The total variation of u in Ω, which be denoted by $TV(u,\Omega)$, is defined by:

$$TV(u,\Omega) = \sup\left\{ \int_\Omega u \,\mathrm{div}\varphi \,\mathrm{d}x : \varphi \in \mathcal{C}_0^1(\Omega;\mathbb{R}^n), \|\varphi\|_\infty \leq 1 \right\}. \tag{2.2}$$

where $\mathrm{div}\varphi = \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{\partial \varphi}{\partial \mathrm{x_i}}$.

Definition 3. [41] **Bounded Variation space**
The space $BV(\Omega)$ of bounded variation functions is defined as:

$$BV(\Omega) = \left\{ u \in L^1(\Omega) : TV(u,\Omega) < +\infty \right\}. \tag{2.3}$$

Proposition 1. [12] *The derivation in sense of distribution of every $u \in BV(\Omega)$ is a bounded Radon measure, which be denoted Du, we have:*

$$TV(u,\Omega) = |Du|(\Omega), \ \forall u \in BV(\Omega). \tag{2.4}$$

And its value on an open set $U \subseteq \Omega$ is:

$$TV(u,U) = |Du|(U) = \sup\left\{ \int_U u\mathrm{div}\varphi \mathrm{d}x : \varphi \in \mathcal{C}_0^1(U;\mathbb{R}^n), \|\varphi\|_\infty \leq 1 \right\}. \tag{2.5}$$

Theorem 1. [41], [39] **Banach properties**
The space $BV(\Omega)$, endowed with the norm $\|u\|_{BV} = \|u\|_{L^1} + \Phi_1(u)$ is a Banach space, where $\Phi_1(u) = |Du|(\Omega) = TV(u,\Omega)$.

Then, we give some useful properties of the functional bounded variation space as in theorem below:

Theorem 2. [12] *Assume that $u \in BV(\Omega)$. There exists a sequence of functions $u_i \in C^\infty(\Omega) \cap BV(\Omega)$ such that:*

(i) $u_i \to u$ in $L^1(\Omega)$ as $i \to \infty$;

(ii) $|Du_i|(\Omega) \to |Du|(\Omega)$ as $i \to \infty$, and $TV(u_i,\Omega) \to TV(u,\Omega)$ as $i \to \infty$;

Moreover,

(iii) if $u \in BV(\Omega) \cap L^q(\Omega)$, $q < \infty$, we can find functions u_i such that $u \in L^q(\Omega)$ and $u_i \to u$ in $L^q(\Omega)$;

(iv) if $u \in BV(\Omega) \cap L^\infty(\Omega)$, we can find functions u_i such that $\|u_i\|_\infty \leq \|u\|_\infty$ and $u_i \to u$ in $L^\infty(\Omega)$ - weakly.*

Theorem 3. [12] **Semi-continuous Property**
Suppose that $u_i \in BV(\Omega), i = 1,2,...$ and $u_i \to u$ in $L^1_{loc}(\Omega)$. Then,

$$|Du|(\Omega) \leq \liminf_{i\to\infty} |Du_i|(\Omega). \tag{2.6}$$

Theorem 4. [41], [12] **Embedding Theorem**
Let Ω be open and bounded subset of $\mathbb{R}^n$, with boundary $\partial\Omega$ Lipschitz. Then, the embedding $BV(\Omega) \to L^{\frac{n}{n-1}}(\Omega)$ is continuous, and $BV(\Omega) \to L^p(\Omega)$ is compact for all $1 \leq p \leq \dfrac{n}{n-1}$.

As in [43], let us define $BV_0(\Omega)$ as the space of bounded variational functions that vanish on the boundary $\partial\Omega$ of Ω:

$$BV_0(\Omega) = \{u \in BV(\Omega) : u|_{\partial\Omega} = 0\}. \tag{2.7}$$

More precisely as Ω is bounded and $\partial\Omega$ is Lipschitz, functions of $BV(\Omega)$ have a trace of class L^1 on $\partial\Omega$, and the trace mapping $T : BV(\Omega) \to L^1(\Omega)$ is linear, continuous from $BV(\Omega)$ equipped with the strict convergence to $L^1(\partial\Omega)$ endowed with the strong topology. The space $BV_0(\Omega)$ is then defined as the kernel of T. It is a Banach space, endowed with the introduced norm.

Theorem 5. [31] **Poincaré inequalities**

Let $1 \leq p < +\infty$ and let Ω be a bounded open set. Then, there exists $c_{p,\Omega} > 0$ such that

$$\forall v \in W_0^{1,p}(\Omega);\ \ c_{p,\Omega}.\|v\|_{L^p(\Omega)} \leq \|\nabla v\|_{L^p(\Omega)}. \tag{2.8}$$

Theorem 6. [43] **Poincaré-Wirtinger inequality in $BV(\Omega)$**

Let Ω be a connected and Lipschitz open bounded subset of $\mathbb{R}^n$. There exists a constant $C_\Omega > 0$ such that:

$$\forall u \in BV_0(\Omega) \quad \|u\|_{L^1(\Omega)} \leq C_\Omega TV(u,\Omega). \tag{2.9}$$

2.1.2 The second order bounded variation space

Definition 4. [12] **Second-order total variation**
Let Ω be an open bounded subset of $\mathbb{R}^n$, $n \geq 2$, and for $u \in W^{1,1}(\Omega)$, the second order total variation of u can be defined as:

$$TV^2(u,\Omega) = \sup\left\{\int_\Omega \langle \nabla u, \mathrm{div}\phi\rangle_{\mathbb{R}^n} : \phi \in C_0^2(\Omega), \|\phi\|_\infty \leq 1\right\}. \tag{2.10}$$

where

$$\mathrm{div}\phi = (\mathrm{div}\phi_1, \mathrm{div}\phi_2, ..., \mathrm{div}\phi_n)$$

with

$$\phi_i : \Omega \to \mathbb{R}^n; \phi_i = \left(\phi_i^1, \phi_i^2, ..., \phi_i^n\right) \in \mathbb{R}^n, \forall i = 1, 2, ..., n.$$

Definition 5. [12],[41] **Second-order bounded variation space**
Let $u \in W^{1,1}(\Omega)$, then $u \in BV^2(\Omega)$ if and only if $TV^2(u,\Omega) < +\infty$.

$$BV^2(\Omega) = \left\{u \in W^{1,1}(\Omega) : TV^2(u,\Omega) < +\infty\right\}. \tag{2.11}$$

Proposition 2. [12] *We have:*

$$u \in BV^2(\Omega) \Leftrightarrow \begin{cases} u \in W^{1,1}(\Omega); \\ \dfrac{\partial u}{\partial x_i} \in BV(\Omega), \ \ \forall i \in \{1,2,...,n\}. \end{cases} \tag{2.12}$$

Therefore,

$$BV^2(\Omega) = \left\{ u \in W^{1,1}(\Omega) / \forall i \in \{1,2,...,n\} : \frac{\partial u}{\partial x_i} \in BV(\Omega) \right\}. \tag{2.13}$$

Proposition 3. [4],[41] *The space $BV^2(\Omega)$ is endowed with a norm:*

$$\|u\|_{BV^2(\Omega)} = \|u\|_{BV(\Omega)} + TV^2(u,\Omega). \tag{2.14}$$

is a Banach space.

Theorem 7. [39], [43] **Structural properties of the derivative**

Let Ω be an open subset of $\mathbb{R}^n$ with Lipschitz boundary.

- *For every $u \in BV(\Omega)$, the Radon measure Du can be decomposed into $Du = \nabla u dx + D^s u$, where ∇u is the absolutely continuous part of Du with respect to Lebesgue measure and $D^s u$ is the regular part.*
- *We have the inequalities:*

$$TV^2(u,\Omega) \leq \sum_{i=1}^{n} TV\left(\frac{\partial u}{\partial x_i}, \Omega\right) \leq nTV^2(u,\Omega). \tag{2.15}$$

Theorem 8. [41] **Semi-continuous property**

As the space $BV(\Omega)$, we also have the semi-continuity for $BV^2(\Omega)$. Let $\{u_i\}_{i\in\mathbb{N}} \in BV^2(\Omega)$ which converges to u in $W^{1,1}(\Omega)$. Then,

$$|D^2 u|(\Omega) \leq \liminf_{i\to\infty} \left|D^2 u_i\right|(\Omega). \tag{2.16}$$

Theorem 9. [41],[43],[39] *For $n > 1$, then:*

(i) $BV^2(\Omega) \hookrightarrow W^{1,q}(\Omega)$, for $q \leq \dfrac{n}{n-1}$ with continuous embedding. Moreover, the embedding is compact if $q < \dfrac{n}{n-1}$;

(ii) $BV^2(\Omega) \hookrightarrow L^q(\Omega), \forall q \in [1,+\infty)$ if $n = 2$. In the sequel, if $n = 2$ and Ω is a bounded, open, Lipschitz subset of $\mathbb{R}^2$, so that $BV^2(\Omega) \subset H^1(\Omega)$ with continuous embedding and $BV^2 \subset W^{1,1}(\Omega)$ with compact embedding.

We next define $BV_0^2(\Omega)$ as the space of second bounded variational functions that vanish on the boundary $\partial\Omega$ of Ω:

$$BV_0^2(\Omega) = \left\{ u \in BV^2(\Omega) : u|_{\partial\Omega} = 0 \right\}. \tag{2.17}$$

Theorem 10. [43] **Poincaré-Wirtinger inequality in $BV^2(\Omega)$**

Let Ω be a connected and Lipschitz open bounded subset of $\mathbb{R}^n$. There exists a constant $C_\Omega > 0$ we have:

$$\forall u \in BV_0^2(\Omega) \quad TV(u,\Omega) = \|\nabla u\|_{L^1(\Omega)} \leq C_\Omega TV^2(u,\Omega). \tag{2.18}$$

2.1.3 Relation to Second-order Total Generalized Variation

In [59], K. Bredies and al. proposed and analyzed the concept of *Total Generalized Variation* (TGV), where the definition of order k follows:

Definition 6. [59] **k-order Total Generalized Variation**

Let $\Omega \subset \mathbb{R}^n$ be a domain, $n \geq 1$ and $\alpha_0, ..., \alpha_{k-1} > 0$. Then, the total generalized variation of order k with weight $\alpha = (\alpha_0, ..., \alpha_{k-1})$ for $u \in L^1_{loc}(\Omega)$ is defined as the value of the functional

$$TGV^k_\alpha(u,\Omega) = \sup\left\{ \int_\Omega u \mathrm{div}^k v dx | v \in \mathcal{C}^k_c(\Omega, Sym^k(\mathbb{R}^n)), \left\|\mathrm{div}^l v\right\|_\infty \leq \alpha_l, l = 0, ..., k-1 \right\}. \tag{2.19}$$

with taking the value ∞ if the respective set is unbounded from above.

Here, $Sym^k(\mathbb{R}^n)$ denotes the space of symmetric tensors of order k with arguments in $\mathbb{R}^n$ and $\alpha_l > 0$ are fixed parameters. Choosing $k = 1$ and $\alpha_0 = 1$ gives the total variation functional. In this section we consider the case of $k = 2$, roughly speaking the *second order total generalized variation*.

Definition 7. [59] **Second order Total Generalized Variation**

Let $\Omega \subset \mathbb{R}^n$ be a bounded domain and $\alpha = (\alpha_0, \alpha_1) > 0$. The functional assigning to each $u \in L^1_{loc}(\Omega)$ the value:

$$TGV^2_\alpha(u,\Omega) = \sup\left\{ \int_\Omega u \mathrm{div}^2 v dx | v \in \mathcal{C}^2_c(\Omega, Sym^2(\mathbb{R}^n)), \|v\|_\infty \leq \alpha_0, \|\mathrm{div} v\|_\infty \leq \alpha_1 \right\}. \tag{2.20}$$

is called the Total Generalized Variation of second order.

Correspondingly, the space

$$BGV^2_\alpha(\Omega) = \left\{ u \in L^1(\Omega) | TGV^2_\alpha(u,\Omega) < \infty \right\}. \tag{2.21}$$

equipped with the norm:

$$\|u\|_{BGV^2_\alpha} = \|u\|_1 + TGV^2_\alpha(u,\Omega). \tag{2.22}$$

is called the space of *Bounded Generalized Variation* functions of order 2. Indeed, this second-order bounded generalized variation is slightly different from the second-order bounded variation we mentioned in definitions 4 and 5. In [59], this type of concept is also proposed in order to overcome the disadvantage of staircasing. They similarly found properties and developed a based-variational model in applying to image restoration and denoising. In this thesis, we do not give any comparisons between two different definitions, two different models in denoising, restoration image point of views. We herein just only give a basic understanding of two theories and try to find some relations between them.

Some basic properties of the second order total generalized variation are proposed in [60]. We recall them in theorem 11 as following:

Theorem 11. [60] *Total Generalized Variation of second order enjoys the following properties:*

1. TGV_α^2 *is a semi-norm on the Banach space* $BGV_\alpha^2(\Omega)$*;*
2. $TGV_\alpha^2(u,\Omega) = 0$ *if and only if* u *is a polynomial of degree less than 2;*
3. TGV_α^2 *and* $TGV_{\tilde{\alpha}}^2$ *are equivalent for* $\tilde{\alpha} = (\tilde{\alpha}_0, \tilde{\alpha}_1) > 0$*;*
4. TGV_α^2 *is rotationally invariant;*
5. TGV_α^2 *is proper, convex and lower semi-continuous on each* $L^p(\Omega)$*,* $1 \le p < \infty$*.*

Theorem 12. [60] *Let* $\Omega \subset \mathbb{R}^n$ *be a bounded Lipschitz domain. Then, there exist constants* $0 < c < C < \infty$ *such that for each* $u \in BGV_\alpha^2(\Omega)$ *there holds:*

$$c\,\|u\|_{BV} \le \|u\|_1 + TGV_\alpha^2(u,\Omega) \le C\,\|u\|_{BV}\,. \tag{2.23}$$

Lemma 1. *Let* $\Omega \subset \mathbb{R}^n$ *be a bounded Lipschitz domain. Then, there exists a constant* $n > 0$ *such that for each* $u \in W^{1,1}(\Omega)$ *such that:*

$$TV^2(u,\Omega) \le TV(\nabla u,\Omega) \le nTV^2(u,\Omega). \tag{2.24}$$

Proof. From the definitions 2 and 4 of total variation (TV) and the second total variation (TV^2), we get:

$$TV(\nabla u,\Omega) = \sup\left\{\int_\Omega \langle \nabla u, \mathrm{div}\phi\rangle_{\mathbb{R}^n} : \phi \in C_0^1(\Omega), \|\phi\|_\infty \le 1\right\}. \tag{2.25}$$

$$TV^2(u,\Omega) = \sup\left\{\int_\Omega \langle \nabla u, \mathrm{div}\phi\rangle_{\mathbb{R}^n} : \phi \in C_0^2(\Omega), \|\phi\|_\infty \le 1\right\}. \tag{2.26}$$

Since $C_0^2(\Omega) \subset C_0^1(\Omega)$, we get that $TV^2(u,\Omega) \le TV(\nabla u,\Omega)$.
On the other hand, we analyze the $TV(\nabla u,\Omega)$ as following:

$$\begin{aligned}
TV(\nabla u,\Omega) &= \sup\left\{\int_\Omega \langle \nabla u, \mathrm{div}\phi\rangle_{\mathbb{R}^n} : \phi \in C_0^1\left(\Omega,\mathbb{R}^{n\times n}\right), \|\phi\|_\infty \le 1\right\} \\
&= \sup\left\{\sum_{i=1}^n \int_\Omega \frac{\partial_i u}{\partial x_i}\mathrm{div}v_i;\ v \in C_0^1\left(\Omega,\mathbb{R}^{n\times n}\right), \|v\|_\infty \le 1\right\} \\
&\le \sum_{i=1}^n \sup\left\{\int_\Omega \frac{\partial_i u}{\partial x_i}\mathrm{div}v_i;\ v \in C_0^1\left(\Omega,R^{n\times n}\right), \|v\|_\infty \le 1\right\} \\
&= \sum_{i=1}^n TV\left(\frac{\partial_i u}{\partial x_i},\Omega\right) \\
&\le nTV^2(u,\Omega)(\text{by theorem 7}).
\end{aligned} \tag{2.27}$$

□

Theorem 13. *Let $\Omega \subset \mathbb{R}^n$ be a bounded Lipschitz domain. Then, there exist constants $0 < M_1 < M_2 < \infty$ such that for every $u \in BV_0^2(\Omega)$ satisfying $\nabla u \in BGV^2_{(1,1)}(\Omega)$ there holds:*

$$M_1 \|u\|_{BV^2(\Omega)} \leq \|\nabla u\|_{BGV^2_{(1,1)}(\Omega)} \leq M_2 \|u\|_{BV^2(\Omega)}. \tag{2.28}$$

Proof. First, we consider the norm of ∇u in $BGV^2_{(1,1)}(\Omega)$:

$$\begin{aligned} \|\nabla u\|_{BGV^2_{(1,1)}(\Omega)} &\leq C \|\nabla u\|_{BV(\Omega)} \\ &= C \left(\|\nabla u\|_{L^1(\Omega)} + TV(\nabla u, \Omega) \right) \\ &\leq C \|\nabla u\|_{L^1(\Omega)} + C.n.TV^2(u, \Omega) \\ &\leq C.n. \left(\|\nabla u\|_{L^1(\Omega)} + TV^2(u, \Omega) \right) \\ &\leq M_2. \|u\|_{BV^2(\Omega)}. \end{aligned} \tag{2.29}$$

On the other hand, we have:

$$\begin{aligned} \|u\|_{BV^2(\Omega)} &= \|u\|_{L^1(\Omega)} + \|\nabla u\|_{L^1(\Omega)} + TV^2(u, \Omega) \\ &\leq \|u\|_{L^1(\Omega)} + \|\nabla u\|_{L^1(\Omega)} + TV(\nabla u, \Omega) (\text{by lemma 1}) \\ &\leq C. \left(\|\nabla u\|_{L^1(\Omega)} + TV(\nabla u, \Omega) \right) (\text{by theorem 6}) \\ &= C \|\nabla u\|_{BV(\Omega)} \\ &\leq M_1. \|\nabla u\|_{BGV^2_{(1,1)}(\Omega)}. \end{aligned} \tag{2.30}$$

□

2.2 Variational models

Variational models have been investigated by seminal work by Rudin et al in [18] several years ago. One of the simple problems of image decomposition consists in splitting an original image f into two components u and v. Component u contain the geometrical information while v plays a role of oscillating pattern of f. The regularization term involves only the homogeneous component u, while the remaining term $v := f - u$ models the noise that need to be minimized. We are interested in minimizing functional energies of the following type:

$$\mathcal{F}(u) = \|f - u\|_X + \mathcal{R}(u), u \in Y \subset X \tag{2.31}$$

where X, Y are (real) Banach spaces and $\mathcal{R}$ is a regularization operator.

In this section, we first recall some models as the one by L. Rudin, S. Osher and E. Fatemi [18], or Meyer [66] who gave a different decomposition model in the space $G(\Omega)$ for textures in which the oscillating patterns have a small norm. L. Vese and S. Osher have proposed an approach for the resolution of Meyer's model [57] etc. Then, our approach is introduced to carry out the mathematical study of the second-order model ROF2.

2.2.1 Rudin-Osher-Fatemi model (ROF)

The total variation regularization has been known as a great success in image processing for a long time ago [18]. It is used in many applications of image processing as for example image denoising, image deblurring, restoration and even image zooming and so on. Image restoration model reads:

$$\inf_{u\in BV(\Omega)} F_{ROF}(u); \tag{P_{ROF}}$$

where

$$F_{ROF}(u) = \int_{\Omega} |\nabla u| + \frac{1}{2\lambda} \|f-u\|^2_{L^2(\Omega)}\,. \tag{2.32}$$

Here, Ω is the image domain, a convex Lipschitz open set in $\mathbb{R}^2$: the term $\int_{\Omega} |\nabla u| := TV(u,\Omega)$ stands for the total variation of u and f is the observed image we want to restore.

The minimizer of (P_{ROF}) is the restored image u; and the parameter λ in (P_{ROF}) is a weighting parameter which controls the amount of denoising. The choice of parameter λ affects how much the image is regularized. It can generally be approached via an optimization process where λ is selected to optimize some approximate criteria. We refer to [4], [28] for more details.

The numerical realization is not straightforward. The first idea has been proposed by Chan and Golub [49], and Chambolle generalized the projection algorithm in [4], [50]. This algorithm is now very popular because of the complete proof of convergence: it is the first algorithm that exactly solves the total variation regularization problem. However, the TV minimization, suffers from the so-called staircasing effect, namely, the transformation of smooth regions into piecewise constant ones, which may produce undesirable blocky image. About the analysis of staircasing effect, we can see in Appendix B, or in [42], [5], and [45]. In [42] the authors prove the existence of a staircasing on TV flow; [5] gives a simple explanation to this phenomenon in one-dimensional discrete case of TV minimization. The staircasing in the Perona-Malik diffusion process is described in [45].

We may refer to A. Chambolle [4], [50] and references therein for the mathematical study of this ROF model.

2.2.2 Meyer's model

Yves Meyer has investigated the Rudin-Osher-Fatemi model in 2001 [66], and pointed out some limitations of this model. He proposed a different decomposition model.

Meyer introduced a space G to describe the oscillating component. This space is very close to the dual space of BV space, [2]. The oscillating patterns have a small G-norm. Let us introduce some definitions and useful properties of this space.

Let Ω be an open bounded subset of $\mathbb{R}^n$, $n \geq 2$ with Lipschitz boundary. The total variation of $f \in BV(\Omega)$ is:

$$TV(f,\Omega) = \int_{\Omega} |\nabla f|. \tag{2.33}$$

Definition 8. [2],[66]

G is the Banach space composed of the distributions f which can be written $f = \mathrm{div}(g) = \sum_{i=1}^{n} \partial_i g_i$, where g_i are in L^∞. The following norm is defined on G:

$$\|f\|_G = \inf\left\{ \|g\|_{L^\infty(\Omega,\mathbb{R}^n)} \,|\, f = \mathrm{div}(g) \right\}. \tag{2.34}$$

where $\|g\|_{L^\infty(\Omega,\mathbb{R}^n)} = \mathrm{ess\,sup}_{x\in\Omega} \sqrt{\sum_{i=1}^{n} |g_i|^2(x)}$ *and* $g = (g_1, g_2, ..., g_n)$.

The variational model that Meyer proposed stands:

$$\inf_{u\in BV(\Omega)} TV(u,\Omega) + \alpha \|f-u\|_G. \tag{P_{YM}}$$

As Meyer pointed out in [66], the residual $v = f - u$ in model (P_{ROF}) can be expressed as $v = \mathrm{div}g$ where g is a vector field given, in the sense of distributions, satisfying $g \in L^\infty(\Omega)$. Therefore, the residual $v = f - u$ in the model (P_{ROF}) is expressed, in the sense of distributions, as the divergence of a vetor field $g \in L^\infty(\Omega)$, i.e. as a generalized function, and this also belongs to $L^2(\Omega)$. Then, the space of generalized functions G is given as:

$$G(\Omega) = \{v = \mathrm{div}g, g = (g_1, g_2, ..., g_n);\ \ g_i \in L^\infty(\Omega)\}. \tag{2.35}$$

In fact, G is exactly the space $W^{-1,\infty}(\Omega)$, the dual space of $W_0^{1,1}(\Omega)$ [30], [2]. In the discrete case $\|.\|_G$ is conjugate of the semi-norm of $TV(.)$ [2]. We refer to [66] for the introduction of the space G to model patterns with strong oscillations from the following lemma:

Lemma 2. [66]

Let $f_n, n \geq 1$ be a sequence of functions in $L^2(D)$ with the three following properties (D is a disc centered at 0 with radius R):

1. *There exists a compact set K such that the supports of the $f_n, n \geq 1$ are embedded in K,*
2. *There exist $q > 2$ and $C > 0$ such that $\|f_n\|_{L^q(\mathbb{R}^n)} \leq C$,*
3. *The sequence f_n converges to 0 in the distributional sense.*

Then $\|f_n\|_G$ converges to 0 when n tends towards infinity.

A function that belongs to G may have large oscillations and nevertheless have a small norm. The decomposition model (P_{YM}) has been confirmed to be a good space to model oscillating patterns, especially textures.

2.2.3 Osher-Vese's model

In [57], L. Vese and S. Osher proposed algorithms for the resolution following the ideas of Y. Meyer [66]. They proposed a new model which combines the contour preserving model of (P_{ROF}), with the texture preserving model of Y. Meyer. Their model, quite efficient for image denoising and keeping sharp contours, reads:

$$\inf_{(u,v)\in BV(\Omega)\times G(\Omega)} \int_{\Omega} |\nabla u| + \lambda \|f-u-v\|^2_{L^2(\Omega)} + \mu \|v\|_{G(\Omega)}. \qquad (P_{OV})$$

where Ω is an open bounded set. To compute the solution, they approximate the term $\|v\|_{G(\Omega)}$ by $\left\| \sqrt{\sum_{i=1}^{n} |g_i|^2} \right\|_p$ where $v = \operatorname{div}(g_1, g_2, ..., g_n)$.

2.2.4 Second order model ROF2 - Our approach

Going back to our approach in this section, we are not far away from decomposition model. Given an original image, we split it into a geometrical component and oscillating component (often containing texture, noise and possibly contours). The model we propose is different from the one of Meyer; the oscillating component will be a priori included in the remainder term $v := f - u$ while u represents the cartoon component. We perform a second order analysis to get the smooth component while the oscillating component are not modelled a priori. We plan to introduce this second-order model ROF2 as following.

The observed image f can be divided into two parts:

$$f = u + v$$

where u represents the smooth component (that keeps almost geometrical information of image), while v is the oscillating component. If the image f contains both random noise and texture, then the component v will capture both the texture and the noise. The same situation also happens in the ROF model and the others. For image restoration purpose, ones look for the solution $u + v$ for $u \in BV^2(\Omega)$ and $v \in L^2(\Omega)$. The regularization term involves the BV^2-part u, and the remaining term $v := f - u$ represents the noise that has to be minimized.

Let $\Omega \subset \mathbb{R}^n$ is an open bounded set, we consider the function F as following:

$$\begin{aligned} F_{ROF2}: \quad & BV^2(\Omega) \to \mathbb{R}^+ \\ & u \qquad \mapsto F_{ROF2}(u) \end{aligned}$$

which is defined by:

$$F_{ROF2}(u) = \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV^2(u,\Omega) + \delta \|u\|_{W^{1,1}(\Omega)} \qquad (2.36)$$

where $f \in L^2(\Omega)$ and $\lambda, \delta > 0$. We have the second order model (P_{ROF2}) reads:

$$\inf_{u\in BV^2(\Omega)} F_{ROF2}(u). \qquad (2.37)$$

We note that the term $\delta \|u\|_{W^{1,1}(\Omega)}$ is needed for the existence of solution to problem (2.37). However, we may avoid the use of penalization term $\delta \|u\|_{W^{1,1}(\Omega)}$ if we look for solutions in space BV_0^2 as:

$$BV_0^2 = \left\{ u \in BV^2(\Omega) : u|_{\partial\Omega} = 0 \right\}. \tag{2.38}$$

because we use the Poincaré-Wirtinger inequalities to get an estimation of $\|u\|_{W^{1,1}}$.

From now on, ones can solve the problem:

$$\inf_{u \in BV_0^2(\Omega)} F_{ROF2}(u). \tag{P_{ROF2}}$$

Theorem 14. [43]

Assume that $\lambda > 0$ and $\delta = 0$. Then the problem (P_{ROF2}) has at least a solution.

In this case we follow the proof of Bergounioux in [43].

Proof. Let $u_n \in BV_0^2(\Omega)$ be a minimizing sequence of problem (P_{ROF2}), that is,

$$\lim_{n \to +\infty} F_{ROF2}(u_n) = \inf_{u \in BV_0^2(\Omega)} F_{ROF2}(u) < +\infty. \tag{2.39}$$

The sequence $(u_n)_{n \in \mathbb{N}}$ is bounded in $BV^2(\Omega)$. Indeed, we have $TV^2(u_n, \Omega)$ is bounded. With theorem 10, $TV(u_n, \Omega)$ is also bounded. Since u_n is L^2-bounded it is also bounded in $W^{1,1}(\Omega)$. Therefore, u_n is bounded in $BV^2(\Omega)$. From theorem 9, there is a subsequence of $(u_n)_{n \in \mathbb{N}}$ in $W^{1,1}(\Omega)$ that converges strongly to u^* in $L^2(\Omega)$. As in theorem 8 we get:

$$TV^2(u^*, \Omega) \leq \liminf_{n \to +\infty} TV^2(u_n, \Omega). \tag{2.40}$$

which implies

$$F(u^*) \leq \liminf_{n \to +\infty} F(u_n) = \min_{u \in BV_0^2(\Omega)} F(u). \tag{2.41}$$

Consequently, $u^* \in BV_0^2(\Omega)$. The continuity of the trace operator γ_0 from $W^{1,2}(\Omega)$ to $L^2(\partial\Omega) \subset H^{1/2}(\Omega)$ gives the result since $\gamma_0(u_n) = 0$ for every n and u_n strongly converges to u^* in $L^2(\Omega)$. □

Chapter 3

Three-dimensional second order variational model

Contents

In this chapter, we especially deal with the three-dimensional models that have been generically described in the previous chapter. The so-called isotropic model is performed in details with the fixed point algorithm proposed by Chambolle A. [4] and we also investigate the Nesterov type algorithm to speed up the method, see Weiss P. [23], Piffet L. [41].

As previously mentioned, the observed image f can be divided into a regular part $u \in BV^2(\Omega)$ and an oscillating part $v := f - u \in L^2(\Omega)$. Observing solutions of problem ROF2, we note that the model keeps geometrical information and removes almost noise as well. However, a lot of geometrical information remains together with the texture and noise in the oscillating part v.

In particular, the remainder term $v := f - u$ contains together texture and/or noise and a part of contour shapes. In this sequel, our goal is to locate texture and/or noise and we do not need to work with the cartoon part u any longer. To perform a full texture extraction, we hope to keep most of texture and noise in the oscillating part while many contour shapes disappear. Based on Piffet L. [40], a local anisotropic model is then performed. Moreover, we also sketch out in this chapter two algorithms from Chambolle A. [4] and Nesterov [67], [23] and present some MRI numerical tests in 3D.

3.1 Isotropic model

We consider that a three-dimensional image f can be decomposed into two components $u+v$, $u \in BV^2(\Omega)$ and $v \in L^2(\Omega)$ where $\Omega \subset \mathbb{R}^3$ is an open bounded set. We recall problem ROF2:

$$\inf_{u \in BV_0^2(\Omega)} \frac{1}{2} \|f-u\|^2_{L^2(\Omega)} + \lambda TV^2(u). \qquad (P_{ROF2})$$

In next section, we are going to study problem (P_{ROF2}) in the discrete case.

3.1.1 Discretization process

In the sequel, we consider that $n = 3$ and the size of 3D image $N_1 \times N_2 \times N_3$. The generic component of u is denoted $u_{i,j,k}$.

The space $X = \mathbb{R}^{N_1 \times N_2 \times N_3}$ is endowed with the inner product and norm:

$$\langle u, v \rangle_X = \sum_{\substack{1 \le i \le N_1 \\ 1 \le j \le N_2 \\ 1 \le k \le N_3}} u_{i,j,k} v_{i,j,k} \text{ and } \|u\|_X = \sqrt{\sum_{\substack{1 \le i \le N_1 \\ 1 \le j \le N_2 \\ 1 \le k \le N_3}} u^2_{i,j,k}}$$

and set $Y = X \times X \times X$.

(a) **Discretization of the term $TV(u)$**

We first compute the discrete gradient $\nabla u \in Y$ of the image $u \in X$:

$$(\nabla u)_{i,j,k} = (\nabla u^1_{i,j,k}, \nabla u^2_{i,j,k}, \nabla u^3_{i,j,k})$$

where

$$\nabla u^1_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} & i < N_1 \\ 0 & i = N_1 \end{cases}$$

$$\nabla u^2_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k} & j < N_2 \\ 0 & j = N_2 \end{cases}$$

$$\nabla u^3_{i,j,k} = \begin{cases} u_{i,j,k+1} - u_{i,j,k} & k < N_3 \\ 0 & k = N_3 \end{cases}$$

then we have the first (discrete) total variation $TV(u)$ is given by:

$$TV(u) \simeq J_1(u) = \sum_{\substack{1 \le i \le N_1 \\ 1 \le j \le N_2 \\ 1 \le k \le N_3}} \|(\nabla u)_{i,j,k}\|_{\mathbb{R}^3}. \qquad (3.1)$$

where $\|(\nabla u)_{i,j,k}\|_{\mathbb{R}^3} = \sqrt{(\nabla u^1_{i,j,k})^2 + (\nabla u^2_{i,j,k})^2 + (\nabla u^3_{i,j,k})^2}$.

(b) **Discretization of the divergence operator**

We also perform the discrete setting of divergence operator. We define the continuous setting by $\text{div} = -\nabla^*$ where ∇^* is the adjoint of ∇.

For every $p \in Y$ and $u \in X$ we have:

$$\langle -\mathrm{div} p, u \rangle_X = \langle p, \nabla u \rangle_Y .$$

The discretization of divergence operator is given by:

$$(\mathrm{div} p)_{i,j,k} = \begin{cases} p^1_{i,j,k} - p^1_{i-1,j,k} & 1 < i < N_1 \\ p^1_{i,j,k} & i = 1 \\ -p^1_{i-1,j,k} & i = N_1 \end{cases} + \begin{cases} p^2_{i,j,k} - p^2_{i,j-1,k} & 1 < j < N_2 \\ p^2_{i,j,k} & j = 1 \\ -p^2_{i,j-1,k} & j = N_2 \end{cases}$$
$$+ \begin{cases} p^3_{i,j,k} - p^3_{i,j,k-1} & 1 < k < N_3 \\ p^3_{i,j,k} & k = 1 \\ -p^3_{i,j,k-1} & k = N_3 \end{cases}$$

(c) **Discretization of the term $TV^2(u)$**

We first introduce the Hessian operator of $u \in X$, denoted by Hu, is defined:

$$\begin{aligned}(Hu)_{i,j,k} =&((Hu)^{11}_{i,j,k}, (Hu)^{12}_{i,j,k}, (Hu)^{13}_{i,j,k}, (Hu)^{21,}_{i,j,k}, \\ &(Hu)^{22}_{i,j,k}, (Hu)^{23}_{i,j,k}, (Hu)^{31}_{i,j,k}, (Hu)^{32}_{i,j,k}, (Hu)^{33}_{i,j,k}).\end{aligned}$$

where $(Hu)^{mn}_{i,j,k}$ can be discretized as follows:

For every $i = 1, ..., N_1$, $j = 1, ..., N_2$ and $k = 1, ..., N_3$, the computation of Hu gives:

$$(Hu)^{11}_{i,j,k} = \begin{cases} u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k} & 1 < i < N_1 \\ u_{i+1,j,k} - u_{i,j,k} & i = 1 \\ u_{i,j,k} - u_{i-1,j,k} & i = N_1 \end{cases}$$

$$(Hu)^{12}_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k} - u_{i-1,j+1,k} + u_{i-1,j,k} & 1 < i \le N_1 \\ & 1 \le j < N_2 \\ 0 & j = N_2 \\ 0 & i = 1 \end{cases}$$

$$(Hu)^{13}_{i,j,k} = \begin{cases} u_{i,j,k+1} - u_{i,j,k} - u_{i-1,j,k+1} + u_{i-1,j,k} & 1 < i \le N_1 \\ & 1 \le k < N_3 \\ 0 & i = 1 \\ 0 & k = N_3 \end{cases}$$

$$(Hu)^{21}_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} - v_{i+1,j-1,k} + u_{i,j-1,k} & 1 \le i < N_1 \\ & 1 < k \le N_3 \\ 0 & i = N_1 \\ 0 & k = 1 \end{cases}$$

$$(Hu)^{22}_{i,j,k} = \begin{cases} u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k} & 1 < j < N_2 \\ u_{i,j+1,k} - u_{i,j,k} & j = 1 \\ u_{i,j,k} - u_{i,j-1,k} & j = N_2 \end{cases}$$

$$(Hu)^{23}_{i,j,k} = \begin{cases} u_{i,j,k+1} - u_{i,j,k} - u_{i,j-1,k+1} + u_{i,j-1,k} & 1 < j \leq N_2 \\ & 1 \leq k < N_3 \\ 0 & j = 1 \\ 0 & k = N_3 \end{cases}$$

$$(Hu)^{31}_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} - u_{i+1,j,k-1} + u_{i,j,k-1} & 1 < k \leq N_3 \\ & 1 \leq i < N_1 \\ 0 & k = 1 \\ 0 & i = N_1 \end{cases}$$

$$(Hu)^{32}_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k} - u_{i+,j+1,k-1} + u_{i,j,k-1} & 1 \leq j < N_2 \\ & 1 < k \leq N_3 \\ 0 & j = N_2 \\ 0 & k = 1 \end{cases}$$

$$(Hu)^{33}_{i,j,k} = \begin{cases} u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1} & 1 < k < N_3 \\ u_{i,j,k+1} - u_{i,j,k} & k = 1 \\ u_{i,j,k} - u_{i,j,k-1} & k = N_3 \end{cases}$$

We have

$$\langle \nabla u, \mathrm{div}\phi \rangle = - \langle \phi, \nabla^2 u \rangle \ .$$

Then,

$$TV^2(u) \simeq J_2(u) = \sum_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2 \\ 1 \leq k \leq N_3}} \|(Hu)_{i,j,k}\|_{\mathbb{R}^9}. \tag{3.2}$$

(d) **Discretization for the adjoint operator of H**

Let us consider $H^* : X^9 \to X$ is the adjoint operator of H, it can be defined as follows:

For every $p = (p^{11}, p^{12}, p^{13}, p^{21}, p^{22}, p^{23}, p^{31}, p^{32}, p^{33}) \in X^9$,

$$\begin{aligned}(H^*p)_{i,j,k} = \sigma^{11}_{i,j,k} + \sigma^{12}_{i,j,k} + \sigma^{13}_{i,j,k} + \sigma^{21}_{i,j,k} + \sigma^{22}_{i,j,k} \\ + \sigma^{23}_{i,j,k} + \sigma^{31}_{i,j,k} + \sigma^{32}_{i,j,k} + \sigma^{33}_{i,j,k}\end{aligned}$$

where

$$\sigma^{11}_{i,j,k} = \begin{cases} p^{11}_{i+1,j,k} - 2p^{11}_{i,j,k} + p^{11}_{i-1,j,k} & 1 < i < N_1 \\ p^{11}_{i+1,j,k} - p^{11}_{i,j,k} & i = 1 \\ p^{11}_{i-1,j,k} - p^{11}_{i,j,k} & i = N_1 \end{cases}$$

$$\sigma^{22}_{i,j,k} = \begin{cases} p^{22}_{i,j+1,k} - 2p^{22}_{i,j,k} + p^{22}_{i,j-1,k} & 1 < j < N_2 \\ p^{22}_{i,j+1,k} - p^{22}_{i,j,k} & j = 1 \\ p^{22}_{i,j-1,k} - p^{22}_{i,j,k} & j = N_2 \end{cases}$$

$$\sigma^{33}_{i,j,k} = \begin{cases} p^{33}_{i,j,k+1} - 2p^{33}_{i,j,k} + p^{33}_{i,j,k-1} & 1 < k < N_3 \\ p^{33}_{i,j,k+1} - p^{33}_{i,j,k} & k = 1 \\ p^{33}_{i,j,k-1} - p^{33}_{i,j,k} & k = N_3 \end{cases}$$

$$\sigma^{12}_{i,j,k} = \begin{cases} p^{12}_{i+1,j,k} & i=1, j=1 \\ -p^{12}_{i+1,j-1,k} & i=1, j=N_2 \\ p^{12}_{i+1,j,k} - p^{12}_{i+1,j-1,k} & i=1, 1<j<N_2 \\ -p^{12}_{i,j,k} & i=N_1, j=1 \\ p^{12}_{i,j-1,k} & i=N_1, j=N_2 \\ p^{12}_{i,j-1,k} - p^{12}_{i,j,k} & i=N_1, 1<j<N_2 \\ p^{12}_{i+1,j,k} - p^{12}_{i,j,k} & 1<i<N_1, j=1 \\ p^{12}_{i,j-1,k} - p^{12}_{i+1,j-1,k} & 1<i<N_1, j=N_2 \\ p^{12}_{i,j-1,k} - p^{12}_{i,j,k} - p^{12}_{i+1,j-1,k} + p^{12}_{i+1,j,k} & 1<i<N_1, 1<j<N_2 \end{cases}$$

$$\sigma^{13}_{i,j,k} = \begin{cases} p^{13}_{i+1,j,k} & i=1, k=1 \\ -p^{13}_{i+1,j,k-1} & i=1, k=N_3 \\ p^{13}_{i+1,j,k} - p^{13}_{i+1,j,k-1} & i=1, 1<j<N_3 \\ -p^{13}_{i,j,k} & i=N_1, k=1 \\ p^{13}_{i,j,k-1} & i=N_1, k=N_3 \\ p^{13}_{i,j,k-1} - p^{13}_{i,j,k} & i=N_1, 1<k<N_3 \\ p^{13}_{i+1,j,k} - p^{13}_{i,j,k} & 1<i<N_1, k=1 \\ p^{13}_{i,j,k-1} - p^{13}_{i+1,j,k-1} & 1<i<N_1, k=N_3 \\ p^{13}_{i,j,k-1} - p^{13}_{i,j,k} - p^{13}_{i+1,j,k-1} + p^{13}_{i+1,j,k} & 1<i<N_1, 1<k<N_3 \end{cases}$$

$$\sigma^{21}_{i,j,k} = \begin{cases} p^{21}_{i,j+1,k} & j=1, i=1 \\ -p^{21}_{i-1,j+1,k} & j=1, i=N_1 \\ p^{21}_{i,j+1,k} - p^{21}_{i-1,j+1,k} & j=1, 1<i<N_1 \\ -p^{21}_{i,j,k} & j=N_2, i=1 \\ p^{21}_{i-1,j,k} & j=N_2, i=N_1 \\ p^{21}_{i-1,j,k} - p^{21}_{i,j,k} & j=N_2, 1<i<N_1 \\ p^{21}_{i,j+1,k} - p^{21}_{i,j,k} & 1<j<N_2, i=1 \\ p^{21}_{i-1,j,k} - p^{21}_{i-1,j+1,k} & 1<j<N_2, i=N_1 \\ p^{21}_{i-1,j,k} - p^{21}_{i,j,k} - p^{21}_{i-1,j+1,k} + p^{21}_{i,j+1,k} & 1<j<N_2, 1<i<N_1 \end{cases}$$

$$\sigma^{23}_{i,j,k} = \begin{cases} p^{23}_{i,j+1,k} & j=1, k=1 \\ -p^{23}_{i,j+1,k-1} & j=1, k=N_3 \\ p^{23}_{i,j+1,k} - p^{23}_{i,j+1,k-1} & j=1, 1<k<N_3 \\ -p^{23}_{i,j,k} & j=N_2, k=1 \\ p^{23}_{i,j,k-1} & j=N_2, k=N_3 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j,k} & j=N_2, 1<k<N_3 \\ p^{23}_{i,j+1,k} - p^{23}_{i,j,k} & 1<j<N_2, k=1 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j+1,k-1} & 1<j<N_2, k=N_3 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j,k} - p^{23}_{i,j+1,k-1} + p^{23}_{i,j+1,k} & 1<j<N_2, 1<k<N_3 \end{cases}$$

$$\sigma^{31}_{i,j,k} = \begin{cases} p^{31}_{i,j,k+1} & k=1, i=1 \\ -p^{31}_{i-1,j,k+1} & k=1, i=N_1 \\ p^{31}_{i,j,k+1} - p^{31}_{i-1,j,k+1} & k=1, 1<i<N_1 \\ -p^{31}_{i,j,k} & k=N_3, i=1 \\ p^{31}_{i-1,j,k} & k=N_3, i=N_1 \\ p^{31}_{i-1,j,k} - p^{31}_{i,j,k} & k=N_3, 1<i<N_1 \\ p^{31}_{i,j,k+1} - p^{31}_{i,j,k} & 1<k<N_3, i=1 \\ p^{31}_{i-1,j,k} - p^{31}_{i-1,j,k+1} & 1<k<N_3, i=N_1 \\ p^{31}_{i-1,j,k} - p^{31}_{i,j,k} - p^{31}_{i-1,j,k+1} + p^{31}_{i,j,k+1} & 1<k<N_3, 1<i<N_1 \end{cases}$$

$$\sigma^{32}_{i,j,k} = \begin{cases} p^{32}_{i,j,k+1} & k=1, 1=1 \\ -p^{32}_{i,j-1,k+1} & k=1, j=N_2 \\ p^{32}_{i,j,k+1} - p^{32}_{i,j-1,k+1} & k=1, 1<j<N_2 \\ -p^{32}_{i,j,k} & k=N_3, j=1 \\ p^{32}_{i,j-1,k} & k=N_3, j=N_2 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j,k} & k=N_3, 1<j<N_2 \\ p^{32}_{i,j,k+1} - p^{32}_{i,j,k} & 1<k<N_3, j=1 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j-1,k+1} & 1<k<N_3, j=N_2 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j,k} - p^{32}_{i,j-1,k+1} + p^{32}_{i,j,k+1} & 1<k<N_3, 1<j<N_2 \end{cases}$$

In the isotropic model, we use the discretized canonical scalar product of X^9 defined by:

$$\langle v, u \rangle_{X^9} = \sum_{\substack{1 \le i \le N_1 \\ 1 \le j \le N_2 \\ 1 \le k \le N_3}} \left(v^1_{i,j,k} u^1_{i,j,k} + v^2_{i,j,k} u^2_{i,j,k} + ... + v^9_{i,j,k} u^9_{i,j,k} \right). \tag{3.3}$$

As in [41], [1], [4], using Legendre-Fenchel conjugate function, for every $u \in X$, we have:

$$TV^2(u) = \sup_{p \in \mathcal{C}} \langle p, Hu \rangle_{X^9} . \tag{3.4}$$

where $\mathcal{C} = \left\{ p \in X^9 / \, \|p_{i,j,k}\|_{X^9} \leq 1; 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \right\}$.

We here adapt the proposition of Chambolle, [4] for the second order framework. As we mentioned in section 2.2.1, for the finite dimensional optimization problem from the discretization of problem (P_{ROF}):

$$\inf_{u \in BV(\Omega)} J_1(u) + \frac{1}{2\lambda} \|f - u\|_X^2; \tag{dP_{ROF}}$$

in which the result holds:

Theorem 15. *The solution of (dP_{ROF}) is simply given by:*

$$u = f - P_{\lambda K_1}(f); \tag{3.5}$$

where $K_1 := \{ \text{div} p \mid p \in Y : \|p_{i,j,k}\|_{\mathbb{R}^3} \leq 1; \ \forall 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \}$ *and* $P_{\lambda K_1}(f)$ *is the orthogonal projector operator of* f *on* λK_1.

This theorem is well proved in [4] and [41] for the case of problem ROF. Moreover, in order to approximate $P_{\lambda K_1}(f)$, Chambolle proposed to solve the problem:

$$\begin{cases} \min \|\lambda \text{div} p - f\|_X^2 \\ p \in X^3 \\ \|p_{i,j,k}\|_{\mathbb{R}^3}^2 \leq 1; \ 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \end{cases} \tag{3.6}$$

In the case of second-order problem, we have the discretization of (P_{ROF2}) can be defined as:

$$\inf_{u \in X} J_2(u) + \frac{1}{2\lambda} \|f - u\|_X^2. \tag{dP_{ROF2}}$$

As in [41], [4], [1], we obtain the theorem as following:

Theorem 16. [4], [41]

The solution to problem (dP_{ROF2}) verifies:

$$v = f - P_{\lambda K_2}(f); \tag{3.7}$$

where $P_{\lambda K_2}$ *be the orthogonal projector operator on* λK_2, *and*

$$K_2 := \{ H^* p | \ p \in X^9, \|p_{i,j,k}\|_{\mathbb{R}^9} \leq 1; \ 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \}.$$

We refer to [41], [4] for the proof of theorem. Moreover, in order to approximate the projection term $P_{\lambda K_2}(f)$ in theorem 16 to get the solution of (dP_{ROF2}), as in [4] we have to solve the following problem:

$$\begin{cases} \min \|\lambda H^* p - f\|_X^2 \\ p \in X^9 \\ \|p_{i,j,k}\|_{\mathbb{R}^9}^2 \leq 1, \ 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \end{cases} \tag{3.8}$$

3.1.2 Chambolle's algorithm

In [4] Chambolle proposed a fixed point method for solving problem (dP_{ROF}). We generalize the 3D algorithm hereafter:

Algorithm 1 Approximate the projection operator $P_{\lambda K_1}(f)$

1. $p^0 = 0$; choose $\tau_1 > 0$.
2. Assume that p^n is known, p^{n+1} can be given as:

$$p_{i,j,k}^{n+1} = \frac{p_{i,j,k}^n - \tau_1 \left(\nabla \left[\operatorname{div} p^n - \frac{f}{\lambda} \right] \right)_{i,j,k}}{1 + \tau_1 \left\| \left(\nabla \left[\operatorname{div} p^n - \frac{f}{\lambda} \right] \right)_{i,j,k} \right\|_{\mathbb{R}^3}} \tag{3.9}$$

In addition, we may give a convergence result:

Theorem 17. *Assume that τ_1 satisfies $\tau_1 \leq 12$. Then, $(\lambda \operatorname{div} p^n)_n$ converges to $P_{\lambda K_1}(f)$ as $n \to +\infty$.*

Proof. As in the proof of [4], let define $\kappa_1 = \sup_{\|p\|_Y \leq 1} \|\operatorname{div} p\|$ and the convergence holds if we can show that $\tau_1 \leq 1/\kappa_1^2$.

First, we have:

$$\begin{aligned}
\|\operatorname{div} p\|^2 &= \sum_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2 \\ 1 \leq k \leq N_3}} \left(p_{i,j,k}^1 - p_{i-1,j,k}^1 + p_{i,j,k}^2 - p_{i,j-1,k}^2 + p_{i,j,k}^3 - p_{i,j,k-1}^3 \right)^2 \\
&\leq \sum_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2 \\ 1 \leq k \leq N_3}} \left[\left(p_{i,j,k}^1 \right)^2 + \left(p_{i-1,j,k}^1 \right)^2 + \left(p_{i,j,k}^2 \right)^2 + \left(p_{i,j-1,k}^2 \right)^2 + \left(p_{i,j,k}^3 \right)^2 + \left(p_{i,j,k-1}^3 \right)^2 \right] \cdot \\
&\leq 12 \|p\|_Y^2
\end{aligned}$$

Hence, $\kappa_1^2 \leq 12$ then τ_1 has to satisfy $\tau_1 \leq 1/12$ for the convergence of problem (dP_{ROF}). □

Similarly, we also perform the 3D fixed point algorithm for second order problem (dP_{ROF2}), algorithm 2:

Algorithm 2 Approximate the projection operator $P_{\lambda K_2}(f)$

1. Choose $\tau_2 > 0$.
2. Let $p^0 = 0, n = 0$.
3. Suppose p^n is known, we compute p^{n+1}:

$$p^n_{i,j,k} = p^{n+1}_{i,j,k} + \tau_2 \left[\left(H \left[H^* p - \frac{f}{\lambda} \right] \right)_{i,j,k} + \left\| \left(H \left[H^* p^n - \frac{f}{\lambda} \right] \right)_{i,j,k} \right\|_{R^9} p^{n+1}_{i,j,k} \right] \tag{3.10}$$

which implies:

$$p^{n+1}_{i,j,k} = \frac{p^n_{i,j,k} - \tau_2 \left(H \left[H^* p^n - \frac{f}{\lambda} \right] \right)_{i,j,k}}{1 + \tau_2 \left\| \left(H \left[H^* p^n - \frac{f}{\lambda} \right] \right)_{i,j,k} \right\|_{\mathbb{R}^9}} \tag{3.11}$$

The parameter τ_2 is related to norm of the adjoint operator H^*. We call it κ_2 and give an estimation in lemma 3.

Lemma 3. *The adjoint operator H^* norm, κ_2 satisfies $\kappa_2 \leq 12$.*

Proof. Recall that $\kappa_2 = \sup_{\|p\|_{X^9} \leq 1} \|H^* p\|$.

As

$$\|H^* p\|_X = \sup_{q \in \overline{B}_{X^9}(0,1)} \langle H^* p, q \rangle \tag{3.12}$$

$$\forall p \in X^9 : \langle H^* p, q \rangle_X = \langle p, Hq \rangle_{X^9} \leq \|Hq\|_{X^9} \|p\|_{X^9} \tag{3.13}$$

For all $p \in X^9$,

$$\|Hq\|^2_{X^9} = \sum_{i,j,k} \begin{bmatrix} (q_{i,j+1,k} - q_{i,j,k} - q_{i-1,j+1,k} + q_{i-1,j,k})^2 \\ + (q_{i+1,j,k} - 2q_{i,j,k} - q_{i-1,j,k})^2 \\ + (q_{i,j,k+1} - q_{i,j,k} - q_{i-1,j,k+1} + q_{i-1,j,k})^2 \\ + (q_{i+1,j,k} - q_{i,j,k} - q_{i+1,j-1,k} + q_{i,j-1,k})^2 \\ + (q_{i,j+1,k} - 2q_{i,j,k} + q_{i,j-1,k})^2 \\ + (q_{i,j,k+1} - q_{i,j,k} - q_{i,j-1,k+1} + q_{i,j-1,k})^2 \\ + (q_{i+1,j,k} - q_{i,j,k} - q_{i+1,j,k-1} + q_{i,j,k-1})^2 \\ + (q_{i,j+1,k} - q_{i,j,k} - q_{i,j+1,k-1} + q_{i,j,k-1})^2 \\ + (q_{i,j,k+1} - 2q_{i,j,k} - q_{i,j,k-1})^2 \end{bmatrix} \leq 9 * 16 \|q\|^2_{X^9}$$

this implies $\|H^* p\|_X \leq 12 \|p\|_{X^9}, \forall p \in X^9$.

We deduce that $\kappa_2 \leq 12$. □

Theorem 18. *Let $\tau_2 \leq 1/12^2$, then $\lambda(H^* p^n)_n$ converges to $P_{\lambda K_2}(f)$ as $n \to \infty$.*

Proof. In [41], [4] it is pointed out that the τ_2 in algorithm can be chosen as $\tau_2 \leq 1/\kappa_2^2$, and from lemma 3 we get $\tau_2 \leq 1/12^2$. □

Hence, the approximate solution u can be computed with $u = f - P_{\lambda K_2}(f)$ as in theorem 16 (section 3.1.1).

3.1.3 Nesterov type algorithm

In this section, we adapt the algorithm to compute $P_{\lambda K_2}(f)$ in problem (3.8). It has been proved in [23] that it is faster than the previous method by Chambolle [1], [4]. We refer to Weiss [23], and Piffet [41] for references.

Definition 9. [23] **L-Lipschitz differentiable function**
Let K be a non empty closed convex subset of $\mathbb{R}^n$. A function F defined on K is said to be L-Lipschitz differentiable if it is differentiable on K and that $\|\nabla F(u_1) - \nabla F(u_2)\|_2 \leq L\|u_1 - u_2\|_2$ for any $(u_1, u_2) \in K^2$.

Nesterov, [67] has given a method to solve:

$$\inf_{u \in Q} E(u) \tag{3.14}$$

where E is a convex and L-Lipschitz differentiable function, and Q is a closed convex of $BV^2(\Omega)$.

Theorem 19. [67]

Consider the following algorithm to find solution x:

1. *Set $k = 0$; $G^0 = 0$; $x^k \in Q$ and L is the Lipschitz constant of ∇E.*
2. *$k = k + 1$, compute $\eta^k = \nabla E\left(x^k\right)$.*
3. *$y^k = \arg\min_{y \in Q} \left\{ \langle \eta^k, y - x^k \rangle_X + \frac{1}{2} L \left\| y - x^k \right\|^2 \right\}$.*
4. *$G^k = G^{k-1} + \frac{k+1}{2}\eta^k$.*
5. *$z^k = \arg\min_{z \in Q} \left\{ \frac{L}{\sigma} d(z) + \left\langle G^k, z \right\rangle_X \right\}$.*
6. *Set $x^k = \frac{2}{k+3} z^k + \frac{k+1}{k+2} y^k$. Go back to 2 until $k = N$, the maximal number of iterations.*

Then,

$$0 \leq E\left(y^k\right) - E(\overline{u}) \leq \frac{4Ld(\overline{u})}{\sigma(k+1)(k+2)}$$

where $\overline{u}$ is assumed to be the solution to problem (3.14), and d is a convex function such that $d(x) \geq \frac{\sigma}{2}\|x - x_0\|^2$ for $x_0 \in Q$, $\sigma > 0$.

Proposition 4. [23],[67]

Choose $d(x) = \frac{1}{2}\|x - x_0\|_2^2$ *where* $x_0 \in Q$, $\sigma = 1$. *Then, step* 3 *reduces to* $y^k = P_Q\left(x^k - \frac{\eta^k}{L}\right)$ *and step* 5 *reduces to* $z^k = P_Q\left(x_0 - \frac{G^k}{L}\right)$ *(these projections are described in Remark 1).*

Back to problem (3.8), we are looking for $P_{\lambda K_2}(f)$, and need to write the problem as in (3.14) to apply the Nesterov algorithm. Problem (3.8) can be rewritten:

$$\min\left\{\|\lambda H^* p - f\|_X^2 / p \in X^9, \|p_{i,j,k}\|_{\mathbb{R}^9}^2 \leq 1; 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3\right\}. \tag{3.15}$$

In this case, the problem writes:

$$\min_{p \in E} F(p) \tag{3.16}$$

where,

$$F(p) = \|\lambda H^* p - f\|_X^2, \tag{3.17}$$

$$E = \left\{p \in X^9, \|p_{i,j,k}\|_{\mathbb{R}^9}^2 \leq 1;\ 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3\right\}. \tag{3.18}$$

Next, let $F(p) = \|\lambda H^* p - f\|_X^2 = \|\lambda A(p) - f\|_X^2$ where $A = H^*$. Then,

$$\nabla F(p) = 2A^*\lambda^2\left(A(p) - \frac{f}{\lambda}\right). \tag{3.19}$$

From that point, we get:

$$\begin{aligned} \|\nabla F(p_1) - \nabla F(p_2)\|_2 &= 2\left\|A^*\lambda^2 A(p_1 - p_2)\right\|_2 \\ &= 2\lambda^2 \|A^* A(p_1 - p_2)\|_2 \\ &\leq 2\lambda^2 \|A\|^2 \|p_1 - p_2\|_2 \\ &\leq 2\lambda^2 .12^2 \|p_1 - p_2\|_2 \\ &\leq 288\lambda^2 \|p_1 - p_2\|_2 \end{aligned} \tag{3.20}$$

where we used the lemma 3 for the norm of H^*.

Finally, $\|\nabla F(p_1) - \nabla F(p_2)\|_2 \leq 288.\lambda^2 \|p_1 - p_2\|_2$ that is F is L-Lipschitz differentiable, with $L = 288.\lambda^2$. We apply theorem 19, and get the Nesterov type algorithm for problem (3.16):

Algorithm 3 Approximate the projection operator $P_{\lambda K_2}(f)$ - Nesterov type algorithm

Inputs:

- N is the number of iteration.
- $p_0 \in E$, usually be zeros.

Output: p^N - the approximated solution for (3.16).

Steps:

- L is the Lipschitz constant of ∇F.
- $i = 0$, $G^0 = 0$.
- For $i = 1 : N - 1$
  1. $\eta^i = \nabla F(p^i)$.
  2. $y^i = P_E\left(p^i - \dfrac{\eta^i}{L}\right)$.
  3. $G^i = G^{i-1} + \dfrac{i+1}{2}\eta^i$.
  4. $\upsilon^i = P_E\left(p^0 - \dfrac{G^i}{L}\right)$.
  5. $p^{i+1} = \dfrac{2}{i+3}\upsilon^i + \dfrac{i+1}{i+3}y^i$.

stop after N iterations to obtain an approximation of projection p^N for problem (3.16).

Remark 1. [23] **L^2-Projections on weighted l^∞-balls**

Inside the Nesterov algorithm above, for discretization of projections on weighted l^∞-balls that we used exactly in step 2 and 4.
Let $K = \left\{y \in X, |\lambda\,(y - f)|_p \leq \alpha\right\}$, with $\lambda \in [0, \infty]^n$. The problem of projection on K can be written analytically:

$$P_K\,(x) = \arg\min_{y \in K}\left(\|y - x\|^2_{L^2}\right). \tag{3.21}$$

Let $\bar{y}$ denotes the solution of (3.21). A first important remark that holds for any p is that if $\lambda_i = 0$, then $\bar{y}_i = x_i$. If $\lambda_i = \infty$ then $\bar{y}_i = f_i$. Thus in all projection algorithms the first step is to set all those known values. This allows to restrict out attention to the case $\lambda \in]0, \infty[^n$.

The L^2 projection on l^∞-balls can be written from:

$$\overline{y}_i = \begin{cases} x_i & \text{if } \ |\lambda_{\mathrm{i}}\,(\mathrm{f_i} - \mathrm{x_i})| \leq \alpha, \\ f_i + \dfrac{x_i - f_i}{|x_i - f_i|}\dfrac{\alpha}{\lambda_i} & \text{otherwise.} \end{cases}$$

So, the approximate solution u can be computed $u = f - P_{\lambda K_2}(f)$ as in theorem 16 section 3.1.1.

3.1.4 Comparison between algorithms

In order to compare algorithms 2 and 3, we propose to test the numerical results for 3D vessels of mouse brain volume. As previously mentioned in chapter 1, this is the stack of 51 2D MR images, which are noisy after scanning process. These two algorithms are performed for image denoising on the same vessel volume. In this case there is no ground truth to compare with, we approximate the 'clean' volume $u^* := f - P^*_{\lambda K_2}(f)$ by performing two algorithms for a large enough number of iterations (in the practical model we run for 10000 iterations), where $P^*_{\lambda K_2}(f)$ represents the approximate projection obtained from each algorithm 2 and 3. We call u^* the solution for each simulation, during the first 5000 iterations of the test, we compute and save all norm values $\|u_k - u^*\|_{L^2}$ at every iterative step, plot them inside the same figure.

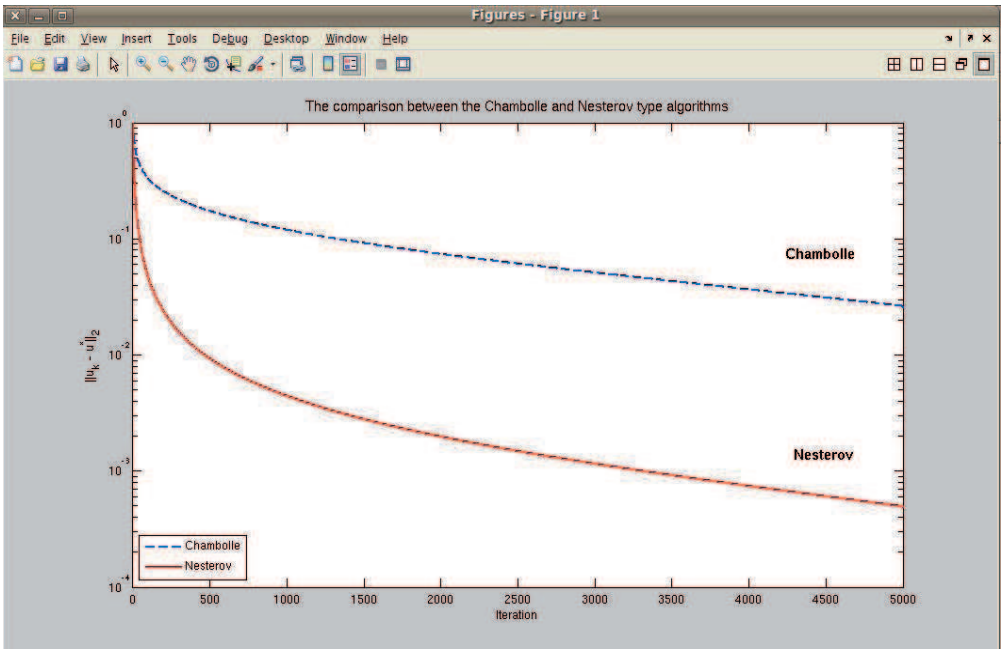


Figure 3.1: Comparison about the norm $\|u_k - u^*\|_2$ through 5000 iterations, tested on algorithm 2 (Chambolle) and 3 (Nesterov).

In Figure 3.1, we show the comparison between the convergence speed of algorithms 2 and 3 on the first 5000 iterations. For both algorithms, we expect that this term of norm $\|u_k - u^*\|_{L^2}$ tends to zero. It can be seen from the Figure that the algorithm 3 converges more quickly than algorithm 2. This shows that algorithm 2 (Chambolle) does not provide as good approximation for (P_{ROF2}) as the other one (Nesterov).

Some two dimensional tests confirmed the same comparison between these two

methods (see [41]). In this thesis, one can apply the improved algorithm 3 in our 3D medical models numerically. Algorithm 3 will be applied for 3D image denoising and/or texture extraction (see the chapter 5).

3.2 Local Anisotropic model

3.2.1 Improvement

In the 2D case, L.Piffet has observed that removing horizontal/vertical coefficients of hessian operator allows to get rid of contour lines corresponding to horizontal/vertical directions in the oscillating part. For example, if we set $Hu^{11} = 0$ horizontal contours disappear, while with $Hu^{22} = 0$ we loose vertical contour lines in texture part. Figures 3.2 give descriptions of local anisotropic tests for Barbara image, Figure 3.2(c) represents oscillating part without horizontal and vertical contour lines.

This method has been improved since there were two major inconveniences for isotropic ROF2 model:

- First, the same transform is performed at every pixel, so that the image is globally treated. All the vertical and horizontal axes are removed;

- Secondly, the transform depends on the chosen (fixed) cartesian axis and it is not possible to remove contours that are not horizontal, vertical or diagonal axes.

Therefore, the Hessian matrix is now locally transformed at every significant pixel (pixels that belong to an edge for example). A rotation is performed such that the gradient direction becomes one of the cartesian coordinate axis, for example the x-axis. We then have the corresponding Hessian operator, and, after rotating we get the new modified Hessian operator. The suitable coefficients are canceled as (3.22) to get rid of the contours. Finally, we come back to the original axis with the inverse rotation.

$$\left(\bar{H}u\right)_{i,j} = \left(0, (Hu)^{12}_{i,j}, (Hu)^{21}_{i,j}, (Hu)^{22}_{i,j}\right) \tag{3.22}$$

For each voxel (i, j, k) that marks a contour position, we compute the 3D Hessian operator and respectively, perform two 3D rotations R_α and R_β such that after rotating, the gradient direction becomes parallel to x-axis for example. Then we get the new modified Hessian operator $\bar{H}$, which coefficients are removed with respect to x-axis as follows:

$$(\tilde{H}u)_{i,j,k} = \begin{pmatrix} 0 & (\bar{H}u)^{12}_{i,j,k} & (\bar{H}u)^{13}_{i,j,k} \\ (\bar{H}u)^{21}_{i,j,k} & (\bar{H}u)^{22}_{i,j,k} & (\bar{H}u)^{23}_{i,j,k} \\ (\bar{H}u)^{31}_{i,j,k} & (\bar{H}u)^{32}_{i,j,k} & (\bar{H}u)^{33}_{i,j,k} \end{pmatrix} \tag{3.23}$$

Finally we go back to the original axis with the inverse rotations $R_{-\beta}$ and $R_{-\alpha}$ with $\tilde{H}$, to obtain the modified Hessian matrix, called H'. Let us detail the process:

(a) Original 2D image - Barbara

(b) Oscillating part after isotropic model

(c) Oscillating part without horizontal and vertical contours

(d) Oscillating part without all contours

Figure 3.2: Effects of anisotropic improvement strategy.

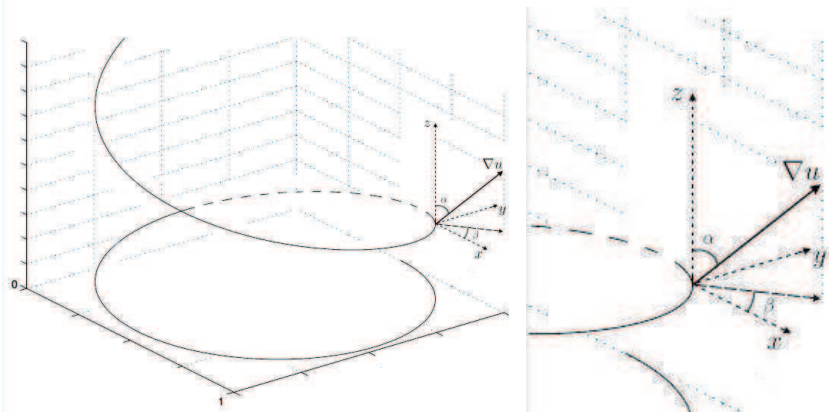


Figure 3.3: Definition of local axis and angles α and β

For example, take a voxel $X_0 = (x_0, y_0, z_0)$. The angles α and β are defined at point $X_0 = (x_0, y_0, z_0)$ as follows: α is the angle between the gradient $\nabla u(x_0, y_0, z_0)$ and the z-axis, β is the angle between the orthogonal projection of:

$$\nabla u(x_0, y_0, z_0) := \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} (x_0, y_0, z_0) \tag{3.24}$$

(on the xOy plane) and the x-axis. Let us define the two rotations: R_α and R_β which matrices are:

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \text{ and } R_\beta = \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} , \tag{3.25}$$

with

$$\alpha = \operatorname{atan}\left(\frac{u_z}{\sqrt{u_x^2 + u_y^2}}\right)(X_0), \quad \beta = \operatorname{atan}\left(\frac{u_y}{u_x}\right)(X_0) . \tag{3.26}$$

Through two rotations R_α first, and next R_β, the change of variables from the fixed basis to the local one is given by:

$$\bar{X} = R_\beta R_\alpha X, \text{ with } X = (x, y, z) \in \mathbb{R}^3 . \tag{3.27}$$

Moreover,

$$X = (R_\beta R_\alpha)^{-1}\tilde{X} = R_\alpha^{-1} R_\beta^{-1}\bar{X} = R_{-\alpha} R_{-\beta}\tilde{X} . \tag{3.28}$$

We next consider the first and second order derivative of $\tilde{u}$:

$$\nabla\bar{u} = \begin{pmatrix} \dfrac{\partial\bar{u}}{\partial\bar{x}} \\ \dfrac{\partial\bar{u}}{\partial\bar{y}} \\ \dfrac{\partial\bar{u}}{\partial\bar{z}} \end{pmatrix} \text{ and } \bar{H} := \begin{pmatrix} \dfrac{\partial^2\bar{u}}{\partial\bar{x}^2} & \dfrac{\partial^2\bar{u}}{\partial\bar{x}\partial\bar{y}} & \dfrac{\partial^2\bar{u}}{\partial\bar{x}\partial\bar{z}} \\ \dfrac{\partial^2\bar{u}}{\partial\bar{y}\partial\bar{x}} & \dfrac{\partial^2\bar{u}}{\partial\bar{y}^2} & \dfrac{\partial^2\bar{u}}{\partial\bar{y}\partial\bar{z}} \\ \dfrac{\partial^2\bar{u}}{\partial\bar{z}\partial\bar{x}} & \dfrac{\partial^2\bar{u}}{\partial\bar{z}\partial\bar{y}} & \dfrac{\partial^2\bar{u}}{\partial\bar{z}^2} \end{pmatrix} .$$

A short computation gives

$$\frac{\partial \bar{u}}{\partial \bar{x}} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial \bar{x}} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial \bar{x}} + \frac{\partial u}{\partial z}\frac{\partial z}{\partial \bar{x}}, \tag{3.29}$$

Finally, we get

$$\nabla \bar{u} = R_{\alpha,\beta} \nabla u \ . \tag{3.30}$$

where $R_{\alpha,\beta}$ denotes results after rotating angles α and β respectively, and the Hessian operator can be modified as:

$$\bar{H} = R_{\alpha,\beta} H \ . \tag{3.31}$$

As already mentioned, the idea is to cancel some terms of the Hessian matrix to get rid of the contours. However, without performing the rotations, there would be only few possible directions, for example vertical, horizontal and diagonal axes in the 2D-case so that many contours are not considered.

Performing the change of variables allows to identify the gradient direction (that is the orthogonal direction to the contour if the gradient is large enough) with the z-axis and then cancel corresponding terms of the matrix $\bar{H}$. Of course, we have to get back to the original situation.

Let us denote $\mathcal{L}$ the (linear) transformation that assigns 0 to some coefficients of $\bar{H}$ (as in Figure 3.3, correspondingly $\bar{H}^{1,1}$, since the rotated gradient vector $\nabla \bar{u}$ respect to the Ox coordinate).

The whole process is described by:

$$H \xrightarrow{R_\alpha, R_\beta} \bar{H} \xrightarrow{\mathcal{L}} \tilde{H} \xrightarrow{R_{-\beta}, R_{-\alpha}} H' \tag{3.32}$$

that is

$$H \to \left[R_{-\alpha} R_{-\beta} \mathcal{L} R_\beta R_\alpha\right] H \ . \tag{3.33}$$

The above rule is the theoretical rule which gives the transformation of the local operator H at X_0.

However, the algorithm computes the respective values of the texture and cartoon parts at a voxel which center X_0 is also the rotation around center.

Therefore performing the direct rotation $R_{\alpha,\beta}$ makes the image rotate (at least locally) around X_0. The computation of p_{x_0,y_0,z_0} is performed and it is not necessary to "go back", we refer to [41] for more details of this different rotation method.

3.2.2 Local anisotropic Algorithm - Chambolle

Algorithm 4 Local anisotropic algorithm - Chambolle

- Choose $\tau > 0, \mu > 0$ is the threshold parameter, compute ∇u.
- Use a thresholding process to identify the contours ($\|\nabla u\| \geq \mu$).
- Set I_μ the set of voxels corresponding to these "significant contours".
    1. For voxels in I_μ, modify as following:

$$H \xrightarrow{R_\alpha, R_\beta} \bar{H} \xrightarrow{\mathcal{L}} \tilde{H} \xrightarrow{R_{-\beta}, R_{-\alpha}} H'$$

    and compute $(H')^*$, where $\mathcal{L}$ denotes the (linear) transformation that assigns 0 to some coefficients of corresponding Hessian matrix.

    2. Same computation as the previous expression (3.11) isotropic algorithm 2, applied with H', we get p_L^N.
- Approximate solution: $u^N = f - \lambda \left(H^* p_L^N\right)$.

3.2.3 Local anisotropic Algorithm - Nesterov

Algorithm 5 Local anisotropic algorithm - Nesterov type

- For $\mu > 0$ - the threshold parameter, λ and compute the L-Lipschitz constant of ∇F (as in algorithm 3). Then, we identify the contours for $(\|\nabla u\| \geq \mu)$.
- For I_μ the set of voxels corresponding to these "significant contours", we perform:
  1. For voxels in I_μ, modify as following:

     $$H \xrightarrow{R_\alpha, R_\beta} \bar{H} \xrightarrow{\mathcal{L}} \tilde{H} \xrightarrow{R_{-\beta}, R_{-\alpha}} H'$$

     and compute $(H')^*$, where $\mathcal{L}$ denotes the (linear) transformation that assigns 0 to some coefficients of corresponding Hessian matrix.
  2. Same as before isotropic algorithm 3, applied with H', for detail:
     (a) $i = 0$, let $G^0 = 0$, $p_L^0 = 0$.
     (b) $\eta^i = \nabla F(p_L^i) = 2\lambda^2 H' \left((H')^* p_L^i - f\right)$.
     (c) $y^i = P_E\left(p_L^i - \dfrac{\eta^i}{L}\right)$.
     (d) $G^i = G^{i-1} + \dfrac{i+1}{2}\eta^i$.
     (e) $v^i = P_E\left(p_L^0 - \dfrac{G^i}{L}\right)$.
     (f) $p_L^{i+1} = \dfrac{2}{i+3}v^i + \dfrac{i+1}{i+3}y^i$.

     we obtain the approximation of projection p_L^N for problem (3.8) after N iterations.
- Approximate solution: $u^N = f - \lambda\left(H^* p_L^N\right)$.

Chapter 4

Wavelet Transforms for 3D data

Contents

Wavelet analysis has been developed during the two past decades. Wavelets are used successfully in many applications such as signal analysis, image analysis, communications systems, and other signal processing applications. The flexibility of wavelets makes them appropriate for many special purposes. Wavelet theory and implementation have been fully described by Daubechies I. [26], Mallat S. [53], [54], [3], [8] and the others.

The ability of wavelets to separate noise from information contained in a signal makes them one of the most popular denoising techniques [51]. Basically speaking, wavelet coefficients of a signal or image are computed using a given wavelet transform and are then thresholded. Wavelet coefficients below a threshold can be replaced by zeros (hard thresholding procedure), and the signal or image is then reconstructed using the inverse discrete wavelet transform.

One and two dimensional wavelet transforms can be further generalized to three dimensions. In a 3D wavelet transform decomposition, a volume is decomposed into eight subbands representing eight octants in the 3D wavelet space. After convolutions in a filter bank, high-pass subbands collect noise and sharp transition features while low-pass subbands contain the trend or approximation of the original data. Therefore, throughout this chapter, we are studying 3D wavelet transform for image restoration using filter banks for medical imaging applications. In addition to the study of the decimated wavelet transform (pyramidal algorithm of Mallat), we also give a presentation of a redundant undecimated wavelet transform (called "à trous" algorithm) therein.

In this chapter, we study three-dimensional discretized wavelet transform that includes wavelet decomposition and reconstruction strategies. In addition, it gives some hints for 3D image denoising and contour detection applications that will be considered in chapters 5 and 6.

4.1 Discrete Wavelet transform - Algorithm of Mallat

4.1.1 3D Wavelet Decomposition

4.1.1.1 Theoretical scheme

Assuming that the 3D volume data can be mathematically represented by the $V(i,j,k) \in [0,...,255]$, for $1 \leq i \leq M, 1 \leq j \leq N, 1 \leq k \leq P$. We denote V as the 3D matrix with size $M \times N \times P$ and $V[i,j,k]$ gives the grey level at voxel $[i,j,k]$ of the volume.

3D Wavelet transform decomposes a 3D volume V into smaller sub-cubes, at resolution level j. Here the original volume V is considered as in the resolution $j=0$. As j increases, the spatial resolution decreases and a "fine-to-coarse" multiresolution decomposition is performed.

The subbanded octant decomposition involves seven separable 3D wavelets and one separable 3D scaling function which can be obtained from one dimensional scaling function $\phi(.)$ and wavelet function $\psi(.)$ by:

$$\Phi(x,y,z) = \phi(x)\phi(y)\phi(z) \tag{4.1}$$

$$\Psi^1(x,y,z) = \psi(x)\phi(y)\phi(z) \tag{4.2}$$

$$\Psi^2(x,y,z) = \phi(x)\phi(y)\psi(z) \tag{4.3}$$

$$\Psi^3(x,y,z) = \phi(x)\psi(y)\phi(z) \tag{4.4}$$

$$\Psi^4(x,y,z) = \psi(x)\phi(y)\psi(z) \tag{4.5}$$

$$\Psi^5(x,y,z) = \psi(x)\psi(y)\phi(z) \tag{4.6}$$

$$\Psi^6(x,y,z) = \phi(x)\psi(y)\psi(z) \tag{4.7}$$

$$\Psi^7(x,y,z) = \psi(x)\psi(y)\psi(z) \tag{4.8}$$

The following theorem generates a wavelet family from translation and dilations of a 3D scaling function $\Phi(x,y,z)$ and 3D wavelet functions $\Psi^i(x,y,z)$ $(i=1,2,...,7)$ on (4.1)-(4.8). This is the 3D extension of the theorem in [53]:

Theorem 20. *Let $(V_j)_{j\in\mathbb{Z}}$ be the multi-resolution approximation of $L^2(\mathbb{R}^3)$, and O_j be the orthogonal complement of V_j in V_{j+1}. Then the wavelet family is generated by seven wavelet functions defined respectively as:*

$$\begin{aligned}
\Psi^1_{j,m_1,m_2,m_3}(x,y,z) &= 2^{-3j/2}\psi\left(x-2^{-j}m_1\right)\phi\left(y-2^{-j}m_2\right)\phi\left(z-2^{-j}m_3\right)\\
\Psi^2_{j,m_1,m_2,m_3}(x,y,z) &= 2^{-3j/2}\phi\left(x-2^{-j}m_1\right)\phi\left(y-2^{-j}m_2\right)\psi\left(z-2^{-j}m_3\right)\\
\Psi^3_{j,m_1,m_2,m_3}(x,y,z) &= 2^{-3j/2}\phi\left(x-2^{-j}m_1\right)\psi\left(y-2^{-j}m_2\right)\phi\left(z-2^{-j}m_3\right)\\
\Psi^4_{j,m_1,m_2,m_3}(x,y,z) &= 2^{-3j/2}\psi\left(x-2^{-j}m_1\right)\phi\left(y-2^{-j}m_2\right)\psi\left(z-2^{-j}m_3\right)\\
\Psi^5_{j,m_1,m_2,m_3}(x,y,z) &= 2^{-3j/2}\psi\left(x-2^{-j}m_1\right)\psi\left(y-2^{-j}m_2\right)\phi\left(z-2^{-j}m_3\right)\\
\Psi^6_{j,m_1,m_2,m_3}(x,y,z) &= 2^{-3j/2}\phi\left(x-2^{-j}m_1\right)\psi\left(y-2^{-j}m_2\right)\psi\left(z-2^{-j}m_3\right)\\
\Psi^7_{j,m_1,m_2,m_3}(x,y,z) &= 2^{-3j/2}\psi\left(x-2^{-j}m_1\right)\psi\left(y-2^{-j}m_2\right)\psi\left(z-2^{-j}m_3\right)
\end{aligned} \tag{4.9}$$

and:

$$\left\{\Psi_j^1\left(\mathbf{x}-2^{-j}\mathbf{m}\right),\Psi_j^2\left(\mathbf{x}-2^{-j}\mathbf{m}\right),\ldots,\Psi_j^7\left(\mathbf{x}-2^{-j}\mathbf{m}\right)\right\}_{\mathbf{m}=(m_1,m_2,m_3)\in\mathbb{Z}^3}$$

is an orthonormal basis of O_j.

Moreover, the

$$\left\{\Psi_j^1\left(\mathbf{x}-2^{-j}\mathbf{m}\right),\Psi_j^2\left(\mathbf{x}-2^{-j}\mathbf{m}\right),\ldots,\Psi_j^7\left(\mathbf{x}-2^{-j}\mathbf{m}\right)\right\}_{(j,m_1,m_2,m_3)\in\mathbb{Z}^4}$$

is an orthonormal basis of $L^2(\mathbb{R}^3)$, *where* $\mathbf{x}=(x,y,z), j=\{1,2,...\}$.

We give hereafter an interpretation of expression (4.9). The formula (4.9) gives dilated and translated version of the mother wavelet function Ψ and scaling function Φ. The variables j and $\mathbf{m}=(m_1,m_2,m_3)$ are integers that scale and dilate the functions ϕ, ψ for generating different wavelets. In three dimensions, the mother wavelet Ψ and scaling functions Φ are here rescaled by a factor two (dyadic scaling), and translated by integers. In particular, from (4.9), the scale at level j is $2^{-3j/2}$ (normalization of the L^2 norm). As in theorem 20 we see that an original function of $L^2(\mathbb{R}^3)$ can be represented in terms of a wavelet expansion (using a linear combination of its wavelet coefficients). Data processing like denoising or compression can then be performed using the corresponding wavelet coefficients. Of course, the results will depend on the choice of wavelet adapted to the data (some type of wavelet are presented in Appendix C).

By theorem 23 of Appendix A we have that for $i=1,2,...,7$:

$$P_{O_j}^i f(\mathbf{x})=2^{-3j}\sum_{\substack{\mathbf{n}=-\infty\\ \mathbf{n}\in\mathbb{Z}^3}}^{+\infty}\left\langle f(u),\Psi_j^i\left(u-2^{-j}\mathbf{n}\right)\right\rangle\Psi_j^i\left(\mathbf{x}-2^{-j}\mathbf{n}\right) \tag{4.10}$$

Detail subband D_j^i, as the details of f at the scale j, is given as follows:

$$D_j^1 f=\left(\left\langle f(u),\Psi_j^1\left(u-2^{-j}\mathbf{n}\right)\right\rangle\right)_{\mathbf{n}\in\mathbb{Z}^3}=\left\{\beta_{j,\mathbf{n}}^1\right\}_{\mathbf{n}\in\mathbb{Z}^3} \tag{4.11}$$

$$D_j^2 f=\left(\left\langle f(u),\Psi_j^2\left(u-2^{-j}\mathbf{n}\right)\right\rangle\right)_{\mathbf{n}\in\mathbb{Z}^3}=\left\{\beta_{j,\mathbf{n}}^2\right\}_{\mathbf{n}\in\mathbb{Z}^3} \tag{4.12}$$

$$D_j^3 f=\left(\left\langle f(u),\Psi_j^3\left(u-2^{-j}\mathbf{n}\right)\right\rangle\right)_{\mathbf{n}\in\mathbb{Z}^3}=\left\{\beta_{j,\mathbf{n}}^3\right\}_{\mathbf{n}\in\mathbb{Z}^3} \tag{4.13}$$

$$D_j^4 f=\left(\left\langle f(u),\Psi_j^4\left(u-2^{-j}\mathbf{n}\right)\right\rangle\right)_{\mathbf{n}\in\mathbb{Z}^3}=\left\{\beta_{j,\mathbf{n}}^4\right\}_{\mathbf{n}\in\mathbb{Z}^3} \tag{4.14}$$

$$D_j^5 f=\left(\left\langle f(u),\Psi_j^5\left(u-2^{-j}\mathbf{n}\right)\right\rangle\right)_{\mathbf{n}\in\mathbb{Z}^3}=\left\{\beta_{j,\mathbf{n}}^5\right\}_{\mathbf{n}\in\mathbb{Z}^3} \tag{4.15}$$

$$D_j^6 f=\left(\left\langle f(u),\Psi_j^6\left(u-2^{-j}\mathbf{n}\right)\right\rangle\right)_{\mathbf{n}\in\mathbb{Z}^3}=\left\{\beta_{j,\mathbf{n}}^6\right\}_{\mathbf{n}\in\mathbb{Z}^3} \tag{4.16}$$

$$D_j^7 f=\left(\left\langle f(u),\Psi_j^7\left(u-2^{-j}\mathbf{n}\right)\right\rangle\right)_{\mathbf{n}\in\mathbb{Z}^3}=\left\{\beta_{j,\mathbf{n}}^7\right\}_{\mathbf{n}\in\mathbb{Z}^3} \tag{4.17}$$

As in one dimensional case Appendix A, we define $D_j^i f$ as:

$$D_j^i f=\left(\left(f(u)*\Psi_j^i(-u)\right)\left(2^{-j}\mathbf{n}\right)\right)_{\mathbf{n}\in\mathbb{Z}^3};\ \ i=1,2,...,7 \tag{4.18}$$

In Figure 4.1, we describe an example of wavelet decomposition scheme performed to one cubic. This volume is decomposed into different coefficients at the level $s = 2$. At the first level $s = 1$, we have 8 octants/subvolumes are D_1^i $(i = 1, 2, ..., 7)$ and A_1. Approximation coefficient A_1 is next decomposed to 8 octants A_2 and D_2^i $(i = 1, 2, ..., 7)$ at the second scale $s = 2$.

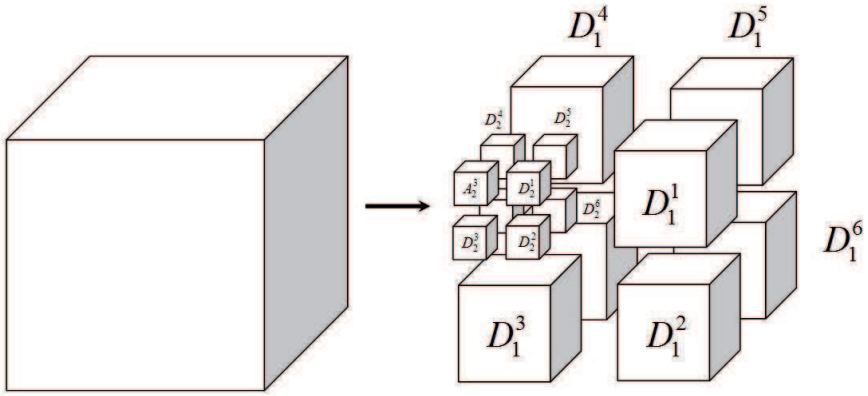


Figure 4.1: The 3D Wavelet Decomposition at first and second scale.

4.1.1.2 Discrete Transform and filter banks

A wavelet transform decomposes successively each approximation $A_j f$ into a coarser approximation $A_{j+1} f$ and wavelet coefficients obtained by $D_{j+1}^i f$. For a discrete image, at scale $j \in \mathbb{Z}(j > 0)$, and for $\mathbf{n} = (n_1, n_2, n_3) \in \mathbb{Z}^3$, we get the approximation and the subband details as:

$$a_j[n_1, n_2, n_3] = \langle f, \Phi_{j,n_1,n_2,n_3} \rangle \tag{4.19}$$

$$d_j^i[n_1, n_2, n_3] = \left\langle f, \Psi_{j,n_1,n_2,n_3}^i \right\rangle, \text{for i} = 1, 2, ..., 7. \tag{4.20}$$

correspondingly being the approximation of f and its details at the scale j.

We note $\langle ., . \rangle$ as the inner product performed between two 3D functions (see details in Appendix A).

A separable three-dimensional convolution can be factorized into one-dimensional convolution along rows (x-direction), columns (y-direction) and images indices (z-direction) of the 3D volume. The multiresolution analysis of f can be obtained using filter banks as:

$$\begin{aligned}
a_{j+1}[n_1, n_2, n_3] &= \left(a_j * \overline{h_x}\ \overline{h_y}\ \overline{h_z}\right)[2n_1, 2n_2, 2n_3] \\
d_hhg_{\{j+1\}}[n_1, n_2, n_3] &= \left(a_j * \overline{h_x}\ \overline{h_y}\ \overline{g_z}\right)[2n_1, 2n_2, 2n_3] \\
d_hgh_{\{j+1\}}[n_1, n_2, n_3] &= \left(a_j * \overline{h_x}\ \overline{g_y}\ \overline{h_z}\right)[2n_1, 2n_2, 2n_3] \\
d_hgg_{\{j+1\}}[n_1, n_2, n_3] &= \left(a_j * \overline{h_x}\ \overline{g_y}\ \overline{g_z}\right)[2n_1, 2n_2, 2n_3] \\
d_ghg_{\{j+1\}}[n_1, n_2, n_3] &= \left(a_j * \overline{g_x}\ \overline{h_y}\ \overline{g_z}\right)[2n_1, 2n_2, 2n_3] \\
d_ghh_{\{j+1\}}[n_1, n_2, n_3] &= \left(a_j * \overline{g_x}\ \overline{h_y}\ \overline{h_z}\right)[2n_1, 2n_2, 2n_3] \\
d_ggh_{\{j+1\}}[n_1, n_2, n_3] &= \left(a_j * \overline{g_x}\ \overline{g_y}\ \overline{h_z}\right)[2n_1, 2n_2, 2n_3] \\
d_ggg_{\{j+1\}}[n_1, n_2, n_3] &= \left(a_j * \overline{g_x}\ \overline{g_y}\ \overline{g_z}\right)[2n_1, 2n_2, 2n_3]
\end{aligned} \tag{4.21}$$

where $*$ denotes the discrete convolution product, h_x (or h_y, h_z) is the one-dimensional low pass decomposition filters by x (or y, z)-direction, g_x (or g_y, g_z) is the one-dimensional high pass decomposition filters by x (or y, z)-direction. These filters are linked to the chosen scaling function Φ and the wavelet function Ψ using the two-scale equation(see [53]). We note the mirror filters:

$$\overline{h}[n_1, n_2, n_3] = h[-n_1, -n_2, -n_3] \tag{4.22}$$

$$\overline{g}[n_1, n_2, n_3] = g[-n_1, -n_2, -n_3] \tag{4.23}$$

This discrete decomposition algorithm is illustrated on Figure 4.2 below with one multiresolution step of the wavelet decomposition.

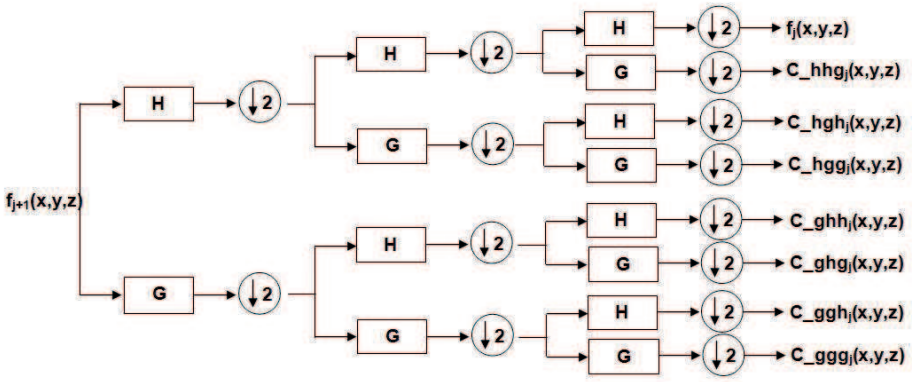
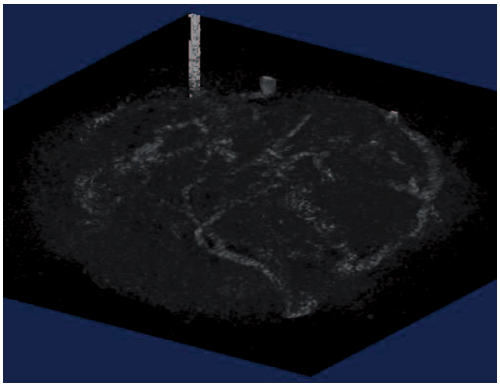


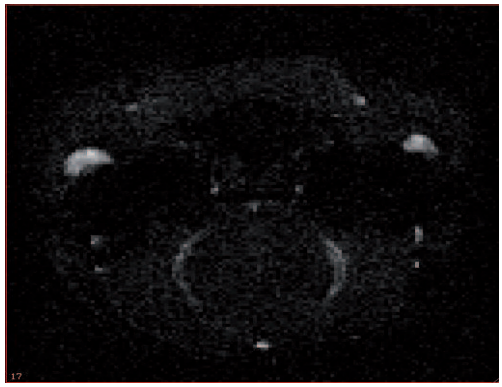
Figure 4.2: Decomposition of a discrete approximation a_j into the next scale $j+1$, an approximation a_{j+1} and $d\_ggg_{\{j+1\}}$, $d\_hgh_{\{j+1\}}$, $d\_hgg_{\{j+1\}}$, $d\_ghg_{\{j+1\}}$, $d\_ghh_{\{j+1\}}$, $d\_ggh_{\{j+1\}}$ and $d\_ggg_{\{j+1\}}$ in the next scale.

The 3D wavelet decomposition is illustrated on the brain vessels volume data, a 3D MRI data that have been presented in chapter 1.

Figure 4.3 shows the original noisy MRI of vessel obtained from biologists (without any pre-filtering step). More details will be given in chapter 8 , where all steps will be detailed more precisely.



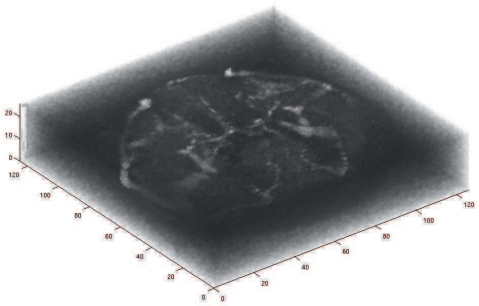
(a) 3D Noisy volume.



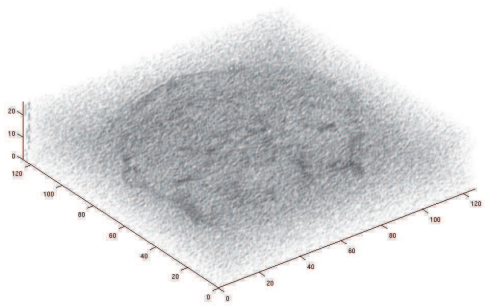
(b) One slice x-axis viewing.

(c) One slice y-axis viewing.

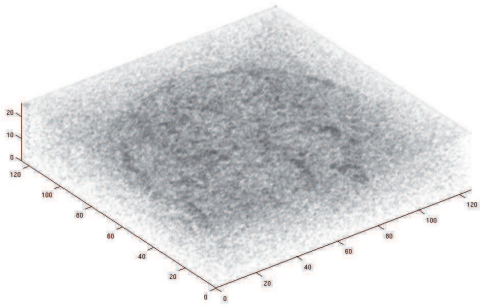
Figure 4.3: Noisy-original Vessel volume.



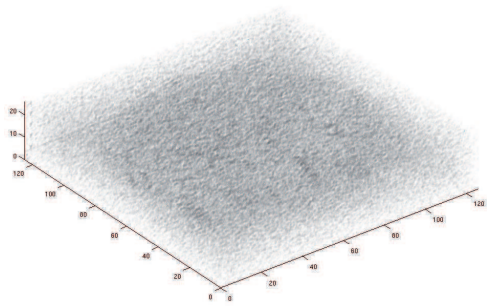
(a) $L^x L^y L^z$.



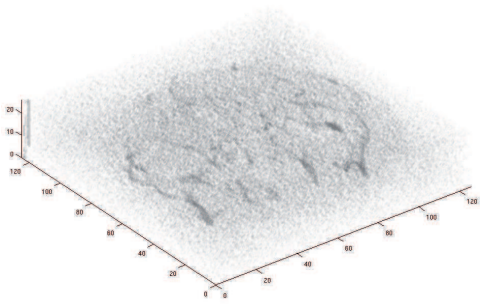
(b) $L^x H^y L^z$.



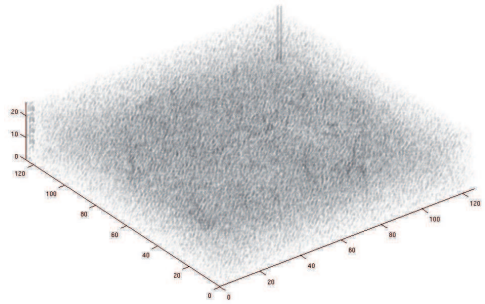
(c) $H^x L^y L^z$.



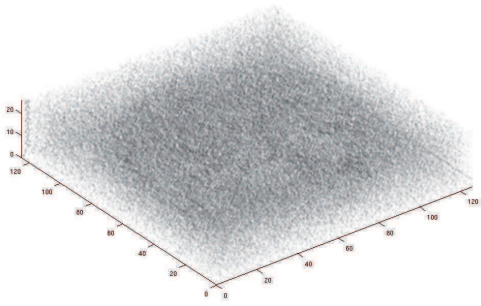
(d) $H^x H^y L^z$.



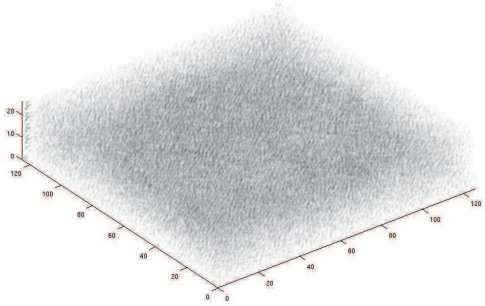
(e) $L^x L^y H^z$.



(f) $L^x H^y H^z$.



(g) $H^x L^y H^z$.



(h) $H^x H^y H^z$.

Figure 4.4: Eight octants/subvolumes resulting from the first level 3D subsampled wavelet decomposition of the MRI Mouse Brain Vessels volume.

The wavelet decomposition scheme has been applied to the vessel volume for one scale of decomposition. Figure 4.4 shows 8 sub-volumes obtained at the first level of decomposition (1 approximation, 7 details). Each octant has a size divided by 2 compared to the original size of the processed volume. These octants were obtained after one level 3D wavelet transform using the *Haar* basis, the simplest orthogonal wavelet basis. We can observe that the detail octants show more textures and contours than the low-pass one $L^x L^y L^z$. The energy (visual vessel filament structures) contained in the low-pass octant is higher than those of high-pass octants. After one scale of decomposition along each direction, the new approximation subband is decomposed further, producing the same number of samples in the subbands than in the original finest resolution image.

4.1.2 3D Wavelet Reconstruction

4.1.2.1 Theoretical scheme

In the previous section, we have seen that the wavelet decomposition is complete. It is then possible to recover the approximation $A^d_{2^j}$ from the coarsest scale approximation $A^d_{2^{j+1}}$ and the other details $D^i_{2^{j+1}}$, $i = 1, .., 7$, for $j < 0$. We show that the original volume can also be reconstructed with a inverse pyramid transformation. The one-dimensional reconstruction is described in details in Appendix A. The three-dimensional scheme is derived hereafter.

Since O_j is the orthogonal complement of V_j in V_{j+1} as in theorem 20, then:

$$\left\{ \begin{array}{l} \sqrt{2^{-3j}}\Phi_j\left(u-2^{-j}\mathbf{n}\right), \sqrt{2^{-3j}}\Psi^1_j\left(u-2^{-j}\mathbf{n}\right) \\ \sqrt{2^{-3j}}\Psi^2_j\left(u-2^{-j}\mathbf{n}\right), \sqrt{2^{-3j}}\Psi^3_j\left(u-2^{-j}\mathbf{n}\right) \\ \sqrt{2^{-3j}}\Psi^4_j\left(u-2^{-j}\mathbf{n}\right), \sqrt{2^{-3j}}\Psi^5_j\left(u-2^{-j}\mathbf{n}\right) \\ \sqrt{2^{-3j}}\Psi^6_j\left(u-2^{-j}\mathbf{n}\right), \sqrt{2^{-3j}}\Psi^7_j\left(u-2^{-j}\mathbf{n}\right) \end{array} \right\}_{\mathbf{n}=(n_1,n_2,n_3)\in\mathbb{Z}^3} \tag{4.24}$$

is an orthonormal basis of V_{j+1}. The function $\Phi_{j+1}(\mathbf{x}-2^{-j-1}\mathbf{n})$ can be decomposed in this basis:

$$\begin{aligned} \Phi_{j+1}\left(\mathbf{x}-2^{-j-1}\mathbf{n}\right) &= 2^{-3j}\sum_{\mathbf{k}}\left\langle\Phi_j\left(u-2^{-j}\mathbf{k}\right), \Phi_{j+1}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\Phi_j\left(\mathbf{x}-2^{-j}\mathbf{k}\right) \\ &+2^{-3j}\sum_{\mathbf{k}}\left\langle\Psi^1_j\left(u-2^{-j}\mathbf{k}\right), \Phi_{j+1}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\Psi^1_j\left(\mathbf{x}-2^{-j}\mathbf{k}\right) \\ &+2^{-3j}\sum_{\mathbf{k}}\left\langle\Psi^2_j\left(u-2^{-j}\mathbf{k}\right), \Phi_{j+1}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\Psi^2_j\left(\mathbf{x}-2^{-j}\mathbf{k}\right) \\ &+2^{-3j}\sum_{\mathbf{k}}\left\langle\Psi^3_j\left(u-2^{-j}\mathbf{k}\right), \Phi_{j+1}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\Psi^3_j\left(\mathbf{x}-2^{-j}\mathbf{k}\right) \\ &+\ldots \\ &+2^{-3j}\sum_{\mathbf{k}}\left\langle\Psi^7_j\left(u-2^{-j}\mathbf{k}\right), \Phi_{j+1}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\Psi^7_j\left(\mathbf{x}-2^{-j}\mathbf{k}\right) \end{aligned} \tag{4.25}$$

Computing the inner product with f gives:

$$\begin{aligned}\left\langle f, \Phi_{2^{j+1}}\left(.-2^{-j-1}\mathbf{n}\right)\right\rangle &= 2^{-3j}\sum_{\mathbf{k}}\left\langle \Phi_{2^j}\left(u-2^{-j}\mathbf{k}\right), \Phi_{2^{j+1}}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\left\langle f(u), \Phi_{2^j}\left(\mathbf{x}-2^{-j}\mathbf{k}\right)\right\rangle \\ &+2^{-3j}\sum_{\mathbf{k}}\left\langle \Psi^1_{2^j}\left(u-2^{-j}\mathbf{k}\right), \Phi_{2^{j+1}}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\left\langle f(u), \Psi^1_{2^j}\left(\mathbf{x}-2^{-j}\mathbf{k}\right)\right\rangle \\ &+2^{-3j}\sum_{\mathbf{k}}\left\langle \Psi^2_{2^j}\left(u-2^{-j}\mathbf{k}\right), \Phi_{2^{j+1}}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\left\langle f(u), \Psi^2_{2^j}\left(\mathbf{x}-2^{-j}\mathbf{k}\right)\right\rangle \\ &+2^{-3j}\sum_{\mathbf{k}}\left\langle \Psi^3_{2^j}\left(u-2^{-j}\mathbf{k}\right), \Phi_{2^{j+1}}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\left\langle f(u), \Psi^3_{2^j}\left(\mathbf{x}-2^{-j}\mathbf{k}\right)\right\rangle \\ &+\ldots \\ &+2^{-3j}\sum_{\mathbf{k}}\left\langle \Psi^7_{2^j}\left(u-2^{-j}\mathbf{k}\right), \Phi_{2^{j+1}}\left(u-2^{-j-1}\mathbf{n}\right)\right\rangle\left\langle f(u), \Psi^7_{2^j}\left(\mathbf{x}-2^{-j}\mathbf{k}\right)\right\rangle\end{aligned} \tag{4.26}$$

where $\mathbf{x}=(x,y,z)\in\mathbb{R}^3$ and $\mathbf{k}=(k_1,k_2,k_3)\in\mathbb{Z}^3$.

$A_{2^{j+1}}f$ can be reconstructed from $A_{2^j}f$ and $D^i_{2^j}f$, for $i=1,...,7$ by convolving with the reconstruction filters $\tilde{H}$ and $\tilde{G}$ (see Appendix A).

4.1.2.2 Discrete reconstruction

The wavelet reconstruction enables to recover each $A_{2^j}f$ from $A_{2^{j+1}}f$ and $D^i_{2^{j+1}}f$. As in (4.26), the discrete reconstruction can be processed quickly at the next scale from eight sub-volumes of the previous scale using reconstruction filter bank. The diagram 4.5 illustrates a one level wavelet reconstruction.

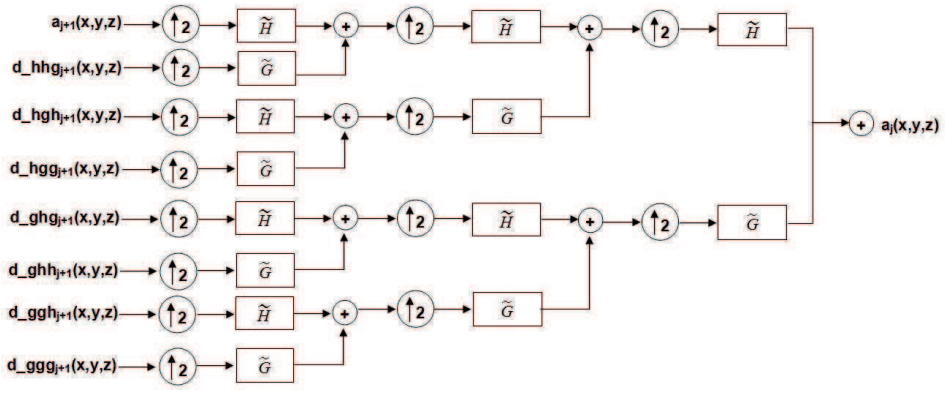


Figure 4.5: Reconstruction of an image $A^d_{2^{j+1}}f$ from $A^d_{2^j}f$ and $D^i_{2^j}f$. The x (y, z)-direction samples are convolved with one dimensional filters $\tilde{H}$ and $\tilde{G}$.

In Figure 4.5, the approximation a_j is recovered from a coarser scale approximation a_{j+1} and d^i_{j+1} ($i=1,...,7$). Eight separable convolutions with reconstruction filters along x, y, z-directions are shown. If no threshold on the wavelet coefficients is applied, image reconstruction is performed without loss and enables to obtain the original 3D image.

4.2 Undecimated Wavelet transform - the "À trous" Algorithm

In the previous section multiresolution analysis using the wavelet transform of Mallat and al. [53]. The multiresolution analysis is a pyramidal analysis because of the decimation step after each convolution. This decimation enables the analysis not to be redundant, but it also implies that the multiresolution analysis is not shift invariant, which is a drawback when denoising signals. Moreover, due to the extension of 1D wavelet into 2D wavelets by tensor product, the analyzing wavelet is not isotropic and favors horizontal and vertical orientations. In this section, we introduce another type of wavelet transform without the decimation step as in the discrete wavelet transform, which is called the "à trous" wavelet transform [56].

The "à trous" algorithm is a fast dyadic wavelet transform and is implemented with filter banks. It is similar to a fast biorthogonal wavelet transform but without subsampling. The "à trous" produces a single wavelet coefficient image or volume at each decomposition level, and it allows the separation between low frequencies (approximation image) from high frequencies (wavelet coefficient).

More precisely, in the "à trous" algorithm, for any scale level $j \geq 0$, the approximation a_j and detail coefficient d_j are obtained by:

$$a_j[n,m,l] = \langle f(x,y,z), \phi_j(x-n)\phi_j(y-m)\phi_j(z-l)\rangle \tag{4.27}$$

$$d_j[n,m,l] = \langle f(x,y,z), \psi_j(x-n)\psi_j(y-m)\psi_j(z-l)\rangle \tag{4.28}$$

where discrete image values are assimilated to $a_0[n,m,l]$.

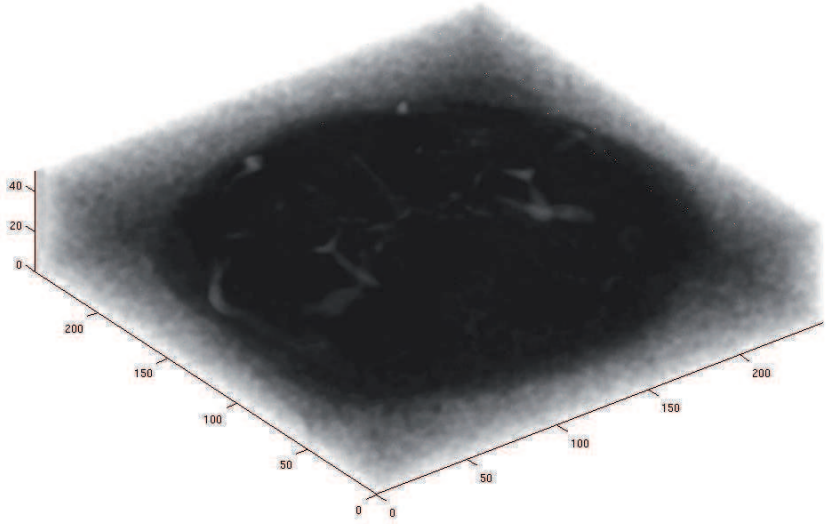
Similar to the DWT in the last section 4.1.1, this wavelet transform uses a filter bank h and g. The set $W = \{d_1, d_2, d_3, ..., d_J, a_J\}$ is obtained, where d_j are the wavelet coefficients at the scale $0 \leq j \leq J$ and a_J are the coefficients at the coarsest resolution.

A filter $x[\mathbf{n}]$ is dilated to make the filter $x_j[\mathbf{n}]$ by inserting $2^{j-1}-1$ zeros between the filter coefficients at each decomposition level j, [46]. Let us denote $\bar{x}_j[\mathbf{n}] = x_j[-\mathbf{n}]$ and $\delta[\mathbf{n}]$ the discrete Dirac function. In addition, $\bar{h}$ is a low-pass filter associated with the scaling function ϕ and $\bar{g}$ is a high-pass filter associated with the mother wavelet ψ.

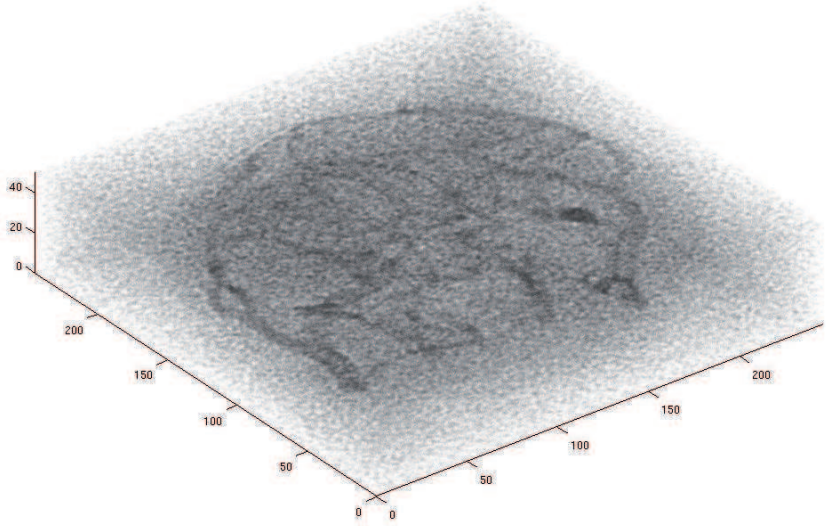
For the practical implementation of the "à trous" algorithm, a 3D filter associated to the scaling function is used. The scaling and wavelet functions ϕ and ψ are cubic B-splines of order three that enable a nearly isotropic analysis of the 3D images, and filters used are separable 1D filters. Spline functions, piecewise polynomials, have good data approximation properties ([61]). If we choose a B_3-spline for scaling function, the coefficients of the convolution mask in one dimension is:

$$B_3 = \frac{1}{16}[1,4,6,4,1] \tag{4.29}$$

As in [64], the "à trous" algorithm can be applied in order to obtain wavelet coefficients at one resolution from the previous approximations using the following



(a) Approximation image.



(b) "À trous" wavelet coefficient.

Figure 4.6: Two coefficients from the first level 3D "à trous" wavelet decomposition of the MRI of Mouse Brain Vessel volume.

equations (in three dimensions):

$$a_{j+1}[n,m,l] = \left(\bar{h}_j\bar{h}_j\bar{h}_j * a_j\right)[n,m,l] \tag{4.30}$$

$$d_{j+1}[n,m,l] = \left(\left[\delta\delta\delta - \bar{h}_j\bar{h}_j\bar{h}_j\right] * a_j\right)[n,m,l] \tag{4.31}$$

where $*$ is the convolution operator, $\bar{h}_j\bar{h}_j\bar{h}_j$ and $\delta\delta\delta$ are 3D filters obtained from $\bar{h}$ and δ by tensor product.

Moreover, the reconstruction is given by summing the details at all scales and

the last approximation:

$$a_0[n,m,l] = a_J[n,m,l] + \sum_{j=1}^{J} d_j[n,m,l] \tag{4.32}$$

Using this decomposition process, this wavelet transform produces two volumes of the same size, respectively called the approximation and the details. The undecimated wavelet decomposition has been applied on figure 4.6. While the approximation keeps all low frequencies, the detail coefficients contains high frequencies of the volume (mostly contours and textures).

The 3D "à trous" algorithm will be used for denoising 3D medical images in the framework of the ROF2 model in the next chapters.

Chapter 5

Application to 3D Image Denoising and Texture Extraction

Contents

This chapter 5 is devoted to numerical tests for 3D image denoising and texture extraction by both mathematical points of view: variational model and wavelet technique discussed previously. In this chapter we consider how these methods bahave on 3D examples of video and MR images. Besides, the chapter also discusses about texture extraction. It is an important task in many computer applications of image detection. During denoising process, we do not want to loose some useful information, that's why the local anisotropic algorithm is performed to determine the texture and/or contour information.

In the second part of the chapter, we shall discuss about the wavelet transform. As previously studied in chapter 4, we have two different types of wavelet transform: Mallat's subsampled wavelet and the "à trous" unsubsampled wavelet transform. Chapter 5 deals with detailed numerical performances for 3D volume examples. In addition, we thank to [51] for the 3D wavelet proposed method that combined 2D and 1D wavelet transform together in the later work on 3D image denoising. This chapter also gives us the 3D denoising representation with fully automated at a high speed of this 2D+1D wavelet technique.

To illustrate each method, we present some numerical results tested on the experimental examples arising in biomedical imaging. It is also provided a comparison between considered methods, analysis of some experimental results and some evaluations, some conclusions about our approaches.

5.1 Variational models

5.1.1 Application to 3D image denoising

In this section we recall the variational approach of the model (P_{ROF2}) that finds out solution of the following problem:

$$\inf_{u\in BV^2(\Omega)} \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV^2(u). \qquad (P_{ROF2})$$

The discretized version stands:

$$\inf_{u\in X} J_2(u) + \frac{1}{2\lambda}\|f-u\|^2_X \qquad (dP_{ROF2})$$

where J_2 has been given in (3.2). Solution u of problem (dP_{ROF2}) is the recovered/denoised image that we expect to obtain.

In what follows we present some denoising numerical tests using this variational approach. The tests are examples of a video moving disk and a small noisy cerebellum data in three dimensions, respectively.

5.1.1.1 Numerical Tests on 2D+T Video - Moving Disk

We here describe the test on a video rendering of a moving disk, it sketches out the movement of a circle from left to right. This visualized volume is the 2D+Time video that composed from 64 slices with the size 128×128, in Figure 5.1. We describe the video of moving and snapshot of this process each slice through 3D volume viewer in Figure 5.1(b), we add the surface viewing of a cutting slice in video Figure 5.1(c) to see how smooth the original version is.

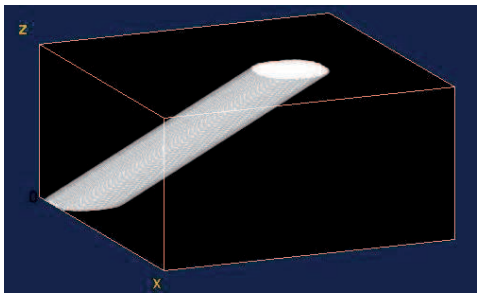
In the sequel we focus on the denoising process, so we consider volumes that are degraded with additive *Gaussian noise* throughout this section. The original moving disk described on Figure 5.1 has been corrupted with a noise with different *standard deviation* σ. We took examples for $\sigma \in \{5, 10, 25, 50\}$. The noisy volume examples and their surface structures are described in Figure 5.2 two cases of $\sigma = 10$ and 50.

The algorithms for discretized problem (dP_{ROF2}) are applied to give an approximate solution for different values of λ. The stopping criterion has been set to maximal iteration *itmax*, which may be chosen large enough (this depends upon the CPU processing speed).

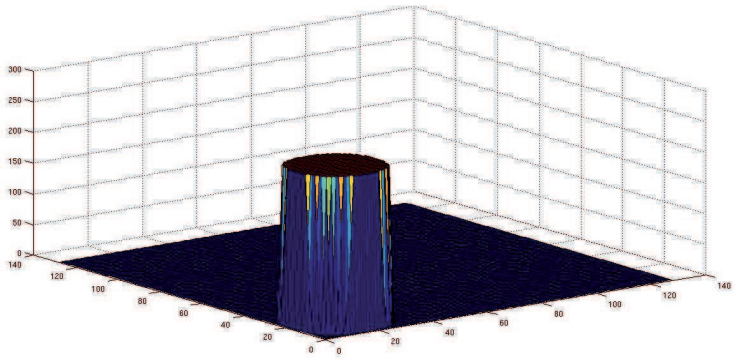
In Figure 5.3 we present some numerical results from noisy volume in Figures 5.2, for standard deviation $\sigma = 50$, up to the change of λ.

For a fixed σ, we could see that the model (dP_{ROF2}) is quite efficient for denoising purpose. In Figure 5.3, since it may not be convenient to have good visualization in 3D viewer, we describe two arbitrary slices representation to observe what happens. We note that, larger λ gives us a smoother solution. It makes blur precisely slice by slice when parameter λ increases. Figure 5.4 represents the surface viewing of each result up to λ in that case.

(a) Original moving disk.



(b) 3D viewer of the video.



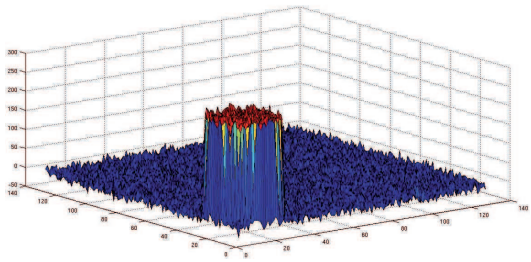
(c) Surface data viewing, the 3^{rd} slice.

Figure 5.1: Test on moving disk video.

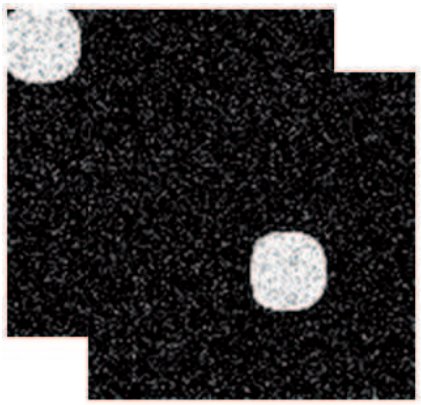
The consideration of video that is degraded by some Gaussian noise with various values of σ gives us some observed results by two Figures 5.3 and 5.4, we can see that the model (P_{ROF2}) is quite efficient for image denoising aspect. Besides that, the method keeps contour information better than the classical model Rudin-Osher-Fatemi (P_{ROF}). Figure 5.4 describes more of loosing staircase effects the model (P_{ROF2}) brings us. Compare to the surface viewing of the original disk in Figure 5.1, the noisy data ($\sigma = 50$), the result with $\lambda \geq 25$ gives us acceptable denoised results.

As previously mentioned, λ plays an important role in denoising process because it affects how much the image is regularized, balancing between removing the noise and preserving the image content. As λ is small the TV^2 term plays a decreasingly strong role, which forces the result to have smaller term of TV^2. Moreover, if λ is too large, the TV^2 term is also large to give a well adapted denoised image, but image details are almost lost. Thus, the choice of regularization parameter is critical to achieve the right amount of noise removal. For the noisy moving disk example ($\sigma = 50$), we visualize the result is acceptable for $\lambda = 25$. We get denoised image with geometrical contents are well retained/preserved better than the other choice

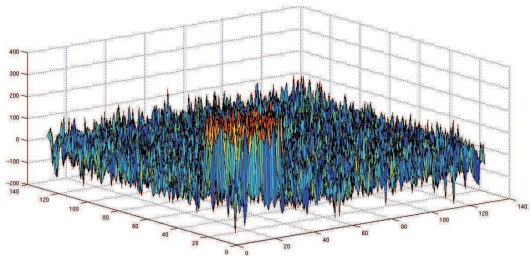
(a) $\sigma = 10$



(b) Surface viewing, $\sigma = 10$



(c) $\sigma = 50$

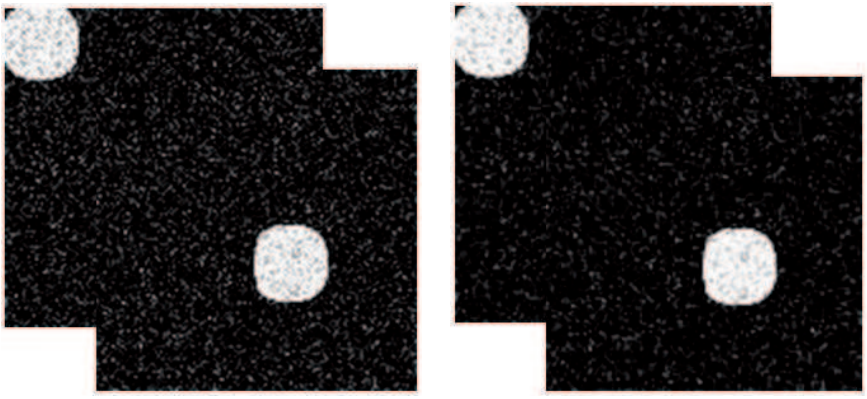


(d) Surface viewing, $\sigma = 50$

Figure 5.2: Adding noise video, for different σ, $\sigma = 10, 50$ and its cutting surface viewing, the 3^{rd} slice.

of λ.

Since noise removal techniques are designed to enhance the image quality, the visualization of image is not enough to compare exactly, moreover we have a lot of 3D viewers of denoised volume data, then it's necessary to perform more evaluation of results. In fact, there is no good objective criterion available for measuring the perceived image similarity. However, there are a number of common error measurements, for instance the Mean Absolute Error (MAE), Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). Firstly, we evaluate their performance regarding two criteria. Criterion 1 considers the quality of denoised image based on its visual impression (MAE). Criterion 2 considers the quality of the removed noise (MSE), that is the simplest of image quality measurement. MSE measures the average amount of difference between pixels of an image and its reconstructed image. If the MSE is small, the reconstructed image looks like the original. Two of them



(a) $\lambda = 1$.

(b) $\lambda = 5$.

(c) $\lambda = 10$.

(d) $\lambda = 25$.

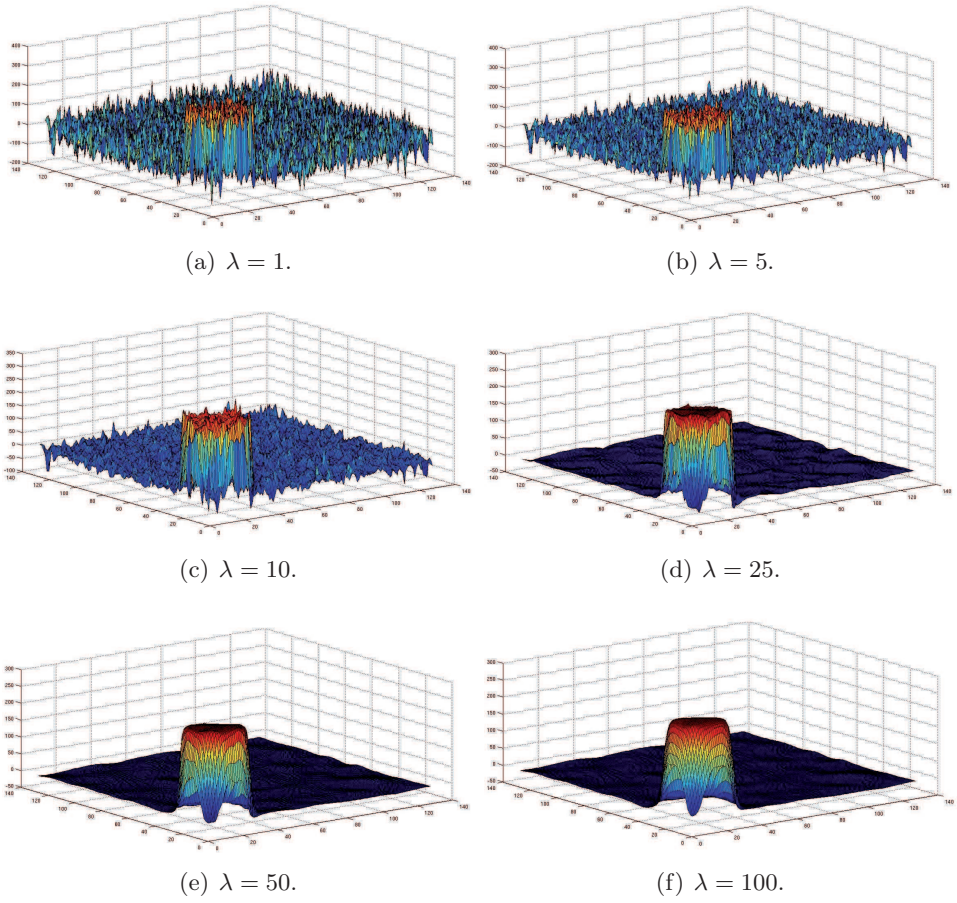
(e) $\lambda = 50$.

(f) $\lambda = 100$.

Figure 5.3: Solution obtained by (P_{ROF2}) model, input noisy data of moving disk, standard deviation $\sigma = 50$.

are calculated as in expressions:

$$\mathrm{MSE} = \frac{1}{\mathrm{MNP}} \sum_{m=0}^{\mathrm{M}-1} \sum_{n=0}^{\mathrm{N}-1} \sum_{p=0}^{\mathrm{P}-1} [I(m,n,p) - K(m,n,p)]^2 \tag{5.1}$$



(a) $\lambda = 1$. (b) $\lambda = 5$.

(c) $\lambda = 10$. (d) $\lambda = 25$.

(e) $\lambda = 50$. (f) $\lambda = 100$.

Figure 5.4: Cutting surface viewing of ROF2 disk denoising, one visual slice.

$$\mathrm{MAE} = \frac{1}{\mathrm{MNP}} \sum_{m=0}^{\mathrm{M}-1} \sum_{n=0}^{\mathrm{N}-1} \sum_{p=0}^{\mathrm{P}-1} |I(m,n,p) - K(m,n,p)| \tag{5.2}$$

where $I(m,n,p)$ and $K(m,n,p)$ represent the original image and the denoised image respectively; M, N, P is the discrete size of our 3D data.

We also perform the concept *Peak-signal-to-noise* ratio (PSNR) to estimate denoised volume. PSNR amount of useful data versus the amount of noise introduced into the image. PSNR is defined in logarithmic scale, in dB (decibels):

$$\mathrm{PSNR} = 20 \log_{10} \left(\frac{\mathrm{MAX_I}}{\sqrt{\mathrm{MSE}}} \right) \tag{5.3}$$

where $\mathrm{MAX_I}$ is the maximum possible pixel value of the image I. When the pixels are represented using 8 bits per sample, this is 255. The higher the number of PSNR, the more accurate the reconstruction. Such a quality measurement is a criterion that can make us confident about the hypothesis we are to consider. In the experiment, the calculated PSNR between the initial images and reconstructed images takes values around 30dB. In most cases, a PSNR greater than 30dB is

considered as leading to a correctly reconstructed image and the image information content is maintained when we have more than 30dB PSNR.

Table 5.1 gives us PSNR, MSE and MAE estimation on the video of moving disk with respect to adding noise standard deviation $\sigma = 50$ and value of λ.

λ	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$	$\lambda = 25$	$\lambda = 50$	$\lambda = 100$
PSNR	38.11	39.11	34.97	30.80	28.29	25.90
MSE	10.04	7.98	20.73	54.11	96.40	167.09
MAE	2.37	0.49	0.68	1.14	1.78	2.81

Table 5.1: PSNR, MSE, MAE comparisons, the noisy video of moving disk for $\sigma = 50$.

Have a look on this simple volume test, it can be seen that the smoothing process is more efficient when λ increases. By evaluating the standard images (video of a noisy moving disk) in terms of PSNR, MSE and MAE, experimental results prove that our proposed method is efficient in denoising application. As λ increases, the fainter results we get, the bad quality of solution gives us the small value of PSNR. But in any case, the results of PSNR are near 30dB that could be acceptable for image denoising. The values of MAE in 5.1 let us know how the noise is removed more successfully with larger λ.

Moreover in the chapter, the PSNR criterion is taken into account to compare with different testing on standard deviation σ, not only $\sigma = 50$. In the following Table the performance evaluation is carried out in terms of PSNR between corresponding denoised results and original image of moving disk video:

σ	Input PSNR	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$	$\lambda = 25$	$\lambda = 50$	$\lambda = 100$
$\sigma = 5$	41.14	51.12	39.28	35.02	30.81	28.29	25.90
$\sigma = 10$	38.13	49.98	39.26	35.02	30.80	28.29	25.90
$\sigma = 25$	34.14	44.09	39.22	35.00	30.80	28.29	25.90
$\sigma = 50$	31.14	38.11	39.11	34.97	30.80	28.29	25.90

Table 5.2: PSNR estimation, the noisy video of moving disk for different values of standard deviation σ of noisy video and λ.

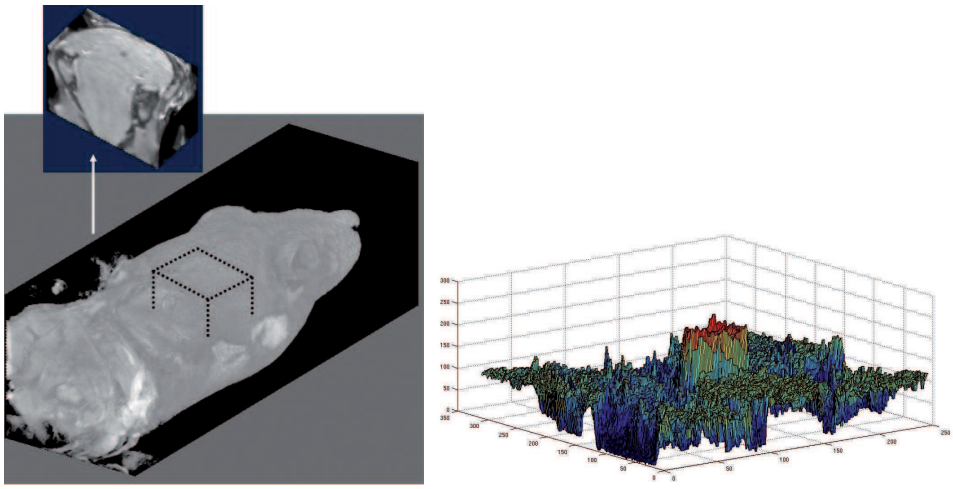
As we know, the less difference of two images is, the larger PSNR we got. And with the large value of σ we will have the small input PSNR, respectively. Generally, for $\lambda \geq 10$ we have the difference between denoised image and the original one is virtually unchanged with a fixed σ.

5.1.1.2 Numerical Tests on 3D original Volume - MRI Cerebellum data

MRI of mouse brain image (cerebellum) often consists of random noise that does not come from tissues but from other sources in the scanner machine during acquisition. The noise of an image gives it a gray appearance and mainly the noise is evenly spread and more uniform. In such a situation it is very difficult to diagnose the

particular disease for biomedical researchers, so it is necessary to remove the noise from the image.

We previously described this special MRI model in chapter 1, biologists give us a full 3D MRI volume of cerebellum. It is the stack of 104 MR images with size 341×110 pixels, that have been introduced in the chapter 1.

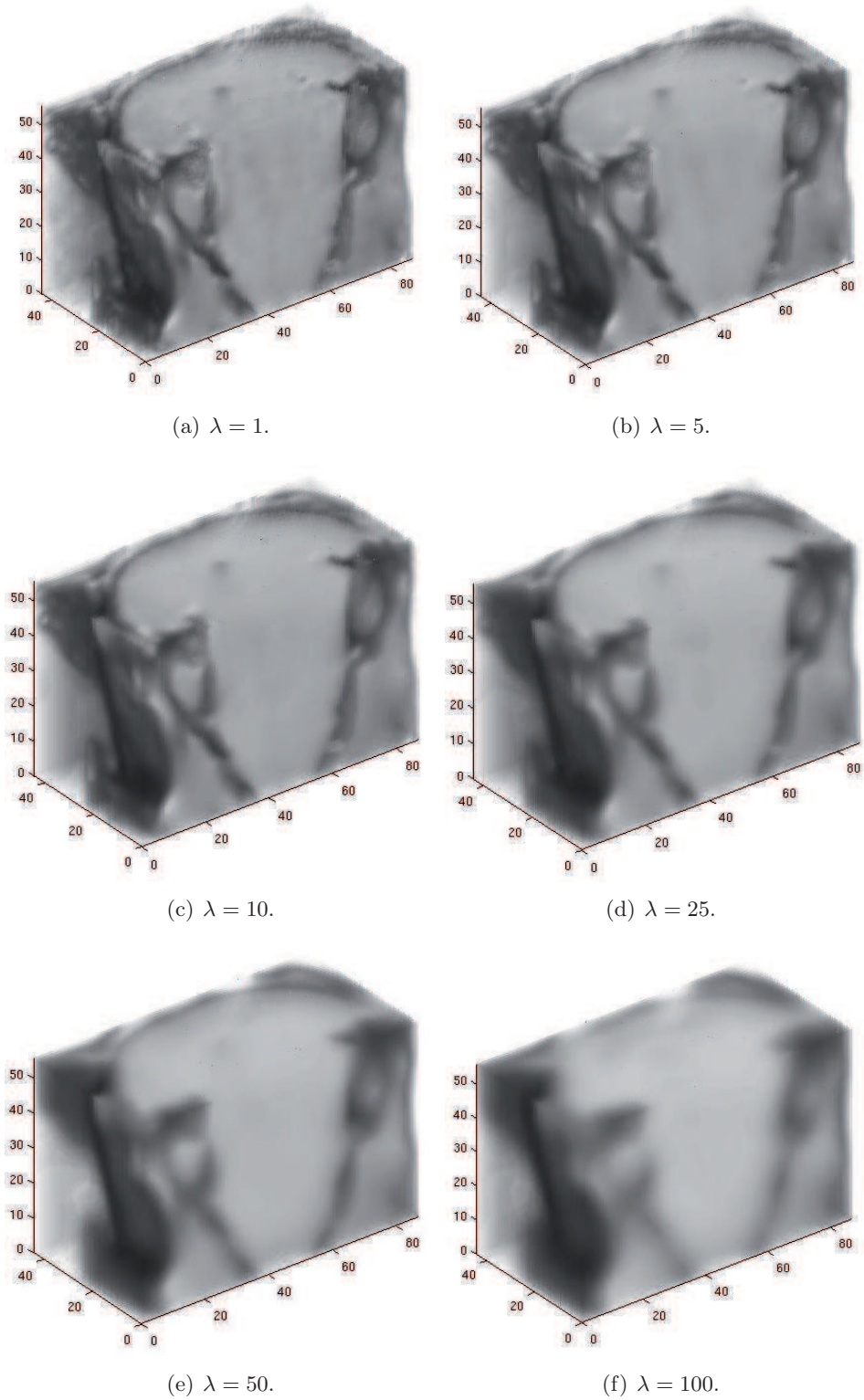


(a) Small cutting cerebellum data. (b) Surface data, the 5^{th} slice.

Figure 5.5: MRI noisy mouse brain data (cerebellum).

In this case, the "original" brain volume is noisy. We performed numerical tests for ROF2 model with different values of λ and the stopping criterion has been set to a maximal number of iterations $itmax = 5000$. These algorithms are implemented with MATLAB software version R2011.

For more details, we performed a test also on the small part of this brain volume (which is shown originally in the Figure 5.5(a)). Experimental results are presented in Figure 5.6. Looking at the numerical results in Figures 5.6, similarly to the previous noisy moving disk tests, we can note that as the more λ is larger, the fainter we obtain. For this 3D volume, we get the same visual evaluation as the video of moving disk. However, the noisy cerebellum is well denoised with isotropic method (section 3.1 chapter 3).



(a) $\lambda = 1$. (b) $\lambda = 5$.

(c) $\lambda = 10$. (d) $\lambda = 25$.

(e) $\lambda = 50$. (f) $\lambda = 100$.

Figure 5.6: Approximated solutions, smooth component of MRI Cerebellum data. Tested for different λ.

The delicate point is the tuning of parameter λ. Obviously if λ is too small we may not remove enough noise. In Figure 5.6, the value of $\lambda = 1$ is not a good choice. On the other hand, if λ is too large, the scheme will remove too many features of the image. The larger next λ presented smooth results, but the visualized solutions give more and more faint feeling than expected. Since in practice, there is no ground truth image to compare to, tuning the parameter λ is time consuming. Tuning λ often relies on experience and visual inspection. We do not have yet any automatic way for choosing λ as far as we know, so we choose λ in a reasonable range without being precise about the choice. For the consideration of this cerebellum volume of medical images, this is a very significant issue.

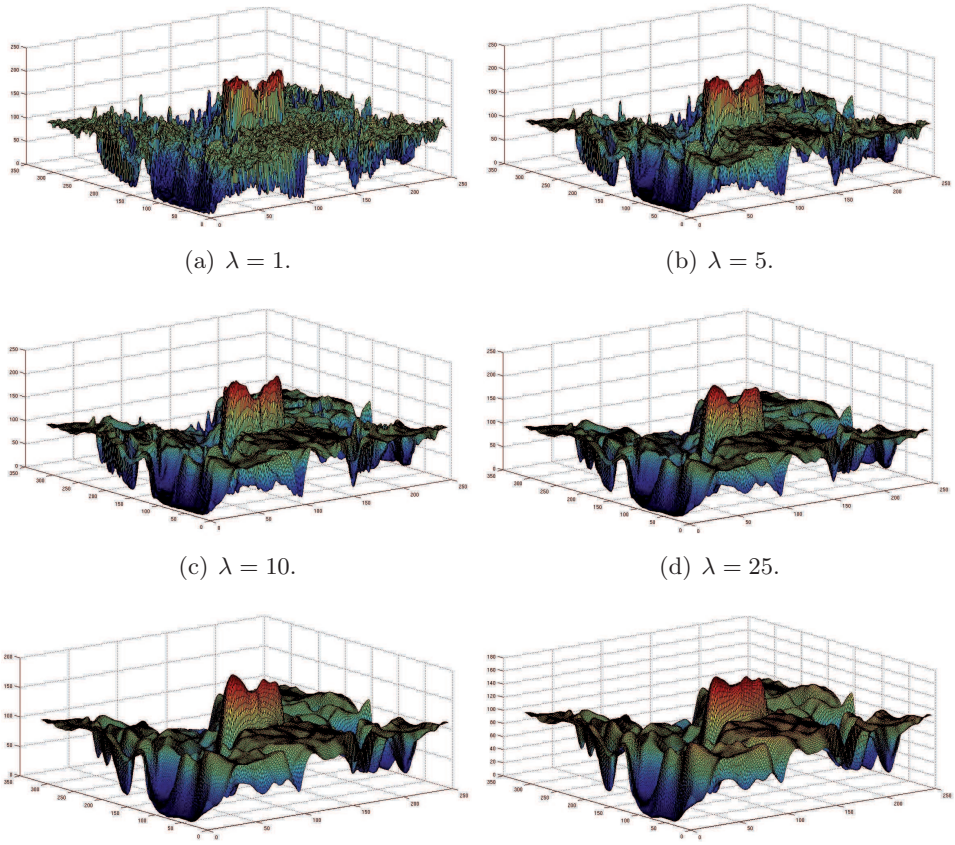
By far there is no criterion of image quality evaluation that can be accepted generally by all. In the context we validate the comparison of cerebellum by evaluating the standard images in terms of PSNR, MSE and MAE. The performance of different values of λ testing is compared by computing the error criteria MSE, MAE and PSNR of the noisy image and the denoised image. Table 5.3 gives us the parametric evaluation for different values of λ, experimental results prove that our variational method is efficient. In the sequel, we note that all values PSNR, MSE and MAE in the Table are computed on the whole brain volume.

λ	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$	$\lambda = 25$	$\lambda = 50$	$\lambda = 100$
PSNR(dB)	40.07	34.33	32.61	30.96	30.20	29.82
MSE	6.39	24.00	35.67	52.13	62.16	67.80
MAE	1.97	3.32	3.93	4.74	5.21	5.46

Table 5.3: PSNR, MSE and MAE comparisons of ROF2 isotropic method, the cerebellum, up to λ.

On Table 5.3, we perform comparison after denoising with isotropic algorithm for different values of λ in problem (P_{ROF2}). As mentioned previously, we do not have original image of cerebellum to perform comparisons. Therefore, some denoised quality measurements are computed with the original volume (reference image) we received from biologists. Our obtained results are computed and evaluated relatively in this case. It can be seen that the mathematical results obtained from the PSNR computation and the experimental results shown in the Figure outputs 5.6 match closely, and while λ increases arbitrarily, the smoothing processing is more efficient. By the compared PSNR Table 5.3 we could choose some suitable values $\lambda = 25$ or 50 with PSNR measurements around 30dB.

In addition, the Figure 5.7 shows us an arbitrary slice taken from denoised volume. We present the cutting surface comparison for different values of λ. We have seen that the larger value of λ gives the smoother surface.



(a) $\lambda = 1$. (b) $\lambda = 5$.

(c) $\lambda = 10$. (d) $\lambda = 25$.

(e) $\lambda = 50$. (f) $\lambda = 100$.

Figure 5.7: Cutting surface viewings of cerebellum results in Figure 5.6, the 5^{th} slice.

The second order variational model via BV^2 space is quite efficient for image denoising, and it seems a good application to 3D MRI volumes. Although our methods are still slow, obtained numerical solutions are acceptable. As in [38], this model is much better than the classical ROF model (P_{ROF}), it makes *stair-casing effect* disappear, contour shapes are well preserved than other denoising methods ...

5.1.2 Application to 3D image texture extraction

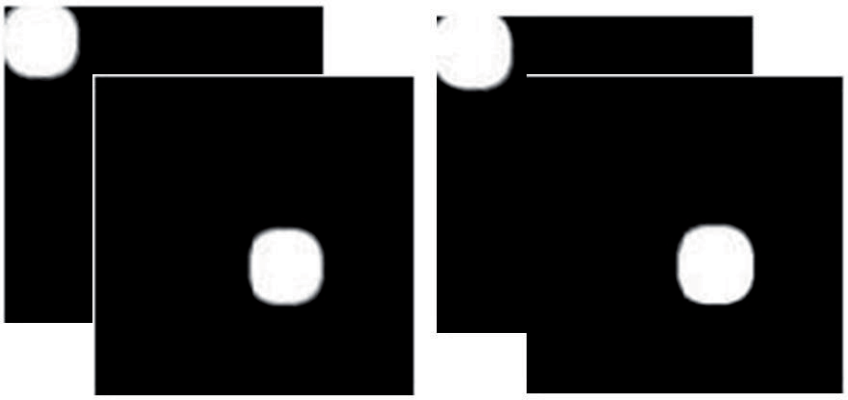
While extracting texture, we hope to keep most of texture and noise in the oscillating part while many contour shapes disappear. Based on the idea from [40], a local anisotropic model is then performed.

The so-called "isotropic" method ROF2 is applied to numerical results for different values of λ, we then choose the λ large enough to remove noise, preserve contour and texture in the oscillating part. Then a gradient calculation can be applied on the smooth image or any classical contour detector. The local anisotropic algorithm is then applied to make contour shapes disappear from the oscillating component. This section presents some numerical results that we get from local anisotropic al-

gorithm 4 or 5, performed on 2D+Time moving disk model and small cerebellum volume of 3D mouse brain data sets.

5.1.2.1 Numerical Tests on 2D+T Video - Moving Disk

A quite simple test is performed in order to give a quick comparison between isotropic and local anisotropic algorithms. We tested with $\lambda = 20$ and 5000 iterations (that should be increased in future). Figure 5.8 gives results for the test without adding noise into the video.



(a) BV^2-isotropic part. (b) BV^2-local anisotropic part.

(c) L^2-isotropic part. (d) L^2-local anisotropic part.

Figure 5.8: ROF2 test on moving disk video without noise, $\lambda = 20$.

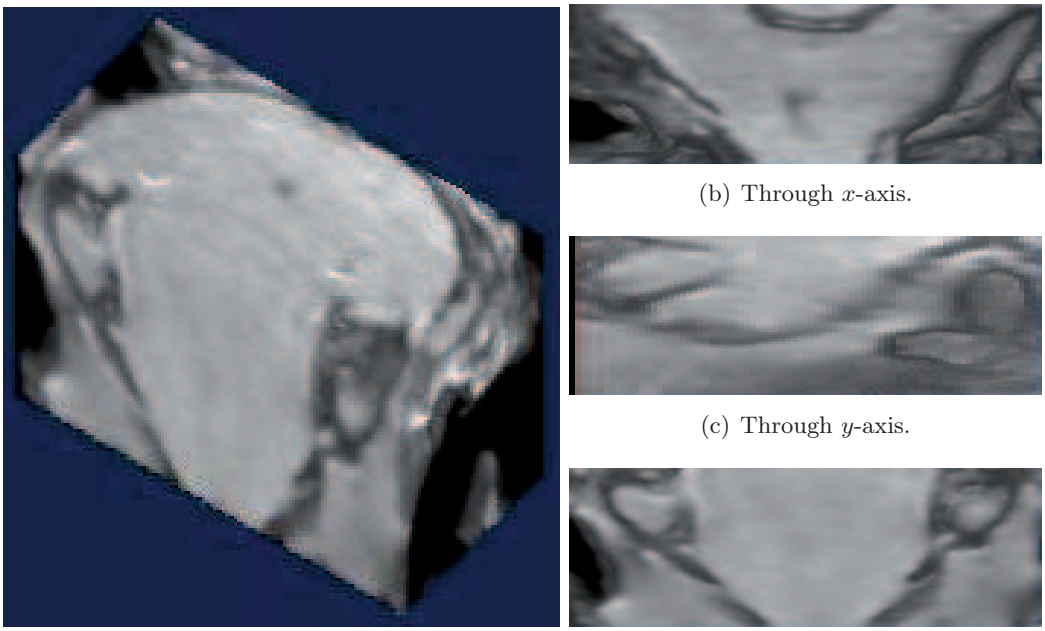
It can be seen that contour shapes on BV^2 component are well preserved using the local anisotropic scheme. For example the simple video moving disk, the isotropic method gives oscillating component that contains geometrical information. In this case we just have contours because the original image does not include noise and texture information. For the local anisotropic method, it helps us to save these geometrical characteristics to the cartoon component. After defining contour positions step, we perform the hessian operator rotation and anisotropic algorithm is applied locally, and we obtain result on Figure 5.8(d). We then get rid of all

horizontal/vertical and diagonal treated contour shapes in the L^2-component. This example is performed just only for $\lambda = 20$, proves how to extract contour from oscillating component, namely they appear in the BV^2-component meanwhile the isotropic algorithm gives the opposite thing.

Testing on video of moving disk convinces us to perform next our MRI volumes in that case. In practice, we have difficulties with these MRI data: they contain noise, texture and contour shapes together in the oscillating component. This is a truly challenge for our research. Each model can be specially considered that expressed in the very next section.

5.1.2.2 Numerical Tests on 3D original Volume - MRI Cerebellum data

Similar to the method used for the video of disk before, we perform the local anisotropic algorithm to our cerebellum 3D images. We extract the textures from the whole noisy volume. We show again cerebellum data from different directions in Figure 5.9.



(a) The whole 3D volume.

(b) Through x-axis.

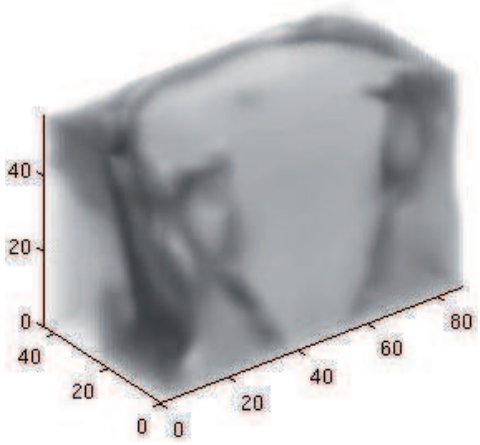
(c) Through y-axis.

(d) Through z-axis.

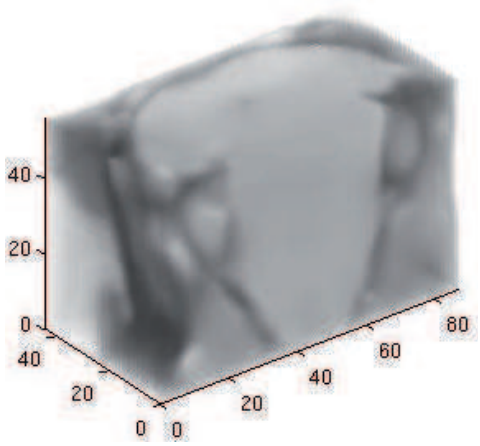
Figure 5.9: Noisy volume of cerebellum data.

For the cerebellum data, the isotropic algorithm ((P_{ROF2}) model) is applied to original image with $\lambda = 25$. It is difficult for us to view how the whole volumes represent during process, so the cross-direction of each numerical result can be performed for better visualization, Figure 5.5.

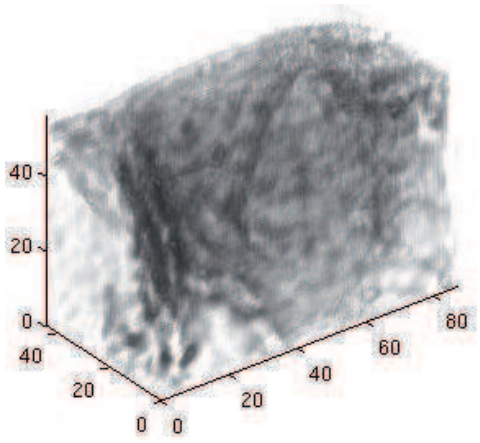
We see on Figure 5.10 that the contour shapes on BV^2 component are preserved using local anisotropic model. Oscillating part after isotropic method presents a lot of remaining dynamic information. And they disappear after local texture modification of the anisotropic method.



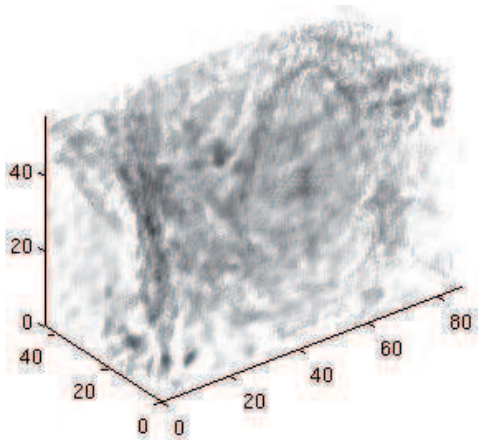
(a) Smooth solution, isotropic model.



(b) Smooth solution, local anisotropic model.



(c) Oscillating term, isotropic model.



(d) Oscillating term, local anisotropic model.

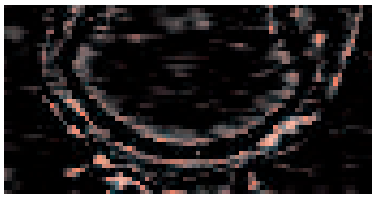
Figure 5.10: Compare solutions (visual criterion), $\lambda = 25$.

Pay attention that the difference between these two methods is emphasized as in the contour shapes (as we mentioned the testing of moving disk in Figure 5.8). That's why here, although BV^2-model keeps the contour information much better than BV-model (ROF problem) [41]; the local anisotropic algorithm gives us the more efficient contour shapes than isotropic algorithm. We may check what happens precisely on the oscillating component of an arbitrary slice taken from these 3D results, which are shown as in Figures 5.11 below.

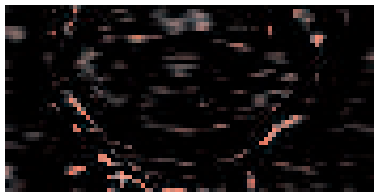
In fact, in the process of finding contour positions within the cerebellum, there are too many positions that are found: it's more than 1.000.000 contour voxels that should be performed one by one in the algorithm of section 3.2.3. Therefore, to save the CPU processing time with so many voxels, each contour voxel was only performed by the isotropic algorithm with very small number of iterations, in this case is $itmax$ is equal to 10 with every voxel performance. For this reason, the results

(a) Smooth solution, isotropic model.

(b) Smooth solution, local anisotropic model.



(c) Oscillating part, isotropic model.



(d) Oscillating part, local anisotropic model.

Figure 5.11: One visual slice, for $\lambda = 25$.

are not optimal. We can see the oscillating parts Figures 5.11(d): some contour shapes are not preserved successfully. This should be improved in the future so that almost the contour shapes are well preserved, each contour voxel being performed with enough increasing $itmax$, depending on the CPU time process.

Moreover, the process of finding contour positions is also important. The threshold parameter selection step to determine contour position by hard or soft thresholding function also affect our results. In the case of local anisotropic application for the video of moving disk and volume of cerebellum above, we have used here very simple methods, where the threshold parameter is chosen small enough and hard thresholding step are included in the algorithm. Some methods for contour shape determination are included in the thesis as a useful application: we give more details in Chapter 6. We also refer to some methods of threshold determination in the Appendix D.

In the rest of this chapter, we present some conclusions about 3D texture extraction strategies. The local anisotropic method is promising in texture identification. We hope to keep all the textures while many contour shapes of image disappear in the oscillating part. In what follows, the dynamic of image is included in the BV^2-term together with contour shapes. We emphasized here that we did not want to compare these two methods in denoising process: the point is that we want to recover the textures as much as possible.

The local anisotropic method gives very promising results in texture extraction process [41], some disadvantages remain behind. Texture component still contains a lot of geometric information: this comes from the thresholding step before applying rotation the Hessian operator. Moreover, the contour shapes we recover also depends on this thresholding process, especially in the case where the image is very noisy. It is difficult to get rid of all geometrical information and avoid loosing texture part in

order to get the results without noise, even when we perform locally to locate texture inside it. Furthermore, future works will include the development of this algorithm, which can be deduced by considering other MRI images, other 3D volumes.

5.2 Wavelet Transform and Denoising

5.2.1 3D Inverse Discrete Wavelet Transform - Mallat's Algorithm

As in the chapter 4 of wavelet description, we summarize here the idea of the discrete wavelet transform (DWT) again. Three dimensional DWT decomposes the image in first level into eight sub-volumes represented by the approximation and detail coefficients. The next level (denoted by $j > 0$) of DWT decomposition is available from the approximation coefficient A_j^d at the lower level. The approximation contains information of lower frequency components meanwhile the detail coefficients represent information of high frequency components in 7 spatial directions.

Once we get the discrete wavelet coefficients, we need to reconstruct them back into the original image. In order to do this, we utilize the inverse wavelet transform. In this section, we shall study how to get the discrete inverse wavelet transform (IDWT) from the DWT using filter bank theory.

For the purpose of denoising image, a synthesized volume V^* should remove high-frequency while preserving edge information as much as possible, if we compare with the noisy data V. The use of wavelet transform to denoise data is accomplished by applying a wavelet transformation to the noisy data, thresholding the resulting coefficients which are below some values in magnitude, and then inverse transforming to obtain a smoother version of the original data. This procedure is shown as follows:

Wavelet Denoising Procedure:

The general denoising procedure involves three steps. The basic version of procedure follows the steps described below:

1. **Decompose: Calculate the wavelet transform of the noisy image**
   - Choose a wavelet basis.
   - Choose a scale s ($s \geq 1$); the original image is supposed to be at the scale $s = 0$.
   - Compute the wavelet decomposition of image at the scale s, as in section 4.1.1.2.
   - Output: All the approximation coefficients and detail coefficients of image at every scale j, $1 \leq j \leq s$.
2. **Thresholding: Modify the noisy wavelet coefficients according to some rule**

To manipulate all coefficients of wavelet octants, we must perform this operation. Hard thresholding deletes all coefficients that are smaller than the threshold t and keeps the others unchanged. The coefficients whose absolute values are lower than the threshold are set to zero. On the other hand, the soft thresholding deletes the coefficients under the threshold, but scales the ones that are left. In practice, hard threshold or soft threshold is applied, they can be expressed as following, by functions V^{soft} and V^{hard} respectively:

$$V_t^{\text{soft}}(x) = sign(x)(|x| - t)V(|x| > t) \tag{5.4}$$

$$V_t^{\text{hard}}(x) = xV(|x| > t) \tag{5.5}$$

During the modification of wavelet coefficient, the threshold t selection is important. One of the most well-known rules for the second step is *soft thresholding*. The main idea is to substract the threshold value t from all wavelet coefficients larger than t, arising from the standard discrete wavelet transform and to set all other coefficients to zero.

Threshold plays an important role in the denoising process. A small threshold value will retain the noisy coefficients whereas a large threshold value leads to the loss of coefficients that carry image details. Normally, hard thresholding and soft thresholding techniques are used for such denoising process. Hard thresholding shrinks the coefficients above the threshold in absolute value. This thresholding step is a nontrivial task. We shall discuss more details in Appendix D.1.

Output: All new detail coefficients after thresholding performance at every scale $j, 1 \leq j \leq s$.

3. **Reconstruction: Compute the inverse transform using the modified coefficients**

Compute wavelet reconstruction using the original approximation coefficients of scale s and the modified detail coefficients of scales from 1 to s. We also perform similarly to the discrete 3D wavelet reconstruction scheme/diagram 4.5 in section 4.1.2.2.

Output: After this step, we obtain the denoised result is reconstructed by wavelet transform.

5.2.2 3D Wavelet Denoising ="2D Wavelet Denoising + 1D Wavelet Denoising"

Regarding to experimental demonstration from [51], the 3D wavelet denoising can also be implemented by 2D wavelet denoising on slices followed by 1D wavelet denoising. A volumetric denoising technique is then performed by a separable 3D wavelet

transform (WT), namely, a "2D WT plus 1D WT scheme". Since 2D wavelet transform is readily available in [33], the "2D wavelet transform plus 1D wavelet transform" scheme is aimed to be more efficient than three 1D wavelet transforms (3D Wavelet Transform performance).

The separable 3D wavelet transform can be carried out by sequentially applying 2D wavelet transform to the slice images (at transaxial planes) of the volume data set, followed by 1D wavelet transform on the columns (along the transaxial axis) of the reassembled 3D array. They proposed another method for wavelet reconstruction (wavelet denoising), and the scheme is referred to be "2D Wavelet transform + 1D Wavelet transform" which is sketched in Figure 5.12 follows, its implementation involves manipulation of the 2D data slices and 1D data columns.

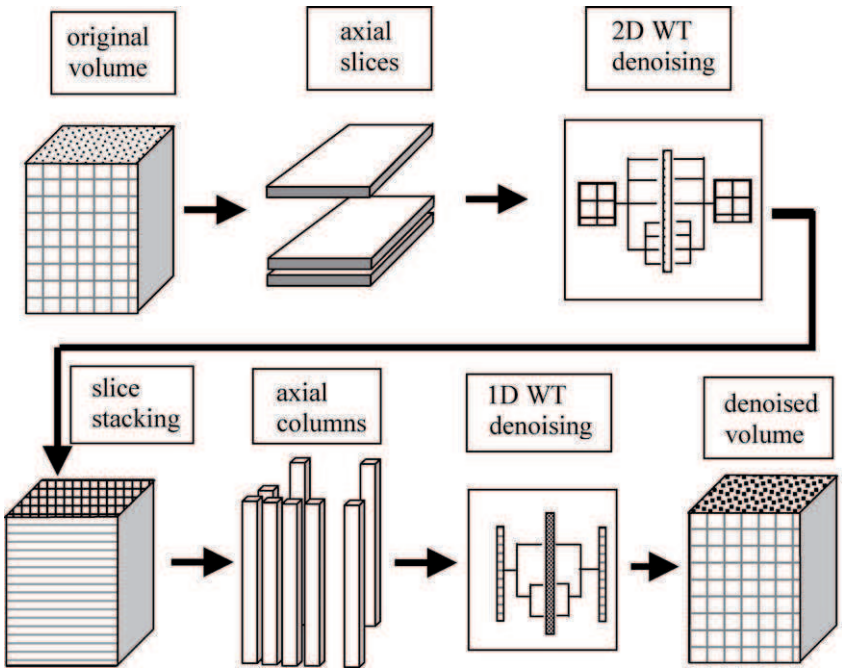


Figure 5.12: 3D wavelet denoising by a "2D WT plus 1D WT"; scheme reproduced from [51].

In the scheme, 2D wavelet denoising was applied to z-axis slices firstly. Then, the denoised slices were stacked, which was followed by 1D denoising along the stacking direction. Next, the soft or hard thresholding strategies can be performed for denoising processes. Let find out more details of the scheme.

Let denote again $V(i, j, k)$ is the 3D noisy volume data that need to be reconstructed; $S_k(i, j)$ represents the 2D image of k^{th} slice in volume and $C_{m,n}(k)$ denote the (i, j) column in the 3D array $V(i, j, k)$.

As wavelet scheme described in Figure 5.12 we express:

1. $S_k(i, j) = V(i, j, k)$ at k is fixed;
2. $S'_k(i, j) =$ 2D denoised image of $S_k(i, j)$;

3. $V'(i,j,k)$ = stack of all $S'_k(i,j)$ for $1 \leq i \leq P$;

4. $C_{m,n}(k) = V'(i,j,k)$ at (m,n) are fixed pixels;

5. $C'_{m,n}(k)$ = 1D denoised image of $C_{m,n}(k)$;

6. $V^*(i,j,k)$ = stack of all $C'_{m,n}(k)$ for all $1 \leq m \leq M; 1 \leq n \leq N$.

We need to give explanations for step 2 and 5. They are the one (two) dimensional wavelet denoising operator, which is performed respectively steps:

- Wavelet Decomposition (1D or 2D): to get the coefficients.
- Coefficient Thresholding.
- Wavelet Reconstruction.

These processes can be applied simply because they are readily available in MATLAB Wavelet toolbox [33]. These are the ones which the Wavelet denoising procedure in section 5.2.1 can be implemented in a similar way for the whole 3D volume data.

During this method, adaptive threshold calculated from the slice images, namely, in step 2, thresholds for 2D wavelet transform are chosen dependently on each 2D slice, and the same for 1D wavelet transform in step 5. We cite here Appendix D to adopt threshold selection for the coefficient thresholding step.

As previously mentioned, in [51] this 3D denoising scheme, it is based on the assumption of separable wavelets of "2D plus 1D" scheme. The non-separable 3D wavelet in section 5.2.1 may improve the result but it is more computation complexity. This gives us the 3D volumetric denoising and multi-resolution representation with fully automated at a high speed.

5.2.3 3D Inverse Discrete Wavelet Transform - "À trous" Wavelet Transform

As already mentioned in chapter 4, the "à trous" algorithm computes the low pass filter at each level by inserting zeros between each of the filter's coefficients. The high pass coefficients are then computed as the difference between the low pass images from the two consecutive levels.

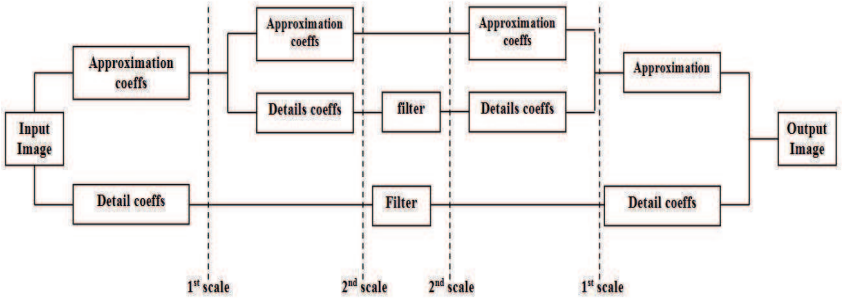


Figure 5.13: Wavelet-based filtering scheme. Example for two scales à trous wavelet transform.

To compute the inverse transform, the detail coefficients from all levels are added to the final low-resolution image. While inefficient in implementation, the "à trous" algorithm provides additional insight into the redundant discrete wavelet transform. In the descibed Figure 5.13 below, an example of two scales scheme is presented for illustrating "à trous" wavelet algorithm. Differs from the Mallat's algorithm, there are just two coefficients for unsubsampled decomposition transform: approximation and the details coefficients (the size of our image does not change in these steps). In the scheme 5.13, the "à trous" algorithm is characterized by three steps similar to the Mallat's algorithm in section 5.2.1: decomposition of image to coefficients until the final scale, thresholding the detail coefficients and apply the filter to the new coefficients (B_3-spline in this case), finally perform the wavelet reconstruction until the coarsest scale to obtain the new modified image.

5.2.4 Numerical Tests

5.2.4.1 Mallat's algorithm

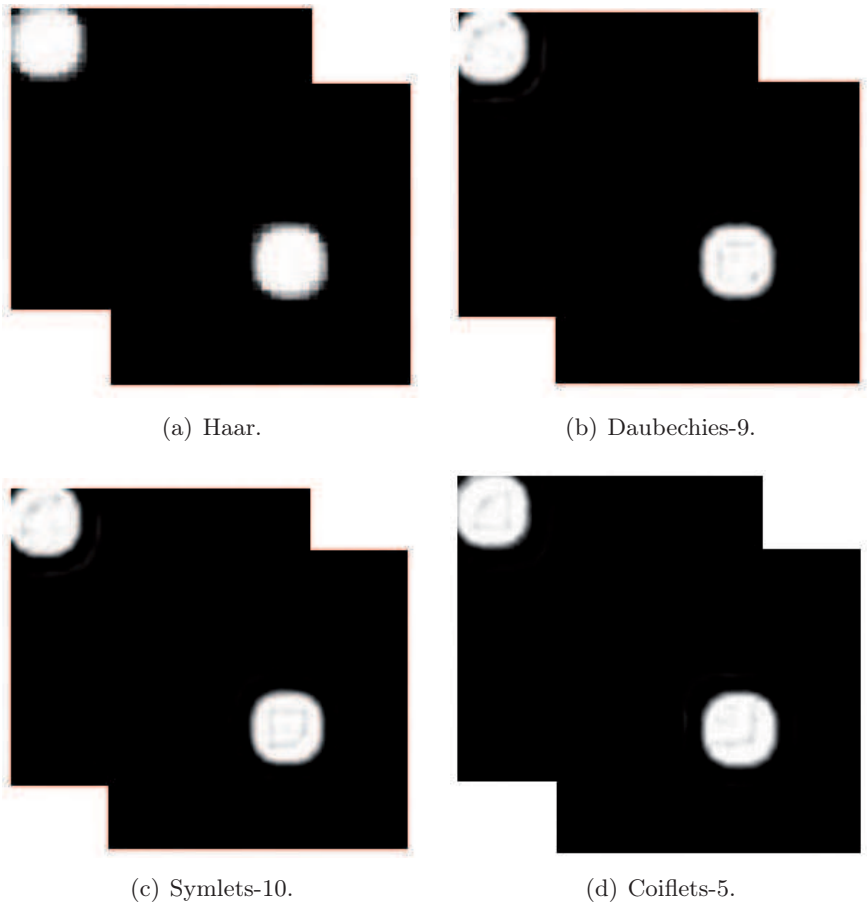
In the sequel, the Mallat's method is proposed for implementing image denoising using wavelet transform. The image is transformed into the orthogonal coefficients, the detail wavelet coefficients are modified according to the shrinkage algorithm. Finally, inverse wavelet transform is taken to reconstruct the denoised image. The algorithm has been implemented using the tools given by the source of MATLAB, the wavelet toolbox. A forthcoming update of this tool will contain all the modules used to obtain the following experiments, so that everyone will be able to reproduce the results and perform additional experiments [33].

The wavelet transform is first implemented to the video of moving disk, note that we test on image by adding noise, for standard deviation $\sigma = 10$, the noisy image and its surface plot are shown in Figure 5.2(a) and (b) respectively. It is also remarkable here that results tested with the same threshold parameter choice for all types of wavelet bases. This step is very effective because denoised results depend completely on the threshold in wavelet denoising process. In the thesis domain, the threshold is chosen from iterative method (we mention in detailed Appendix D), the selected threshold $T \simeq 20$.

The Peak-signal-to-noise ratio is then computed corresponding to each type of wavelet bases, it is clearly shown on the Table 5.4 compared between PSNRs evaluation. From the Table, we choose some bases that give large PSNR in order to present results, such as *daubechies-9*, *symlets-10* and the *coiflets-5*. See the Figure 5.14 to see how the video is denoised associated to these wavelet bases. The ability of the discrete wavelet transform to reduce distortion reconstructed video volume while retaining all the significant features present in the image volume is displayed in Figure 5.14.

Wavelet bases	PSNR	Wavelet bases	PSNR	Wavelet bases	PSNR
haar	25.6985	*db9*	**26.7245**	*sym8*	26.7556
db2	26.0985	*db10*	26.6975	*sym9*	26.6964
db3	26.5120	*sym2*	26.0985	*sym10*	**26.8740**
db4	26.4972	*sym3*	26.5120	*coif1*	26.1183
db5	26.6577	*sym4*	26.6009	*coif2*	26.6237
db6	26.7152	*sym5*	26.5968	*coif3*	26.7330
db7	26.6336	*sym6*	26.7794	*coif4*	26.7548
db8	26.6742	*sym7*	26.8663	*coif5*	**26.7836**

Table 5.4: PSNR comparison between different wavelet types, the tested noisy video of moving disk, $\sigma = 10$.



(a) Haar. (b) Daubechies-9.

(c) Symlets-10. (d) Coiflets-5.

Figure 5.14: Wavelet denoising tests, on the video of moving disk, $\sigma = 10$. We present some chosen wavelet bases that have a good PSNR evaluation: *haar, db-9, sym-10* and *coif-5*.

In addition, the volume of cerebellum is also taken to test and compare between some types of wavelet orthogonal bases: *Haar*, *Daubechies-p*, *Symlets-p* and *Coiflets-p*. And the said method is evaluated using the quality measure PSNR. The Table 5.5 we have the results compared in terms of PSNR, MSE and MAE.

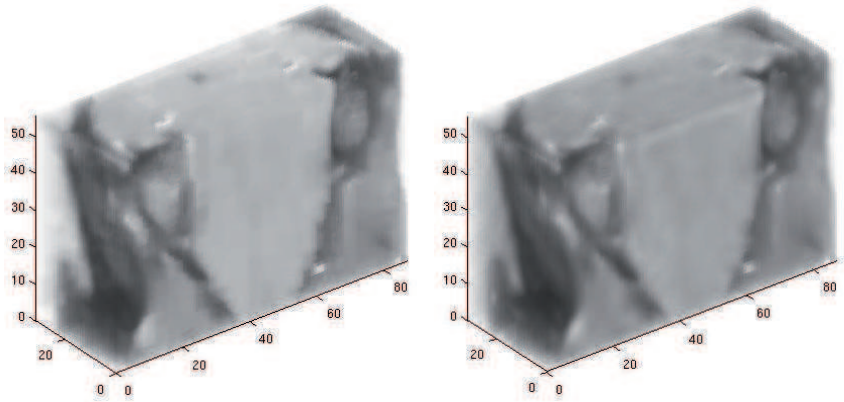
Wavelet bases	PSNR	MSE	MAE	Wavelet bases	PSNR	MSE	MAE
haar	30.51	57.82	4.31	*sym4*	31.33	47.83	4.30
db2	31.04	51.18	4.39	*sym5*	31.26	48.69	4.20
db3	31.13	50.11	4.44	*sym6*	**31.43**	46.77	4.23
db4	31.18	49.51	4.38	*sym7*	31.32	47.97	4.36
db5	**31.20**	49.27	4.29	*sym8*	31.22	49.05	4.36
db6	31.07	50.88	4.44	*sym9*	31.20	49.29	4.31
db7	31.01	51.55	4.43	*sym10*	31.33	47.84	4.37
db8	31.13	50.16	4.33	*coif1*	31.08	50.71	4.34
db9	30.65	55.94	4.57	*coif2*	**31.33**	47.82	4.34
db10	30.69	55.48	4.61	*coif3*	31.32	47.99	4.36
sym2	31.04	51.18	4.38	*coif4*	31.27	48.49	4.38
sym3	31.13	50.11	4.38	*coif5*	31.26	48.60	4.39

Table 5.5: PSNR, MSE and MAE comparison with the small cerebellum, tested for different wavelet bases.

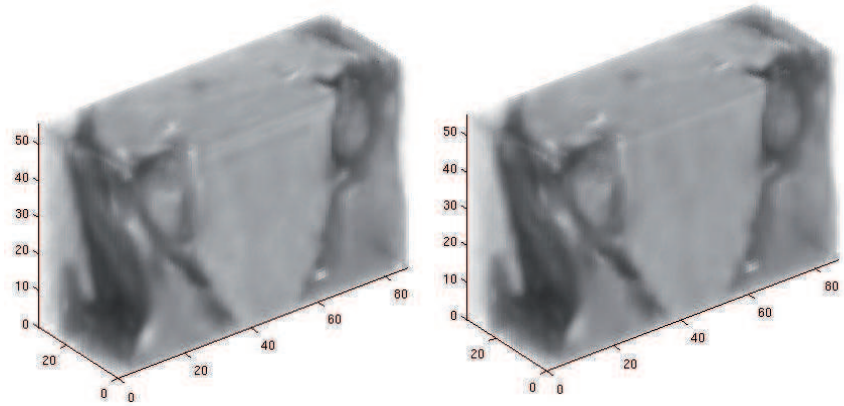
In Figures 5.15 and 5.16, some numerical tests are performed on the MRI cerebellum data, in full 3D volume and one sample visual slice respectively. The proposed 3D discrete wavelet denoising algorithm has been evaluated on the noisy volume, by visual inspection and by computing quantitative measures of the similarity between the reference image and the denoised image. The performance of the different wavelets is compared by computing the error criteria MSE, MAE and PSNR of the noisy image and the denoised image.

From the PSNR estimation on Tables 5.5 tested for the cerebellum data, we choose some examples with the large computed PSNR: *Daubechies-5* (among all *Daubechies-p*), *Symlets-6* (among all *Symlets-p*) and *Coiflets-2* (among all *Coiflets-p*) and the simplest wavelet basis *Haar*. In all the cases, the *sym6* wavelet outperforms other wavelets as can be seen from the increase of the PSNR values.

Since the algorithm of WT coding allocates more bits to the low frequencies areas, for which human eye is more sensitive, thus after decoding the high frequencies images have a low PSNR. The higher value of PSNR gives the better images. The MSE, MAE values must be low for a better quality images. Large values of PSNR and small values of MSE indicate less noise power in an image irrespective of the degradation undergone.

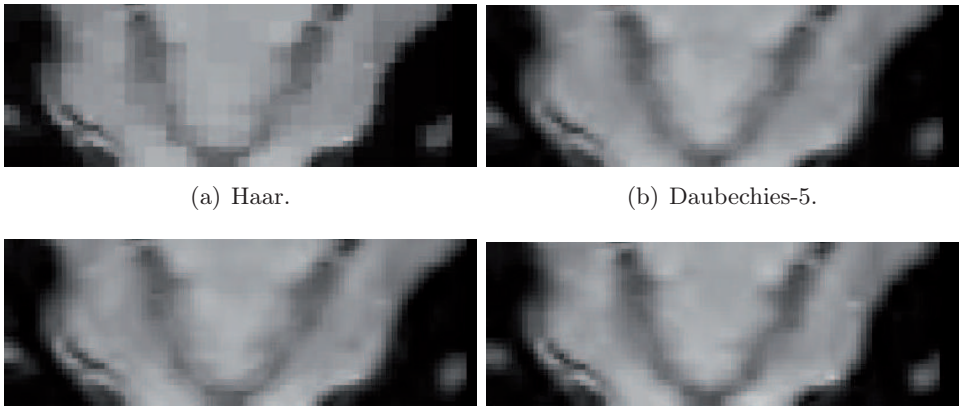


(a) Haar. (b) Daubechies-5.



(c) Symlet-6. (d) Coiflet-2.

Figure 5.15: Wavelet denoising tests, on the 3D cerebellum data.



(a) Haar. (b) Daubechies-5.

(c) Symlet-6. (d) Coiflet-2.

Figure 5.16: One visual slice of denoised images, tested on the cerebellum data.

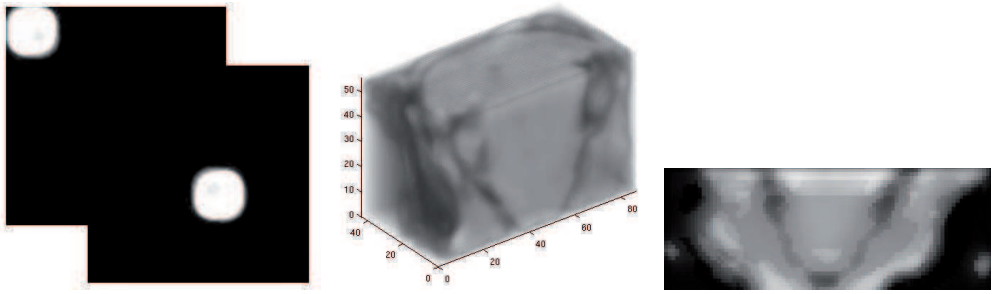
Results in Table 5.5 obtained with the popular Daubechies (*db*), Symlets (*sym*)

and Coiflets (*coif*), we can clearly see that the Haar basis (*haar*) is not appropriate for image denoising since the computed PSNR is smallest compared to all wavelet bases. Otherwise, results we have reached with the Symlets wavelet bases have a good visualisation with the large obtained PSNR. They are quasisymmetric extension of Daubechies, which make them more suitable than the classical Daubechies. About Coiflets, they are another extension of Daubechies, with vanishing moment conditions both for the wavelets and the scaling functions. They are also more symmetrical than the classical Daubechies. All these characteristics make them a little more efficient than Symlets, [44].

5.2.4.2 "À trous" algorithm

As in the scheme described in 5.13, the "à trous" wavelet transform is also performed to the cerebellum with the same input threshold value of the subsampled wavelet described previously. The tested noisy moving disk, for standard deviation $\sigma = 10$ gives us the result of PSNR=24.8838. Table 5.4 give an objective PSNR evaluation that compared to this algorithm, we have the small value of ratio gives a bad denoised image compared with the Mallat's algorithm. The Figure 5.17(a) below shows denoised video considered relatively with results in the Figure 5.14.

And in the rest of testing on cerebellum, we take a small space for displaying these results, in the Figures 5.17 as following. The reason we choose the same inputs to this type of wavelets is to compare with all above wavelet bases. However, the "à trous" wavelet does not give desirable results; in detail we got PSNR=28.56 with MSE=90.51 and MAE=6.35. In Figure 5.17(b), denoised volume of cerebellum gives us the blurry feeling of eyes and the smooth surface viewing in the second Figure also confirms the same visualisation. Results from "à trous" algorithm do not preserve edge information as in subsampled algorithm performed previously.



(a) Denoised moving disk, $\sigma = 10$.

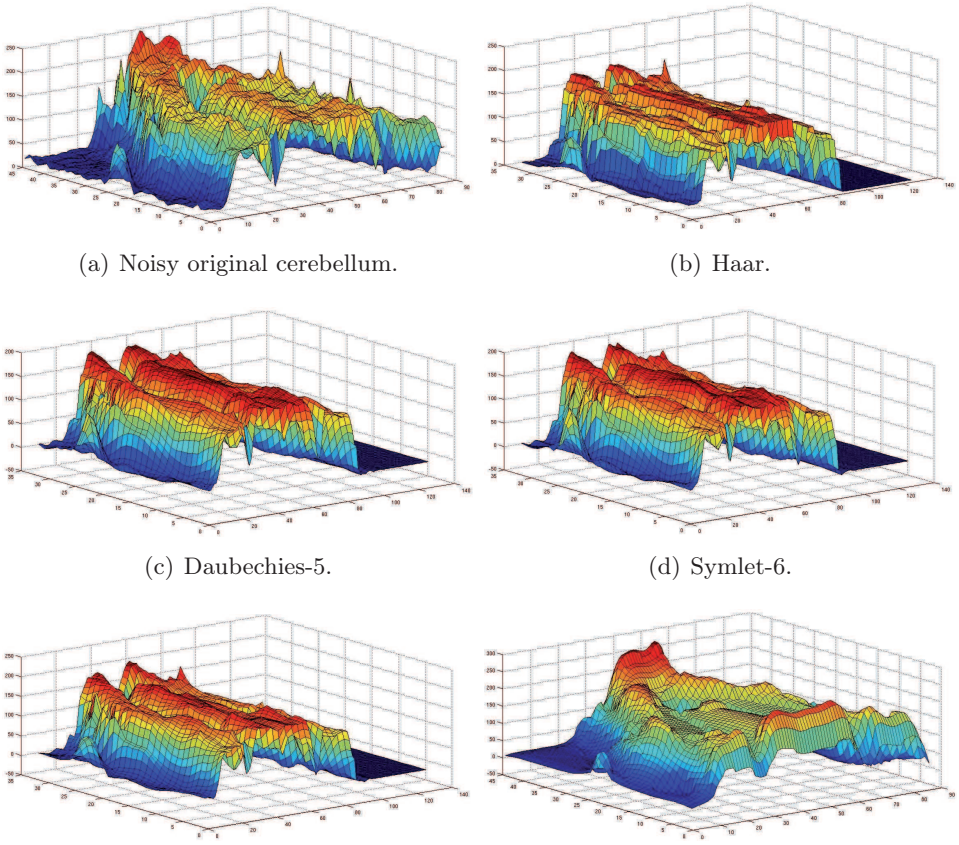
(b) Denoised Cerebellum.

(c) One visual slice of denoised cerebellum.

Figure 5.17: The "à trous" wavelet transform, results tested on the noisy disk and the cerebellum.

In the comparison that we could perform to this MRI volume image denoising using DWT is analyzed. The experiments are conducted to study the suitability of different wavelet bases. Figures in 5.18 sketch out the surface viewing of each

result corresponding to wavelet types (*haar, daubechies-5, symlets-6* and *coiflets-2*), compared to the result obtained by the the "à trous" algorithm in Figure 5.17.



(a) Noisy original cerebellum.

(b) Haar.

(c) Daubechies-5.

(d) Symlet-6.

(e) Coiflet-2.

(f) By the "à trous" algorithm.

Figure 5.18: Surface viewing of the original and denoised image by wavelet transform, tested with the Mallat's and the "à trous" algorithm.

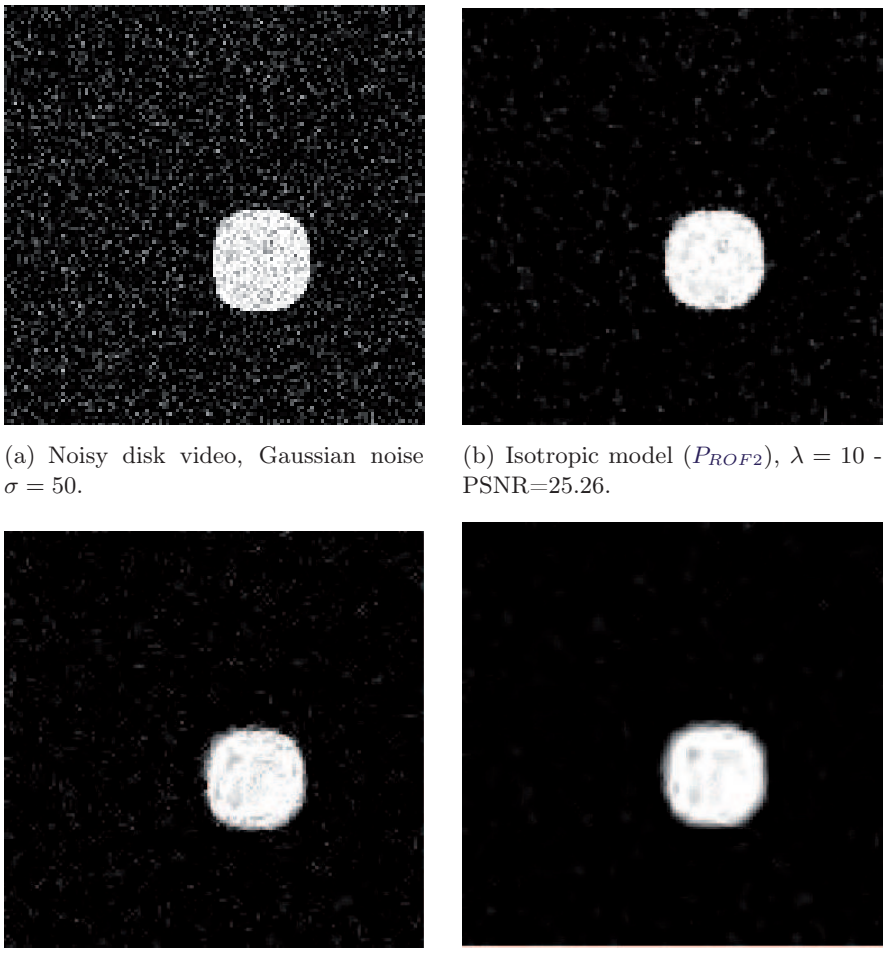
5.3 Comparisons

Wavelet transform, compare to the denoised variational methods, the PSNR Tables 5.3 and 5.5 both give us how good the visual quality of results are. Although more subjective in nature, the Figures 5.15 show that wavelet denoising frame is also superior. Inside the section we have an effective approach in terms of PSNR and visual quality. The variational method (that based on the isotropic and/or local anisotropic model from solving problem (P_{ROF2})) and wavelet techniques are the different algorithms that we have to compare based on different design and computational strategies.

The wavelet transform provides high PSNR values and can remove noise very efficiently, meanwhile the variational methods of (P_{ROF2}) also remove noise successfully, but in that case the result quality depends on the chosen value λ, this regularization parameter decides how much noise can be reduced. Comparing model (P_{ROF2}) with wavelet denoising is delicate, as there are free parameters of the algorithm which can affect their performance. Namely, the choice of λ is critical, we have verified experimentally that λ is large enough for the smooth solution but not too small to guarantee the denoising process.

Wavelet-based methods hold the current state of the art in image processing. A key feature of wavelet-based method is their ability to remove noise while keeping important image detail, such as contours. Whereas, our variational methods of model (P_{ROF2}) denoising is able to reduce noise, but does so while preserving image contour information, which are important.

In this section we provide some examples to illustrate the comparison between these methods therein. The first example is for testing with the video of moving disk adding Gaussian noise inside, artificial noise is added to a clean video that is described in Figure 5.1, and the density of noise is large by choosing $\sigma = 50$. In the Figures 5.19 we show some numerical solutions of the variational models and wavelet thresholding schemes, inside problem (P_{ROF2}) we choose a fixed fitting parameter $\lambda = 10$ large enough, meanwhile wavelet denoising tests can be performed with *Daubechies-8* bases. We note here that the threshold is selected to reach the best PSNR performance, the same threshold are used for standard thresholding hard/soft schemes therein.



(a) Noisy disk video, Gaussian noise $\sigma = 50$.

(b) Isotropic model (P_{ROF2}), $\lambda = 10$ - PSNR=25.26.

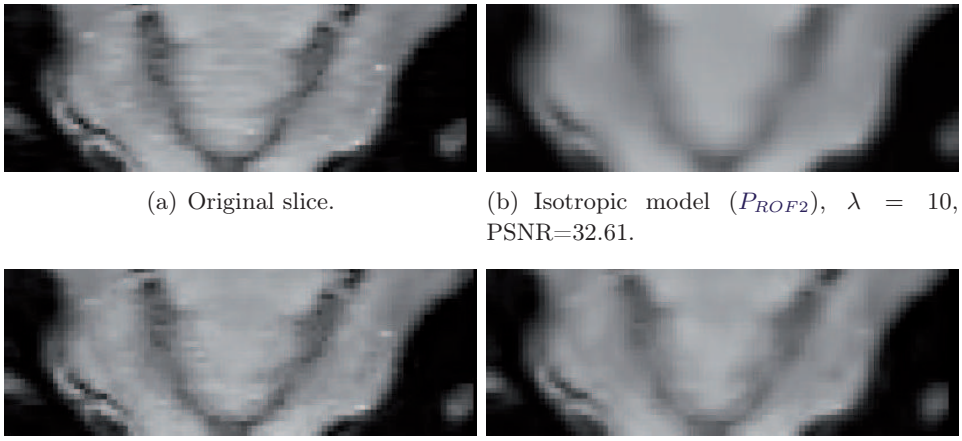
(c) Wavelet hard thresholding-*db8*, PSNR=23.67.

(d) Wavelet soft thresholding-*db8*, PSNR=25.56.

Figure 5.19: One visual slice extracted from the video of moving disk tests.

In the second example, we test for the original MRI brain, which appears quite noisy. In the previous section 5.2.4 we choose the good wavelet denoised result that returns the best PSNR, which is performed with *Symlets-6* bases. A conventional criterion is that larger PSNR signifies better performance. From Figures 5.20, it is remarkable that the obtained PSNRs of our methods are acceptable for denoising application, the hard thresholding gives better PSNR performance, however the soft thresholding gives better visual quality. Because in the case of MRI volumes we introduced at the beginning chapter 1, we cannot judge the performance by examining the PSNR as we do not have a noise-free image in which we can compare. Remind that we just have a noisy original data and try to remove noise while keeping geometrical structures significantly. However, by visual inspection it is evident that the denoised image, while removing a substantial amount of noise, suffers virtually no degradation in sharpness and details.

We have also carried out a detail experimental comparison of (P_{ROF2}) model



(a) Original slice.

(b) Isotropic model (P_{ROF2}), $\lambda = 10$, PSNR=32.61.

(c) Wavelet hard thresholding, *sym6*, PSNR=35.60.

(d) Wavelet soft thresholding, *sym6*, PSNR=31.43.

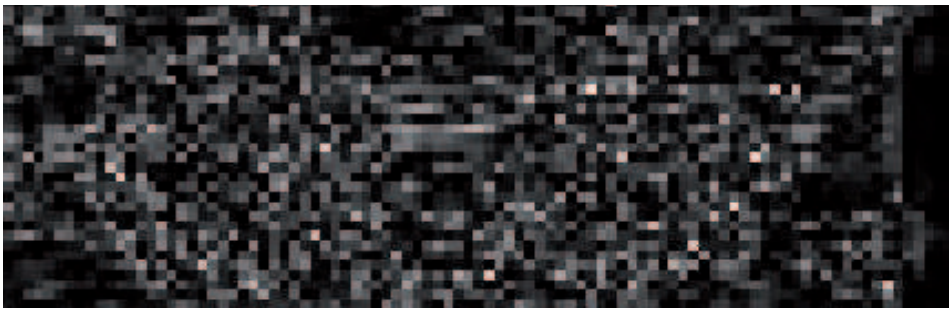
Figure 5.20: One visual slice extracted from the noisy cerebellum scan tests.

versus wavelet denoising. With variational model (P_{ROF2}), we can see that in order to obtain good denoised results (for λ is large enough), our images become to be faint, almost geometrical elements (containing texture and contours) are lost in the cartoon component. On the other hand, the approximation of wavelet performance keeps more features in denoised images, with the well chosen value of threshold parameter T (which depends upon the considered image).

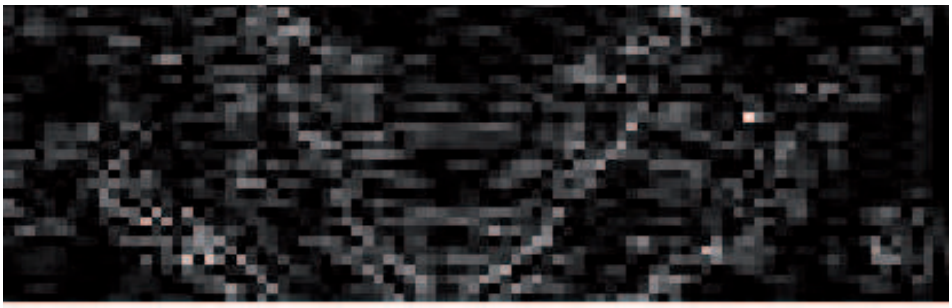
Figure 5.21 gives the representation of difference between original (noisy) and approximated data, we highlight that our methods include removing noise, restoring sharper edges and geometric features, which cannot be reflected through PSNR values. However, it seems that the wavelet soft thresholding scheme gives better edges are restored than variational model, also soft thresholding is better than hard thresholding. In addition, we use visual inspection to compare the performance in preservation of geometrical information, which is not reflected through the PSNR measurement. The work carried out here can be extended to speech denoising for sentences recorded in varies noisy environment also it can be extended for the real time image denoising. Wavelet package transform can be implemented further to achieve good performance.

In the section, we were studying the denoising application separately the variational method of model (P_{ROF2}) and wavelet thresholding schemes. Because each method has the advantages and disadvantages, BV^2-variational method gives good denoised results meanwhile the wavelet shrinkage technique reduces well edge/contour artifacts. In this situation we propose a new modified model that combines these schemes, we then have a reconstructed image has fewer oscillations near edges and noise is smoothed. A lot of references, for instance in [24], [25] etc, gave ideas of denoising algorithm based on a combination of these frameworks several years ago. It permits us to improve another approach that employs variational framework, in particular the minimization model (P_{ROF2}) plus the wavelet thresholding to re-

(a) Isotropic model (P_{ROF2}), $\lambda = 10$.



(b) Wavelet hard thresholding, *sym6*.



(c) Wavelet soft thresholding, *sym6*.

Figure 5.21: One visual slice of the difference between original and denoised volumes tested on the cerebellum.

duce oscillations, remove noise while maintain the sharpness of image. The ideas introduced here can be considered to perform in the chapter 7 and use as a post-processing technique for image denoising application. Taking advantages of both two methods (variational method and wavelet based scheme) mentioned previously, we apply them in a new model that combines two based methods the numerical results are well denoised and almost geometrical details are well preserved.

Chapter 6

Application to 3D Image Segmentation Process

Contents

Chapter 6 addresses the application to 3D image **segmentation** process that is performed on a 3D image. In particular we locate the contour information of image with both variational model and wavelet based method. From a different point of view, the contour detection is another application to the denoising application in chapter 5. Here, we deal with contour detection simulations and compare some experimental results by variational method and wavelet scheme. In particular, the difference between isotropic and local anisotropic results represent contour shapes in variational method, on the other hand, the contour detection methods using wavelet transform scheme is also studied to isolate the contour shapes of image, [55], [36], [17].

Visual perception evolved in a world of objects many of which are bounded by smooth closed contours. Especially with the three-dimensional dataset that need to be marked the locations of filaments to get the region of interest. There also exist other structures with high values (noise) or we may loose some information within the vessels through contour detection process. That is the reason why we should perform contour closing algorithms. Once the contour shapes are detected, we next provide methods of contour closing implementation. This chapter also presents two classical methods, which be applied for three-dimensional contour closing, the hysteresis thresholding [48] and a local operator based on chamfer distances methods [35], [9], [6]. We try to give an understanding of the original derivation and motivation of each algorithm.

6.1 Contour Detection strategy

Human can easily extract the contours of any objects by watching their defining set of points, but this simple and almost trivial action for the human being is a quite difficult task to be automatically performed, [52]. As a simple understanding, contour detection strategy is a general scheme of contours extraction carried out in the computer vision community. In [27], contours are distinguished from edges as follows. Edges are variations of intensity level in a gray level image whereas contours are salient coarse edges that belong to objects and region boundaries in the image. From that point of view, the contours/edges detection scheme is driven by the structure of our images. In the case of three dimensional medical images, a local context of a contour significantly affects the global saliency of the contours.

In this section, we shall study two simple contour detection methods by applying computationally variational model and the wavelet technique, that have been studied in previous chapters. Classically, there are a lot of known contour detection methods that have been developed during the past decade, however in the thesis discussion we have just focused on the methods coming from our scope of researches, that are inspired by the 3D second order variational model (P_{ROF2}) and the wavelet based scheme. These methods mentioned in the thesis maybe do not give better results comparing with some other proposed methods, however we could point out their applications in the contour detection process, and especially for three dimensional medical images, our research of interest.

6.1.1 Variational model

As the description in variational performance of model (P_{ROF2}), it is easy to recognize the contour shapes represented by the difference between smooth components of isotropic and local anisotropic algorithms. Indeed, the isotropic algorithm gives that the observed image f can be combined by:

$$f = \underbrace{C_1}_{\substack{\text{Smooth component}\\ \text{without geometrical contour}}} + \underbrace{T_1}_{\substack{\text{Oscillating component}\\ \text{contains geometrical contour}}} \tag{6.1}$$

On the other side, the local anisotropic algorithm transfers the contour elements (geometrical information) successfully from oscillating component to the smooth component, f is then represented by:

$$f = \underbrace{C_2}_{\substack{\text{Smooth Component}\\ \text{contains geometrical contour}}} + \underbrace{T_2}_{\substack{\text{Oscillating Component}\\ \text{without geometrical contour}}} \tag{6.2}$$

Hence, the "geometrical contour" of our 3D image can be extracted as:

$$\text{"Geometrical Contour"} = |C_2 - C_1| = |T_1 - T_2| \tag{6.3}$$

The contour detection process of "geometrical contour" component is next obtained as a binary image with a simple thresholding step. This step classifies pixels into two categories:

- Those for which some properties measured from the image falls below a threshold parameter T, and those for which the property equals or exceeds T.

- Thresholding creates a binary image, which only contains values 0 and 1, for instance, the object pixels (pixels of interest region) are given a value of 1 and the background pixels (pixels of the others) are given a value of 0. Binary image is created by coloring each pixel white or black depending on pixels' value.

In a fixed thresholding (or global thresholding) step, the threshold value is held constant throughout the image. This makes the choice of threshold value be the most important feature in segmentation process. The determination of a single threshold value by treating each pixel is independent of its neighborhood: a too low threshold parameter will imply a loss of information, meanwhile a too high one we will provide an image involving undesirable elements.

Several different methods for choosing threshold exist, for instance automatic thresholding, Otsu's method, Bayesian method, histogram shape-based method and so on ... In Appendix D.1, we introduce some thresholding methods that we used for the segmentation process. The choice of global (local) threshold depends on the image's region or the adaptive contour detector method.

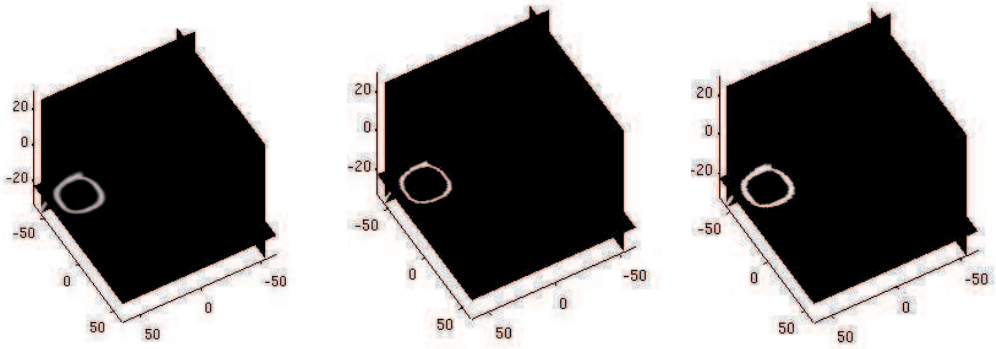
Back to the contour detection process with the variational second order methods (P_{ROF2}), we perform a thresholding strategy of "geometrical contour" to get the exact contour. During the segmentation process, we chose a fixed threshold value for every contour detection method. Table 6.1 gives the threshold parameter T computation by two classical methods: Histogram shaped-based and Automatic thresholding algorithms.

Data	Video of Moving disk	Cerebellum
Histogram shaped-based method	$T = 80.00$	$T = 15.00$
Automatic method	$T = 54.96$	$T = 8.85$

Table 6.1: Threshold Computation for the variational model.

After the threshold determination step, a hard thresholding is performed to detect the contours. We give next examples, namely the video of a moving disk and 3D stack of MRI cerebellum images (Figures 5.1 and 5.5). Results are shown in Figures 6.1, 6.2 respectively. The experimental segmentation results of variational model depend on the thresholding step. In the "geometrical contour" viewing of Figure 6.1(a) and 6.2(a), it is well known that the gradient terms can detect the boundary very well for sharp edges.

Examining 3D images of moving disk video and the cerebellum, it is clear that two segmentation methods find almost the true image contours and not respond to other image features. With automatic thresholding method, we get a good contour information; however, the detected contour is smoother and thicker than histogram shaped-based method. Although the acceptable results obtained with moving disk video in 6.1(d)-(e)-(f), we see the difficulties when used on the cerebellum images



(a) Geometrical contour.

(b) Binary image of contour, thresholding by histogram shape-based method.

(c) Binary image of contour, thresholding by automatic method.

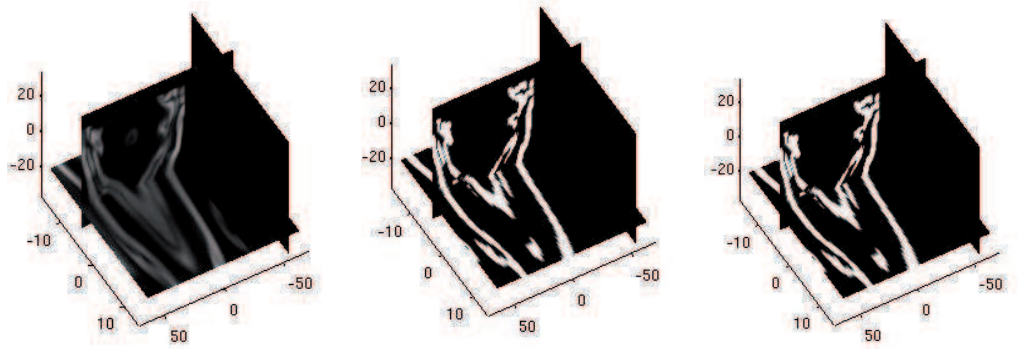
(d) Geometrical contour, one visual slice.

(e) Histogram shape-based method; one visual slice.

(f) Automatic method; one visual slice.

Figure 6.1: Contour detection for the variational method - The moving disk video: 3D and one visual slice viewing.

in 6.2(d)-(e)-(f). Since the histogram shape-based threshold is globaly selected, it seems that the value of T is not small enough for contour determination process and we loose a part of segmented information, see Figures 6.2(d)-(e). With the contour maps obtained using the difference between two smooth components from the variational model (P_{ROF2}), we may conclude that the proposed contour detection scheme is well performing since it is able to identify the in depth, curved and linear contours efficiently. However, the obtained contour shapes can be either quite thick or thin depending on the intensity across the contour and how much the image was blurred. Therefore, the non-maximum principle should be performed to reduce thick contours response to thin shapes. The non-maximum suppression section is presented in the Appendix E.



(a) Geometrical contour.

(b) Binary image of contour, thresholding by histogram shape-based method.

(c) Binary image of contour, thresholding by automatic method.

(d) Geometrical contour, one visual slice.

(e) Histogram shape-based method; one visual slice.

(f) Automatic method; one visual slice.

Figure 6.2: Contour detection for the variational method of cerebellum test.

6.1.2 Contour Detection methods with Wavelet Transform

In this section, we investigate contour detection methods related to wavelet transform. Mallat et. Al [55] studied detection using wavelet transform by Modulus Maxima Method in two dimensional case. This method is based on finding local maxima of horizontal and vertical wavelet coefficients at the first level of wavelet decomposition, and the method was simulated with a lot of wavelet functions to improve results of edge detection. Mathematical principles and applications of Mallat's method should be studied to three dimensional mentioned MRI images in the future works.

In addition, the wavelet analysis of images makes it possible to extract a new image from which we can isolate the contours [36]. The general idea of contour detection using wavelet transform is illustrated below:

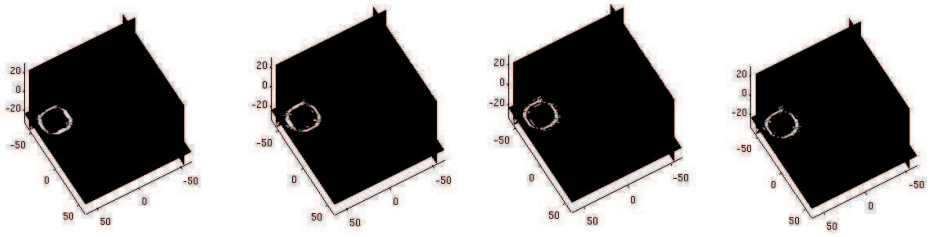
- Choose a suitable wavelet function;
- Use the function to transform images into decomposition scales;
- The wavelet detailed coefficients containing significant noisy energy are filtered out;
- Finally contours are detected from the filtered detailed coefficients.

In the wavelet decomposition scheme, while the approximation coefficients contain most of low-frequencies of the image, detail coefficients represent contours. A

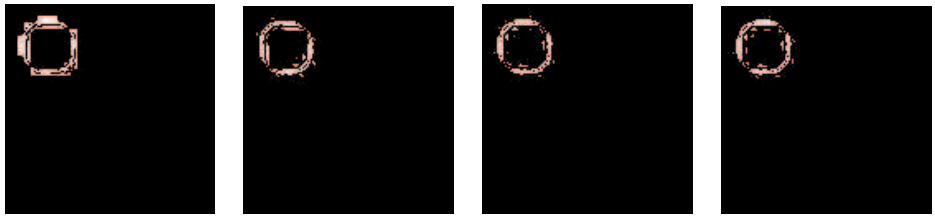
proper modification of approximation coefficients is one of the easiest way to detect contours. For the three-dimensional point of view, we deal with some simple methods of contour detection using wavelet transform such as replacing approximation coefficients by zeros and by the modified approximation coefficients of Canny detector.

6.1.2.1 Approximation coefficients replaced by zeros

This is the simplest method of contour detection with wavelet methods. The described 3D wavelet transform scheme in 5.2.1 can also be applied similarly. However, in the reconstruction strategy, the approximation coefficients are replaced by zeros. In fact, the wavelet decomposition separates the image into eight parts at the first level, each of them containing different informations: detail coefficients represent high-frequencies, which contain contours information while the approximation coefficients represent low-frequencies. This method removes all low-frequencies from the image as well and high-frequencies are kept to extract contours.



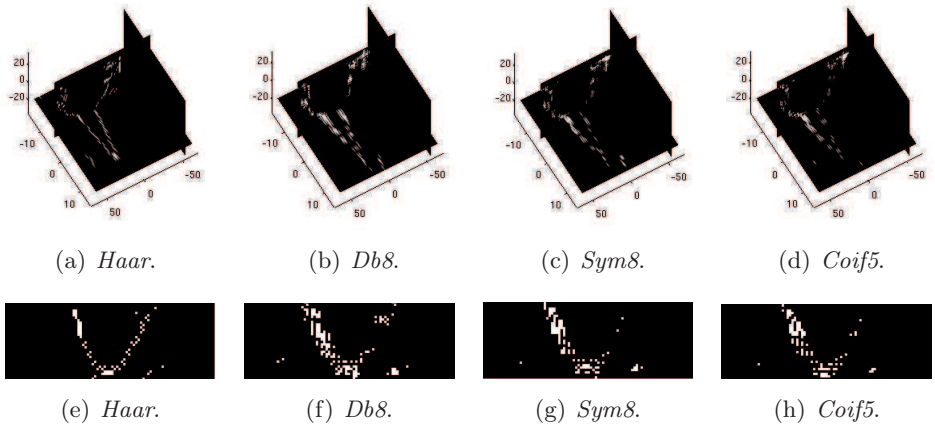
(a) *Haar.* (b) *Db8.* (c) *Sym8.* (d) *Coif5.*



(e) *Haar.* (f) *Db8.* (g) *Sym8.* (h) *Coif5.*

Figure 6.3: A test on the original moving disk video, contours found by replacing approximation coefficients by zeros and using some different wavelet basis (*haar, db8, sym8* and *coif5*). The first row: 3D viewing of data. The second row: one slice is taken.

The results of this method are shown in Figures 6.3 and 6.4. Experimental tests are performed with the original video of moving disk and the cerebellum MRI scan, respectively. We notice here that the taken input cerebellum for the proposed algorithm is well denoised applying isotropic variational model (P_{ROF2}) of section 3.1.2 ($\lambda = 25$. In these tests, different kinds of wavelet basis are taken into account: Haar, Daubechies (8 vanishing moments), Symlets (8 vanishing moments) and the Coiflets (5 vanishing moments). Then a simple threshold operation is used to identify



(a) *Haar.* (b) *Db8.* (c) *Sym8.* (d) *Coif5.*

(e) *Haar.* (f) *Db8.* (g) *Sym8.* (h) *Coif5.*

Figure 6.4: A test on the denoised volume of MRI scan of cerebellum (denoised results is performed by variational model (P_{ROF2})), contours found by replacing approximation coefficients by zeros and using different wavelet basis (*haar, db8, sym8* and *coif5*). The first row: 3D viewing of data. The second row: one slice is taken.

the contours.

In the test of moving disk, the contour shapes are well extracted. Haar basis provide undesired edge features extraction while the others give better continuous edges. In the medical image of cerebellum, there is a lot of textures in the image background that makes the task ambiguous. This is more difficult when seeing just one visual slice extraction Figure 6.4 (e)-(h). However, main contours are well detected by four testing types of wavelet bases. We see that replacing by zeros in wavelet contour detection gives bad results. In particular, all results provide contour shapes which are nor closed, nor continuous, nor retaining enough segmented details compared to results in Figures 6.1 and 6.2. These results are comparable to the previous contour shapes obtained with variational method (figures 6.1 and 6.2). Therefore the variational method gives better results than this wavelet based method.

6.1.2.2 Modification of Approximation coefficients by Canny detector

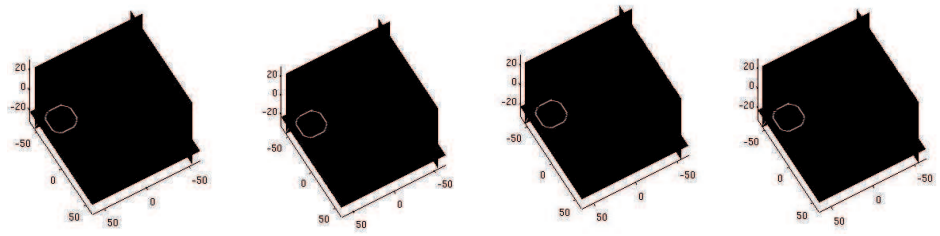
The Canny algorithm is a well-known edge detection algorithm. It is a multi step detector which performs smoothing and filtering, non-maxima suppression, followed by a connected-component analysis stage to detect contours, while suppressing non edge filter responses. The Canny edge detector is widely used in computer vision to locate sharp intensity changes and to find object boundaries in an image. In this section we give an outlook of the three dimensional Canny detector description (that is fully described in the Appendix E). A typical implementation of the Canny detector follows the steps below:

1. *Smoothing*: Smooth /filter the image to remove noise with an appropriate Gaussian filter;

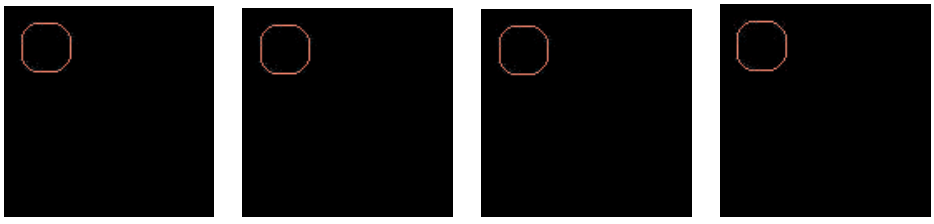
2. *Finding gradients*: Determine gradient magnitude and gradient direction at every pixel;

3. *Non-maximum suppression*: Only local maxima should be used to link edge points and deal out non-edge points, i.e., if the gradient magnitude at a pixel is larger than those at its two neighbors in the gradient direction, mark the pixel as an contour, otherwise, mark the pixel as the background.

In this subsection, we present another wavelet based method for contour/edge detection. This is a modification of approximation coefficients by the Canny detector. This kind of detector is applied to the approximation coefficients that are obtained at the first level of wavelet decomposition. Then, the wavelet reconstruction is performed from all remaining coefficients and modified approximation coefficients. Next step is a thresholding step applied to the reconstructed results. The results of the proposed method are shown in Figures 6.5 and 6.6. The numerical results show that the proposed algorithm, that combines wavelet transform and canny operator, accurately detects contours. We also remark that the input data of cerebellum for this method is denoised applying isotropic variational model (P_{ROF2}), is similar to the wavelet based approximation zeros replacement in previous section.



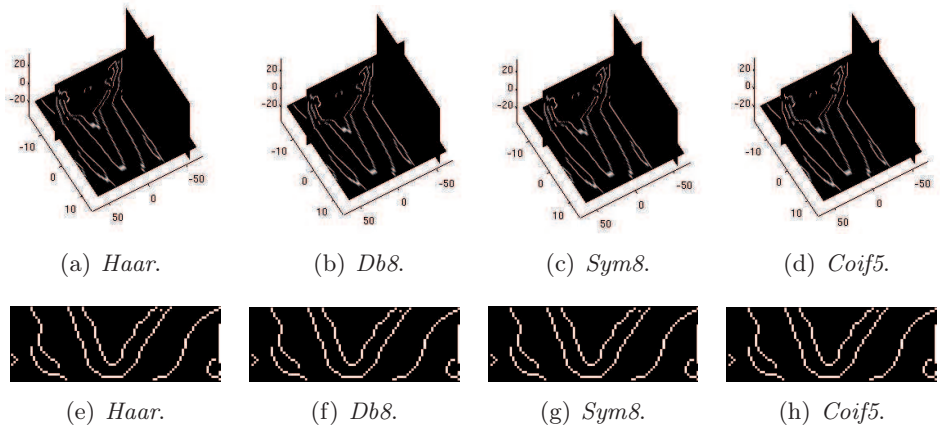
(a) *Haar.* (b) *Db8.* (c) *Sym8.* (d) *Coif5.*



(e) *Haar.* (f) *Db8.* (g) *Sym8.* (h) *Coif5.*

Figure 6.5: A test on the original moving disk video, contours found by Canny detector applied to the approximation coefficients and using different wavelet basis: *haar,daubechies-8,symlets-8* and *coiflets-5*. The first row: 3D viewing of data. The second row: one slice is taken.

These results prove that the proposed wavelet based contour detector gives comparable results to Canny detector in the case of noiseless images (contour location and thickness). Results are the same with different wavelet basis. Comparing to the contour maps produced by *zeros approximation replacement* (previous section



(a) *Haar.* (b) *Db8.* (c) *Sym8.* (d) *Coif5.*

(e) *Haar.* (f) *Db8.* (g) *Sym8.* (h) *Coif5.*

Figure 6.6: A test on the denoised volume of MRI scan of cerebellum (denoised results is performed by variational model (P_{ROF2})), contours found by Canny detector applied to the approximation coefficients and using some examples wavelet bases: *haar,daubechies-8,symlets-8* and *coiflets-5*. The first row: 3D viewing of data. The second row: one visual slice is taken.

6.1.2.1), the one produced by Canny algorithm provides more detail information. It is clear that the zeros replacing method cannot detect every edge of the image.

In video moving disk test, the original shape is obtained with a good balance between details and the contour localization accuracy is also good. The visual performance of the proposed method is clearly perceptible from Figures 6.5. The behavior of the proposed algorithm in presence of cerebellum is evaluated in Figures 6.6. The segmented results show a quite good detection of sharp contours. For both examples (moving disk video and the cerebellum) main objects and fine textures are perfectly detected. Compare to the results by variational methods, this method gives sharper contours better than the canny algorithm could do. Consequently, the proposed schemes play a very useful role in performance improvement.

6.2 Contour Closing Methods

Considering the three-dimensional angiography dataset for example, we need to recover the locations of blood-vessels. A simple global thresholding, based on gradient edge detection will give us poor results even when if we choose an optimal threshold value. Indeed, there exist many other structures with high grey level values. Global thresholding may keep noise if threshold parameter is too low. We may loose some information if threshold parameter is too large (as very thin vessels). That is the reason why we have to perform contour closing algorithms.

Here we describe two algorithms to be applied for 3D contour closing: the hysteresis thresholding and a local operator based on the so-called chamfer distance transformation. We give a quick overview of the original philosophy and motivation of each algorithm.

6.2.1 Hysteresis Thresholding

Hysteresis thresholding, is also called *double thresholding*, which is usually adopted in the Canny detector. Using the gradient gives a first approximation of image contours; however, some parts of contours may not be emphasized enough. On the other hand we may get contours which are not continuous. With hysteresis thresholding, the obtained object boundaries given are usually more complete and continuous than those given by pixel based techniques. This is challenging when considering images involving filaments structures as angiography supported vessels or skin medical image for human disease study. This is why hysteresis thresholding is proposed as a method filling gaps in image contours, [48].

Let f be a grayscale input image. We choose two intensity threshold parameters θ_1 (high threshold) and θ_2 (low threshold), where $\theta_1 > \theta_2$ can be set such that $\min_x f(x) \leq \theta_1, \theta_2 \leq \max_x f(x)$: θ_1 and θ_2 are upper and lower thresholds used in the hysteresis process. We construct two sets of pixels:

$$H = \{x \in \mathbb{R}^n | f(x) \geq \theta_1\} \tag{6.4}$$

$$L = \{x \in \mathbb{R}^n | f(x) \geq \theta_2\} \tag{6.5}$$

which can be used to produce binary masks such that all pixels above the threshold are "in" and all others are "out". We now wish to obtain $\mathcal{C}$ such that $H \subset \mathcal{C} \subset L$.

6.2.1.1 Principle

In our case, for closing contour objective, the hysteresis thresholding is performed after the step contour detection, then we will search the extreme points of the contour (if it is discontinuous) and try to extend them.

Consider Figure 6.7 below: two points A and B are two extremes of an edge after detection for contour curves, the closing contour method suggests to search in their respective neighborhoods $V(A)$ and $V(B)$ the corresponding remaining parts (or closest approximation) of image contours. This means that the pixels in the neighborhood close to A are not in the outline after the first threshold. So we apply a more permissive threshold on this neighborhood. Consequently, the contours will be gradually completed.

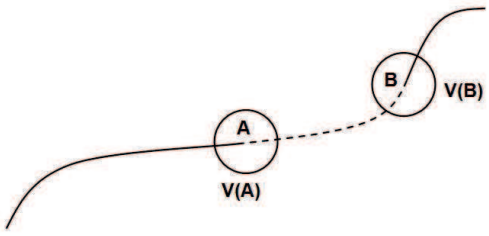


Figure 6.7: Hysteresis thresholding

6.2.1.2 Algorithm

The algorithm for hysteresis thresholding is performed by using two thresholds T_{high} and T_{low} ($T_{high} > T_{low}$). Consider an image f, we also denote ∇f the gradient operator of f. For contour-detection one usually uses a thresholding of gradient norm $\|\nabla f\|$.

1. **Step 1: First thresholding**: The image $I(m, n, p)$ is the binary image, which value is 1 if pixel (i, j, k) satisfies $\|\nabla f(i, j, k)\| \geq T_{high}$ and 0 other wise.

2. **Step 2: Local thresholding**: For every pixel C that belongs to the image contour, we consider a neighborhood. If there exists a neighbor pixel C' of C such that $I(C') = 0$ and $\|\nabla f(C')\| >= T_{low}$, then, we set $I(C') = 1$. Therefore C' is involved in the contour set.

3. **Step 3: Iteration**: Step 2 is iterated up to convergence.

4. **Step 4: Selection**: Outlines of few pixels are removed to keep only the relevant contours. Test the result and choose the suitable threshold parameter.

In the case we could choose $T_{high} = 2T_{low}$, then the image from T_{high} contains fewer contours but has gaps in the contours, meanwhile the image from T_{low} has many false contours. We combine result from T_{high} and T_{low} to link the contour shapes of T_{high} until we reach a gap and link contour from T_{high} with contour pixels from a T_{low} contour until a T_{high} contour is found again.

6.2.1.3 Examples

In figures 6.8 and 6.9, we give examples to describe the hysteresis thresholding method performed on 2D images. We consider the test of cerebellum in Figure 6.8: one slice is extracted from the denoised image after the variational method of (P_{ROF2}) for $\lambda = 25$ in particular (3D visual image is shown in Figure 5.6). Then we perform a thresholding to detect contour (Figure 6.8(b)). Two threshold parameters are selected and the hysteresis thresholding algorithm is applied: numerical result is shown in Figure 6.8(c). The difference between the two contours is presented in 6.8(d). The segmentation is done by hysteresis thresholding. However, the two thresholds have to be computed by an efficient fast rule. In the case of cerebellum, finding closed contours highly depends on the quality of the image.

(a) One slice of cerebellum test (denoised from variational model (P_{ROF2})).

(b) Detected contour by a simple thresholding step.

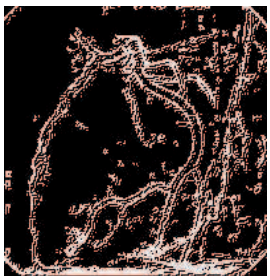
(c) Contour found by hysteresis thresholding.

(d) Difference between the input (b) and closed contours (c).

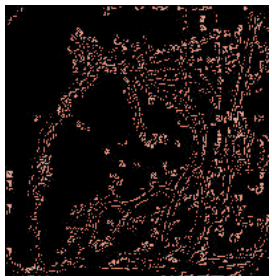
Figure 6.8: An example of 2D visual slice test on the denoised volume of MRI scan of cerebellum (denoised results is performed by variational model (P_{ROF2})), contours found by hysteresis thresholding closing method.

(a) 2D example of original blood vessel.

(b) Detected contour by a simple thresholding step.



(c) Closed contour found by hysteresis thresholding.



(d) Difference between the input (b) and closed contours (c).

Figure 6.9: An example of 2D blood vessel test, contours found by hysteresis thresholding closing method.

The hysteresis thresholding method has been also used to close contours of the "blood-vessel" test (Figure 6.9) on a 2D slice. We describe comparable the vessel contours shown in 6.9(b), (c). The method we have described performs well. However, it gives some false features of blood vessels. Some unexpected details make closed contours not accurate. In particular, two threshold values determination creates a lot of false pixels around vessels contour lines. In addition, the hysteresis thresholding method should be improved. Indeed, threshold upper and lower values should be found automatically. We report results obtained on a larger data set (volume in three dimensions) in the near future, chapter 8.

6.2.2 Chamfer Distances Transformation

We now study another local operator that has been proposed to close contours. The method is described in [35]. The "chamfer" distance measures distance between elements.

In what follows a *contrast point* is a point where the image intensity gradient is high. Contour points are constrast points if we precisely define what "high" means.

The basic idea of the method of [35] is the following:

(a) Generate a distance Table where the intensity (grey level) of a point is replaced by the distance to nearest contrast point.

(b) Find the saddle-points of the intensity function .

(c) Eliminate the points which associated distance is too large;

(d) Add those saddle points to contrast points and repeat steps $(a), (b), (c)$ and (d) as far as possible;

(e) Remove the contour points that do not close any the "meaningful" area.

Therefore the first step is the contrast points extraction. In this section, the concept of distance transformation and some of their properties, applications are described. We start with the two dimensional distance transform (DT), and the generation of distance transform to higher dimensions (three dimensions in particular) will be given. The distance transform was described in [21] and we just give hereafter some discrete definitions and implementations of DT.

6.2.2.1 Two-dimensional Distance Transformation

Definition 10. [35], [21] **Discrete Distance**
Let $\mathbb{E} = \mathbb{Z}^N$, $N \geq 2$. *An image* I *is an application defined on* $\mathbb{E}$.
We call $d : \mathbb{E} \times \mathbb{E} \to \mathbb{N}$ *the discrete distance if it verifies:* $\forall A, B, C \in \mathbb{E}$

1. $d(A, B) \geq 0$; $d(A, B) = 0 \Leftrightarrow A = B$: *positive;*
2. $d(A, B) = d(B, A)$: *symmetric;*

(a) Telephone - Original image (b) Telephone - Binary image (c) Telephone - Distance image (Euclidean)

Figure 6.10: A 2D example for distance transformation.

3. $d(A,B) \leq d(A,C) + d(C,B)$: *triangular inequality.*

Definition 11. [35], [21] **Distance Transformation**
Let d be a distance on $\mathbb{R}^2$, I be 2D binary image and $O \subset I$. For each pixel in the background $\bar{O} := I \backslash O$, the distance transformation image denoted DT is defined as following:

$$DT : p \mapsto DT(p) = \min \left\{ d(p,q), q \in \bar{O} \right\} .$$

In other terms, the value assigned at any point p is the shortest distance from p to all points in O.

Depending upon various applications, the distance function may be defined in different ways, some most important kinds of function that are included in [21]. For example, the n-neighbor distances, they are 4- and 8-neighbor distance (for two dimensional case) and $6-$, $18-$ and $26-$neighbor distance (for three dimensional case), the Euclidean distance, octagonal distance, and the chamfer distance ... In this thesis, we are interested in choosing of chamfer distances map for a binary image respectively.

In two dimensions, each pixel has two kinds of neighbors. The first kind is 4 horizontal/vertical neighbors, the neighbors joined by a line. The second kind is the 4 diagonal neighbors, the neighbors joined only by a point. The global distance is computed from the local distances between neighbors. The distance between horizontal/vertical neighbors is denoted d_1, and the distance between diagonal neighbors is denoted d_2.

For example, in [21], with 4-neighbors distance we have $d_1 = 1$ and $d_2 = \infty$ (this means that all sums including d_2 can be ignored), the discs of this distance are diamonds; for 8-neighbors distance $d_1 = 1$ and $d_2 = 1$, the discs of this distance are squares.

The (2D) chamfer distance transform has real values based on d_1 (horizontal) and d_2 (vertical) as well. The name of this distance transform comes from the two-pass process for obtaining the distances, which is known as *chamfering*. However, in most applications real numbers are not desirable. The good integer approximations of the optimal values are found, $d_1 = 2$, $d_2 = 3$, the incremental distance values of

2 and 3 provide relative distances that approximate the Euclidean distances 1 and the square root of 2. This is why it is an euclidian pseudo-distance.

Position	$j-1$	j	$j+1$
$i-1$	$+d_2$	$+d_1$	$+d_2$
i	$+d_1$	0	

	0	$+d_1$	i
$+d_2$	$+d_1$	$+d_2$	$i+1$
$j-1$	j	$j+1$	Position

Table 6.2: The forward pass (left) and backward pass (right) from $I(i,j)$, by chamfer algorithm.

The algorithm for distance transformation can be described by masks in Table 6.2. Inside these masks, two passes over the picture are necessary:

- During the forward pass the mask starts at the upper left corner of the picture, moves row-by-row to perform computation on image I from left to right, and from top to bottom to find the partial distance value $d(i,j)$ for every pixel (i,j). The local distances, d_1 and d_2, in the maskpixels are added to the pixel values in I, and new value of the zero pixel is the minimum of five sums. During the forward pass, the optimal distance for the pixel (i,j) of the mask is updated by the formula:

$$d\,(i,j) = \min \left\{ \begin{array}{l} d\,(i,j)\,; \\ d\,(i,j-1) + d_1; \\ d\,(i-1,j-1) + d_2; \\ d\,(i-1,j) + d_1; \\ d\,(i-1,j+1) + d_2 \end{array} \right\}. \tag{6.6}$$

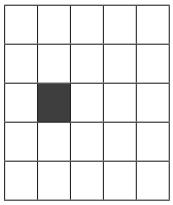
- Similarly, in the backward pass the mask starts in the lower right corner, executes from right to left, and from bottom to top to find the second partial distance value $d(i,j)$ for each pixel (i,j). The distance for pixel (i,j) of the mask can be calculated in terms of:

$$d\,(i,j) = \min \left\{ \begin{array}{l} d\,(i,j)\,; \\ d\,(i,j+1) + d_1; \\ d\,(i+1,j-1) + d_2; \\ d\,(i+1,j) + d_1; \\ d\,(i+1,j+1) + d_2 \end{array} \right\}. \tag{6.7}$$

Table 6.3 gives an example, shows the results of different settings of 2D distance transformation for a certain 5×5 image where the black pixel is the object.

6.2.2.2 Three-dimensional Distance Transformation

Three-dimensional elements that are corresponding to two-dimensional case, are usually called voxels, and there are 6−, 18− or 26−neighbor distances may be considered. For instance, with 26−neighbor distance, voxel has 26 neighbors of three different kinds. The first, closest, kind of neighbors is the 6 ones joined to the



3	2	3	4	5
2	1	2	3	4
1	0	1	2	3
2	1	2	3	4
3	2	3	4	5

2	2	2	2	3
1	1	1	2	3
1	0	1	2	3
1	1	1	2	3
2	2	2	2	3

5	4	5	6	8
3	2	3	5	7
2	0	2	4	6
3	2	3	5	7
5	4	5	6	8

Table 6.3: Results of different settings distance transformations. From left to right: the first Table is the 5×5 image with black pixel is an object; the second one is the result after 4-neighbor distance; next after 8-neighbor distance and the last one is result after chamfer distance transformation.

voxel by a plane, the second kind is the 12 neighbors joined by a line, and the third kind is the 8 neighbors joined by only a point. The three different local distances are denoted d_1, d_2 and d_3. As in two dimensions, the algorithm can be illustrated

$+d_3$	$+d_2$	$+d_3$
$+d_2$	$+d_1$	$+d_2$
$+d_3$	$+d_2$	$+d_3$

$+d_2$	$+d_3$	$+d_3$
$+d_1$	0	

	0	$+d_1$
$+d_2$	$+d_1$	$+d_2$

$+d_3$	$+d_2$	$+d_3$
$+d_2$	$+d_1$	$+d_2$
$+d_3$	$+d_2$	$+d_3$

Table 6.4: The forward pass (first row) and backward pass (second row) in 3D case, by chamfer algorithm.

by the same masks as in Table 6.4 by using d_1, d_2 and d_3, and two passes over the volume are needed. The forward mask is moved over the volume left to right, top to bottom, and front to back. The backward mask is moved in the opposite way. In each position, the sum of the local distance in each maskvoxel and the value of the voxel it covers is computed, and the new value of the zero voxel is the minimum of these sums.

As remarked before in two-dimensions, the values of d_1, d_2 and d_3 are set dependently on which kind of distance transformation. For example, in [21] we have:

- $6-$neighbor distance: $d_1 = 1$, $d_2 = d_3 = \infty$.
- $18-$neighbor distance: $d_1 = 1$, $d_2 = 1$ and $d_3 = \infty$.
- $26-$neighbor distance: $d_1 = d_2 = d_3 = 1$.
- Chamfer distance: as 2D case, it is often desirable to use only integers. A very good integer approximation of the optimal local chamfer distances is $d_1 = 3, d_2 = 4$, and $d_3 = 5$. We can refer to [35] for a detailed explanation, and this chamfer $(3, 4, 5)$ distance is used in this chapter.

6.2.3 Contour Closing - A local operator based on Chamfer Distance Transformation

6.2.3.1 Saddle points determination

Definition 12. [9] *Let $f : V \times W \to \mathbb{R} \cup \{\infty\}$ be a function defined on $V \times W$, where V and W are vector spaces. A point $(a, b) \in V \times W$ is called a saddle point of f if:*

$$\begin{cases} a \in \arg\min\limits_{x \in V} f(x, y) \\ b \in \arg\max\limits_{y \in W} f(x, y) \end{cases}$$

Proposition 5. [9] *The point (a, b) is a saddle point of f if and only if:*

$$\min_x \max_y f(x, y) = \max_y \min_x f(x, y) = f(a, b) \tag{6.8}$$

Proof. Recall the definition, $\forall (x_0, y_0)$ we have:

$$\max_y f(x_0, y) \geq f(x_0, y_0) \geq \min_x f(x, y_0) \tag{6.9}$$

We can conclude the proposition since the above inequality still holds when we move to min and max on the left and right, so the saddle point definition is equivalent to:

$$\max_y f(x_0, y) \geq f(x_0, y_0) \geq \min_x f(x, y_0) \tag{6.10}$$

□

We give thereafter equivalent formulations:

Definition 13. [35] *Let $f(x, y)$ be a real function of two variables defined over a real open subspace $\Omega \subset \mathbb{R}^2$ and differentiable with respect to these two variables on Ω. We say that the point $(a, b) \in \Omega$ is the saddle point of f if:*

(a) $\dfrac{\partial f}{\partial x}(a, b) = \dfrac{\partial f}{\partial y}(a, b) = 0,$

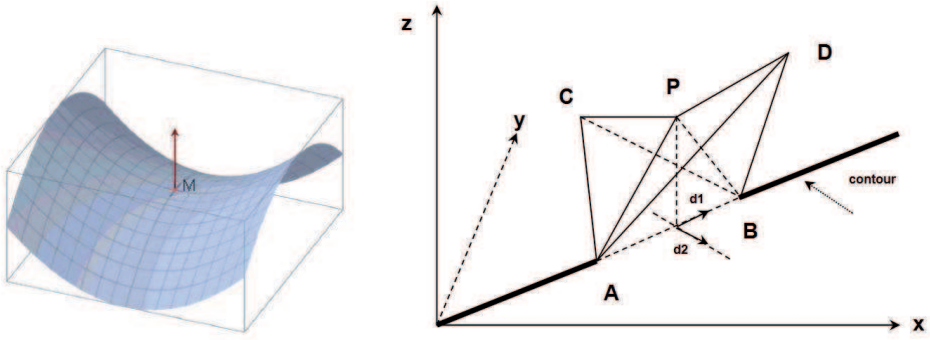
(b) *(a, b) is neither a local maximum nor a local minimum of f.*

Definition 14. [35] *Consider now an open bounded subset K of Ω (K will play the role of contour points in our saddle point method). Let a function $f_K : \Omega \to \mathbb{R}$ be defined as:*

$$f_K(x, y) = \min \{ d((x, y), (u, v)) \mid (u, v) \in K \} = d((x, y), K)$$

We say that $P \in \Omega$ is a saddle point if there exist two directions d_1 and d_2 in $\mathbb{R}^2$ such that $f(P)$ is the maximum in direction d_1 and the minimum in d_2 :

$$\exists d_1, d_2 \in \mathbb{R}^2, \exists r > 0, \forall \lambda : |\lambda| \leq r \Rightarrow \begin{cases} f(P + \lambda d_1) < f(P) \\ f(P + \lambda d_2) > f(P) \end{cases}$$



(a) Example for the 1^{st} and 2^{nd} definition of Saddle point, with $z = x^2 - y^2$, $M(0,0)$ is the saddle point.

(b) The 3^{rd} definition of Saddle point.

Figure 6.11: The saddle point example presentations.

Proposition 6. [9], [35]
Let K be a finite union of segments in space and A, B be two end (distinct) points of two segments in K. Let P be the medium of (A, B) and $r = d(A, P) = d(B, P)$. We denote $D(M, t)$ the closed disk with center M and radius t.
Suppose that we have:

(a) $D(P, t) \cap K = \{A, B\}$.

(b) $\exists \varepsilon > 0, D(P, r + \varepsilon) \cap K = \{(A, A'), (B, B')\}$.

Then, P is the saddle point of f_K, that means:

- *$f_K(P)$ is the local minimum respect to direction d_1.*
- *$f_K(P)$ is the local maximum respect to direction d_2.*

This proposition is well proved in [35].

6.2.3.2 Discretized Saddle Points

We now consider the 2D discrete case corresponding to a scanned image (the whole Ω) and a set of contour points (the points of contrast). In this case, we look for the discretized saddle points y using 8-neighbor pixels of pixel P.

The point P is called a (discretized) saddle point if there exists four points I, J, Q, R such that:

(1) I, J and P are aligned.

(2) Q, R and P are aligned.

(3) The segments (I, J) and (Q, R) intersect each other.

(4) $f_K(I) \leq f_K(P)$ and $f_K(J) \leq f_K(P)$.

(5) $f_K(Q) \geq f_K(P)$ and $f_K(R) \geq f_K(P)$.

(6) At least three of the inequalities represented by (4) and (5) are strict.

Let us give an example in Table 6.5:

							I			
$I \rightarrow$	1	4	2			3	3	3		
$Q \rightarrow$	4	3	5	$\rightarrow R$	$R \rightarrow$	4	4	5	$\rightarrow Q$	
	7	4	2	$\rightarrow J$		3	3	3		
							J			

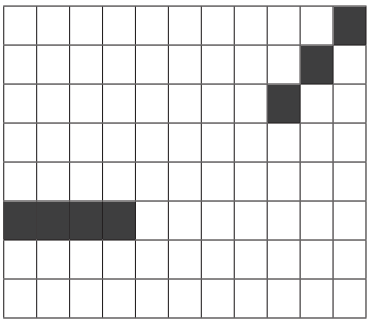
Table 6.5: Some example of a 2D discretized saddle point.

6.2.3.3 Algorithm

Let K_0 the set of all contrast points that need to be updated, and $f_0(M) = f_{K_0}(M) = d(M, K_0)$ (we want to consider if M is the saddle point or not).

1. Step 1: $K = K_0$.
2. Step 2: Compute $\mathcal{P}(f_K)$, is the set of all saddle points of image f_K.
3. Step 3: If $\mathcal{P}(f_K) = \emptyset$ then go to step 4. If not, update $K = K \cup \mathcal{P}(f_K)$ and repeat step 2.
4. Step 4: Stop.

The process of found and replaced saddle points by the contour positions is iterated until a contour is reached or until a fixed number of iterations is attained. Let consider an example in Table as following, which represents two segments. We obtain how this algorithm performs for closing them.



10	10	10	10	10	8	6	5	3	2	0
8	8	8	8	9	7	5	3	2	0	2
6	6	6	6	7	6	4	2	0	2	3
4	4	4	4	5	6	5	3	2	3	5
2	2	2	2	3	5	6	5	4	5	6
0	0	0	0	2	4	6	7	6	7	8
4	2	2	2	3	5	7	9	8	9	10
∞	4	4	4	5	6	8	10	10	11	12

Table 6.6: Left: Object that need to be closed with two black segments. Right: The chamfer distance transformation of the object.

10	10	10	10	10	8	6	5	3	2	0
8	8	8	8	9	7	5	3	2	0	2
6	6	6	6	7	6	4	2	0	2	3
4	4	4	4	5	6	0	3	2	3	5
2	2	2	2	3	0	6	5	4	5	6
0	0	0	0	2	4	6	7	6	7	8
4	2	2	2	3	5	7	9	8	9	10
∞	4	4	4	5	6	8	10	10	11	12

Table 6.7: Closing result of the object in Table 6.6 after one iteration, two saddle points are found and replaced.

10	10	10	10	10	8	6	5	3	2	0
8	8	8	8	9	7	5	3	2	0	2
6	6	6	6	5	3	2	0	0	2	3
4	4	4	4	3	2	0	0	2	3	5
2	2	2	2	0	0	2	3	4	5	6
0	0	0	0	0	2	3	5	6	7	8
4	2	2	2	3	4	5	6	8	9	10
∞	4	4	4	5	6	8	10	10	11	12

Table 6.8: Closing result of the object in Table 6.6 after two iterations, four next saddle points are found and replaced. At this time we have $\mathcal{P}(f_K) = \emptyset$, we stop.

6.2.3.4 Examples

In this section, we give examples (cerebellum and blood vessel) as previously done in hysteresis section. The 2D closed contours are shown in figures 6.12 and 6.13. Numerical results show that the chamfer distance based method is slightly better than the hysteresis thresholding one. The efficiency has been also evaluated in Figures 6.12(c) and 6.13(c). In particular, a lot of the contour pixels which were not detected in the hysteresis thresholding method are found now. It requires little computation. The presence of a saddle point is taken during a second scan of the image. The obtained results are good and allow the closure of a majority of gaps of 5 and 10 pixels after a few iterations. This method can be used to perform preprocessing during image segmentation operations. It is more efficient than the hysteresis thresholding method.

The chamfer distance operator could be applied for closing contours after contour extraction step. Moreover, this step-contour-detector is well-suited to extract contour shapes, especially our three-dimensional medical images consideration. In the next chapter 8, an experimental application will be performed for the 3D vessel of mouse brain data. We shall briefly discuss the method and algorithm we used.

(a) One slice of cerebellum test (denoised from variational model (P_{ROF2})).

(b) Detected contour by a simple thresholding step.

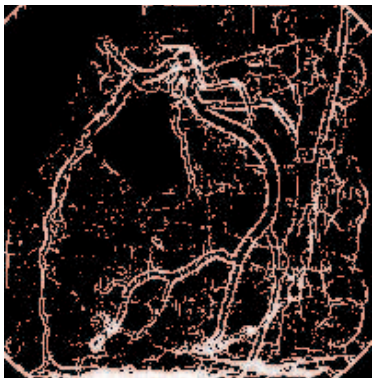
(c) Contour found by Chamfer-distance.

(d) Difference between the input (b) and closed contours (c).

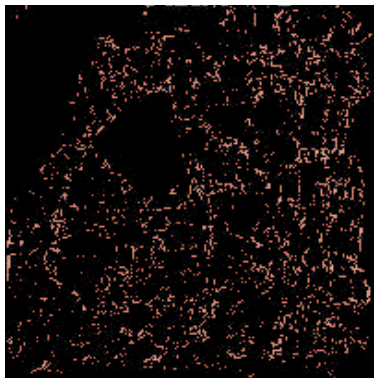
Figure 6.12: An example of 2D visual slice test on the denoised volume of MRI scan of cerebellum (denoised results is performed by variational model (P_{ROF2})), contours found by chamfer-distances closing method.

(a) 2D example of original blood vessel.

(b) Detected contour by a simple thresholding step.



(c) Contour found by Chamfers distance.



(d) Difference between the input (b) and closed contours (c).

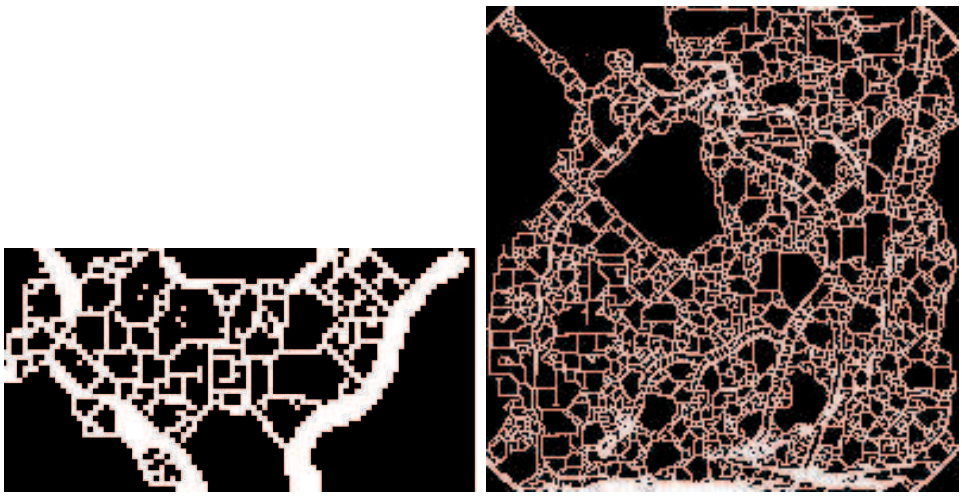
Figure 6.13: An example of 2D blood vessel test, contours found by chamfers distance closing method.

Remark 2. *Because of the discretization process, there are some cases with false detection. This problem can be resolved by making two types of filtering at each iterations.*

- *When these false detections occur for distant points of the contours, it can be carried out by setting a minimum distance of intervention of the operator;*
- *Assuming that the contours separate regions homogeneous grayscale, false saddle-points do not correspond to contrast points. Gradient variations at these points are low, this can be used to not validate them.*

Figure 6.14 shows false detected pixels of contour shapes during the saddle points finding process. Therefore, it is proposed to catch neighboring pixels that mark contours. Finally, we perform a selection process of contour positions accurately to find all true contours.

In Figure 6.15 below, we display again the closed contour results that have been performed by two based methods: the closing by hysteresis thresholding and by the



(a) One 2D visual slice of cerebellum. (b) 2D blood vessel testing.

Figure 6.14: False detections occur when finding closed contours by Chamfer-distances, described in remark 2.

chamfer distance transform. We may see that the method based on chamfer distance is worth comparing to the hysteresis, while this method provides contour details are more efficiently than another.

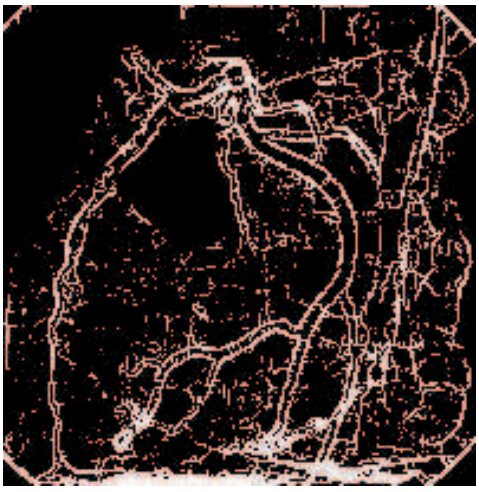
The method has been successfully tested on 2D examples of MRI scan (one visual denoised slice of cerebellum and a vessel model with thin structures). As we know, the Canny detector is a successful method in contour detection which is available in MATLAB toolbox. But in this section, we do not perform any comparison between Canny detector and proposed methods that only give a basic understanding of each closing methods (by hysteresis thresholding and the chamfer distance transform) and compare between them together. Some experimental tests give us an evaluation to compare two closing methods. Final results of chamfer distance based method present more accuracy than the hysteresis thresholding method. As a comparison, hysteresis follows the ridges of the gradient norms, it creates more false edges and small regions in fibers of contours meanwhile this local proposed method that prolongs/suppresses contours with the candidate along the path.

We also need to prove that such algorithm is able to segment any type of image, even if it includes textured or non-textured areas. Further testing on more demanding tasks such as vessel of mouse brain segmentation are being conducted, especially in the case of three dimensional space.

(a) Closed contour of denoised cerebellum found by hysteresis thresholding.

(b) Closed contour of denoised cerebellum found by Chamfer-distance.

(c) Closed contour of blood-vessel found by hysteresis thresholding.



(d) Closed contour of blood-vessel found by Chamfers distance.

Figure 6.15: A comparison between two based methods: hysteresis thresholding and the chamfer distance contour closing. On the first row: tested on one visual slice of cerebellum data. The second row: tested on the 2D blood vessels image.

Chapter 7

Image Decomposition model based on variational method and wavelet transform

Contents

Throughout previous chapters, we have studied two based methods and their applications in image implementation, the second-order variational method and wavelet transform simulated on three dimensional medical images. Because each method has the advantages and disadvantages, BV^2-variational method gives good denoised results meanwhile the wavelet shrinkage technique reduces well edge/contour artifacts. In this situation we propose a new modified model that combines these schemes, we then have a reconstructed image which has fewer oscillations near edges and noise is smoothed. A lot of references, for instance in [24], [25] etc, gave ideas of denoising algorithm based on a combination of these frameworks several years ago. It permits us to improve another approach that employs variational framework, in particular the minimization model (P_{ROF2}) plus the wavelet thresholding to reduce oscillations, remove noise while maintain the sharpness of image.

The ideas introduced here can be considered to perform in this chapter and use as a post-processing technique for image denoising application. Taking advantages of both two methods (variational method and wavelet based scheme) mentioned previously, we apply them in a new model that combines them; the numerical results are well denoised and almost geometrical details are well preserved. Therefore, here we propose a new small variational model for image denoising and decomposition (the so-called "merged-problem"), which combines the second-order total variational model of Rudin-Osher-Fatemi (P_{ROF2}) and the Wavelet transform.

In [66], Y. Meyer has introduced an image decomposition model to split an image into two components: a geometrical components and texture (oscillatory) component. Inspired by his work, numerical models has been developed to carry out the decomposition of gray scale images. Other recent and related image decomposition models combined from variational model and wavelet transform are well proposed.

In our approach, the considered model will decompose an image into several components, where each of them is characterized by a special space of function. More precisely, the aim of work in this chapter is to construct a model which decompose image into two components: the first one represents all geometrical information of the image, the second one contains the oscillating structures of image. In the rest of this chapter, we present some numerical results applied both (P_{ROF2}) and an undecimated wavelet transform (the "à trous" algorithm) in order to deal with the case of 3D noisy images.

7.1 Introduction of the model

The goal of the proposed method is to split a 3D image f into two components, $f = u + v$. The first component is TV^2 minimization term so that u is the second order bounded variation component of the original image f. The BV^2-model is well adapted because of the good geometrical information identification and a better contours preservation than the BV model. The second term gives the v component containing the oscillating part of the image, namely textures and noise.

In [2], the authors propose a model to decompose an image into two parts; one of them is the structure of image, characterized by the BV space, and another part of image is defined as the oscillating patterns. Their assumption is to consider noise as a distribution modelized by the Besov space $\dot{B}^{\infty}_{-1,\infty}$, with the standard setting $s = -1, p = q = \infty$ for the Besov space $\dot{B}^{p}_{s,q}$; we can refer to [66], [2] some references of the Besov space.

We denote the E here is a dual space to model oscillating patterns. $\dot{B}^{1}_{1,1}$ is the usual homogeneous Besov space and the dual space of $\dot{B}^{1}_{1,1}$ is the Banach space $E = \dot{B}^{\infty}_{-1,\infty}$. The Besov space $\dot{B}^{\infty}_{-1,\infty}$ is adapted to modelize the noise.

This decomposition has been proposed by Aujol et al. in [2] and is computed by minimizing a convex functional which depends on two variables (u, v) as following:

$$\inf_{(u,v)\in X^2} J_2(u) + B^*(v/\delta) + \frac{1}{2\lambda}\|f - u - v\|_X^2 \qquad (\mathcal{P})$$

where X is the discretized space and J_2 represents the discretization of the second total variation term TV^2, see chapter 3 mentioned in the section 3.1.1. In this case, the solution $v \in E_\delta$ is defined by:

$$E_\delta = \left\{\omega \in \dot{B}^{\infty}_{-1,\infty} / \|\omega\|_{\dot{B}^{\infty}_{-1,\infty}} \leq \delta\right\}. \qquad (7.1)$$

The term $B^*(\cdot)$ is the indicator function on E_δ defined in the following expression:

$$B^*\left(\frac{\omega}{\delta}\right) = \chi_{\{\|\omega\|_E \leq \delta\}}(\omega) = \begin{cases} 0 & \text{if } \|\omega\|_{\mathrm{E}} \leq \delta, \\ +\infty & \text{otherwise.} \end{cases} \qquad (7.2)$$

As the principle proposed by Aujol F. [2], in order to solve the problem $(\mathcal{P})$, one considers to solve the two following problems and discretize their solutions:

1. v being fixed, look for u as a solution to problem:

$$\inf_{u\in X} J_2(u) + \frac{1}{2\lambda}\|f-u-v\|_X^2, \tag{7.3}$$

2. u being then fixed, look for v as a solution to:

$$\inf_{v\in\delta B_E} \|f-u-v\|_X^2. \tag{7.4}$$

The solution to problem (7.3) is given by:

$$u^* = f - v - P_{\lambda K_2}(f-v). \tag{7.5}$$

(we applied the theorem 16), where $P_{\lambda K_2}$ is the orthogonal projector operator on λK_2. We could apply the Chambolle or Nesterov-type algorithm described in chapter 3 to obtain the approximated solution to this problem (7.3).

Solution of (7.4) is given by:

$$\hat{v} = P_{\delta B_E}(f-u) \tag{7.6}$$

As in [2], in order to compute this term of projection, we need to consider and solve the dual problem of (7.4) as:

$$\inf_{\omega\in X} \frac{1}{2}\|f-u-v\|_X^2 + \delta\|\omega\|_{\dot{B}^1_{1,1}}. \tag{7.7}$$

The approximated solution to this problem (7.7) is obtained by using the universal threshold δ during the iteration process [2] on an undecimated wavelet transform, the 'à trous' algorithm (details are illustrated in the section 4.2 chapter 4). The solution is $\hat{\omega} = UWT(f-u,\delta)$. Consequently, the solution to problem (7.4) can be written $v^* = P_{\delta B_E}(f-u) = f-u-UWT(f-u,\delta)$, where UWT denotes the undecimated wavelet thresholding.

Throughout the section of introduction, we do not give a very detailed description of decomposition model, because it has recently studied through a lot of references, for instance, Meyer Y. in [66] , and Aujol F. [2] etc ... We herein just follow their works in which the second-order bounded variation BV^2 takes place instead of the first order variational space BV inside the considered two components decomposition model. We could refer to these works in [58], [2], and [66].

7.2 Algorithm

Image decomposition model is solved by the following iterative algorithm:

Algorithm 6 Image decomposition $u + v$ model

1. Initialization: $u_0 = v_0 = 0$,
2. Iterations on n:
$$u_{n+1} = f - v_n - P_{\lambda K_2}(f - v_n) \tag{7.8}$$
$$v_{n+1} = f - u_{n+1} - UWT(f - u_{n+1}, \delta) \tag{7.9}$$
the threshold parameter δ is updated associated to $f - u_{n+1}$ as n increases.
3. Stopping test: if the following condition is fullfilled:
$$\max\left(|u_{n+1} - u_n|, |v_{n+1} - v_n|\right) \leq \epsilon \tag{7.10}$$

7.3 Numerical results

In this section, we present numerical results obtained with the proposed new model. Numerical results for image denoising, image decomposition and texture discrimination, show that the new model performs decomposition of a possible noisy given image, into a cartoon and oscillating term.

The 3D image decomposition method is applied to 3D medical data (mentioned throughout the thesis): the MRI of cerebellum. The model is tested for different values of regularizing parameter λ. Since in practice, there is no denoised volume to compare to, tuning of parameter λ often relies on visual inspection. The stopping criterion has been set to a maximal number of iterations which can be chosen arbitrary large.

It can be seen that the new algorithm is good for denoising and texturing image purposes. Based on two separable methods: variational method and the wavelet shrinkage, the model is well adapted to combine the advantages of each method that we mentioned in chapter 5. The second-order total variational method gives good denoised results and the wavelet shrinkage technique reduces well edge/contour artifacts. Therefore, the framework based on both of them is a good approach and it clearly outperforms numerical results better than each classical models.

One can observe that the algorithm is able to separate the initial MRI image into several parts; for instance the component u that contains the regularized (denoised) image, the remaining noises (oscillating patterns) are included in the v component. We then compute the last term $\omega = f - u - v$ and see that this contains mostly textures and contours information of our image, Figure 7.1.

The good ability to denoise the initial 3D image is confirmed on Figure 7.2, which shows one slice on the 3D image represented as a 2D surface, its regularized component u ($\lambda = 10$). Experimental results show that the proposed model give better denoised solution than the model (P_{ROF2}). From the Figure 7.2, this model makes fewer visible edges/contours in the texture component v and more edges/contours information in the cartoon component u compared with the ROF2 model.

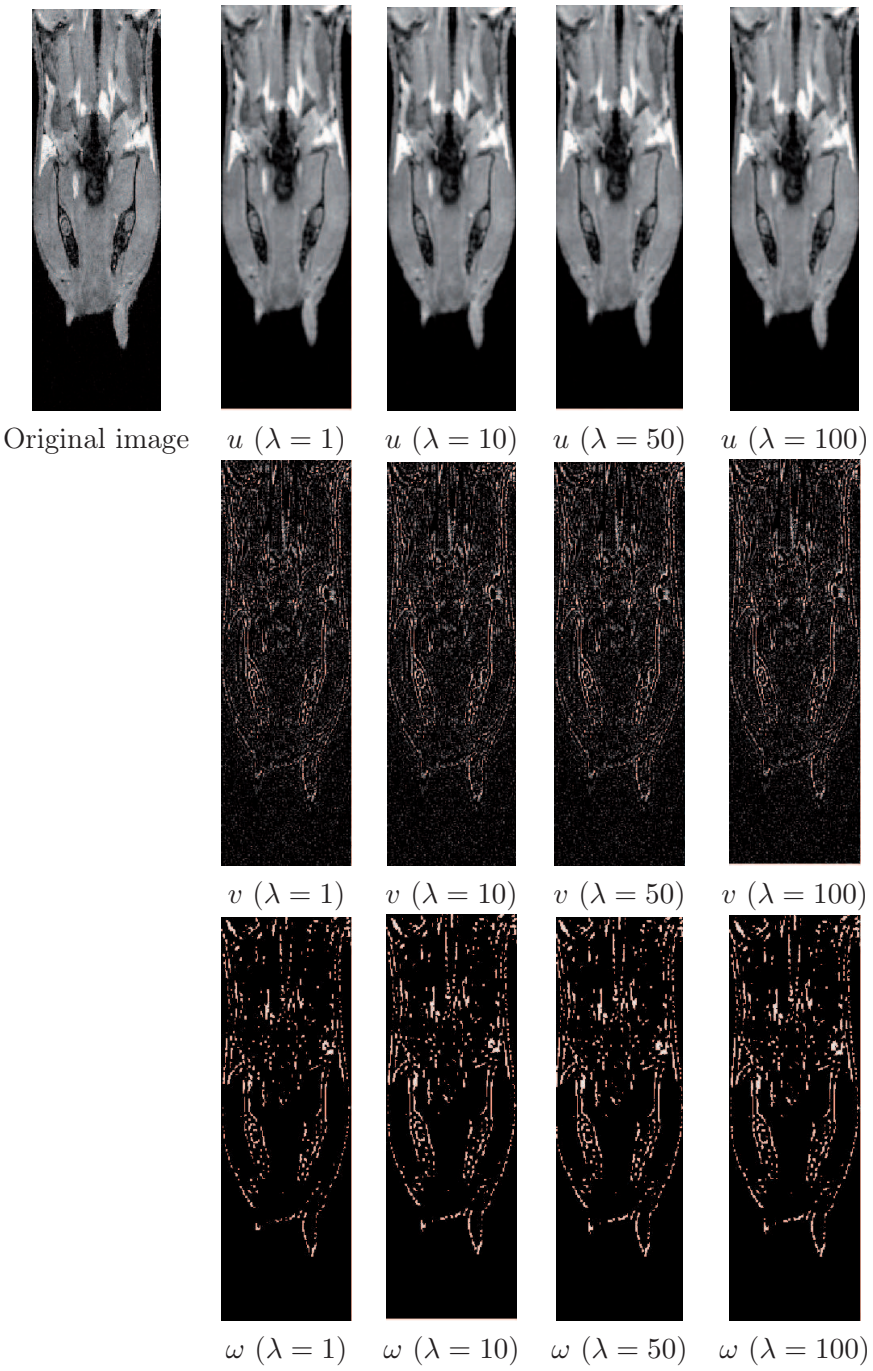
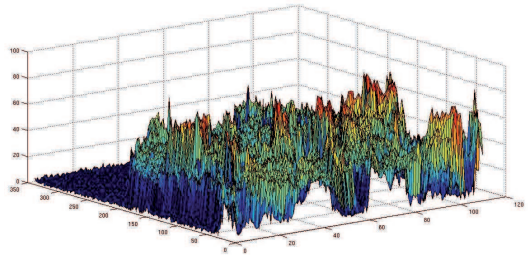
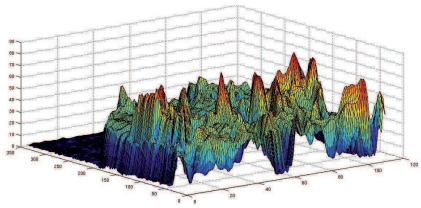
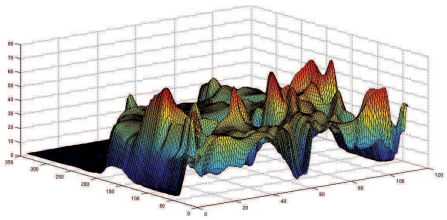


Figure 7.1: Numerical tests on the cerebellum data, one visual slice viewing. Decomposition model considers original image f to the u that contains geometrical information (the first row), v represents the oscillating component (second row). We then compute the remaining term $\omega = f - u - v$ and plot them on the third row. Comparison of the $u + v$ decomposition for different values $\lambda = 1, 10, 50, 100$.

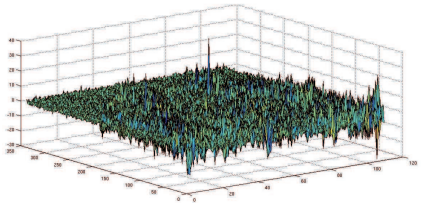




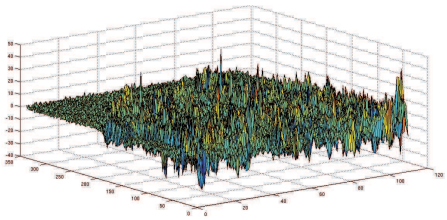
(a) u component observed by the $(\mathcal{P})$ model.



(b) u component observed by the (P_{ROF2}) model.



(c) v component observed by the $(\mathcal{P})$ model.



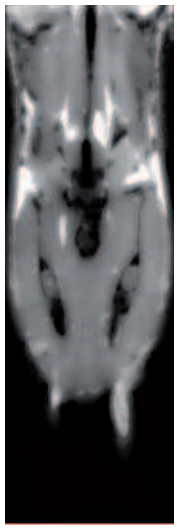
(d) v component observed by the (P_{ROF2}) model.

Figure 7.2: Surface representation of one slice of the original 3D cerebellum volume (top), the u component (the second row) and v component (the third row). Tests illustrated for $\lambda = 10$.

One can conclude that the proposed model gives the best visual effects. A comparison using the same decomposition model without undecimated wavelet shrinkage has also been performed (using the same value for $\lambda = 10$). It can be noticed on Figure 7.2 (right column), that the u component is a bit oversmoothed and thus region borders are blurred.

In addition, one sees on the Figure 7.3 the comparison between restored results observed by our model $(\mathcal{P})$ and two previous strategies we have studied: the applied (P_{ROF2}) only, the wavelet denoising scheme applied "à trous" algorithm. The restored image of cerebellum by our approach has been more regularized than two remaining classical methods. Moreover, almost geometrical details of image are well preserved while noise is removed, too. This new approach combines the advantages of our two methods: second-order variational minimization model and the wavelet thresholding scheme. Once the model (P_{ROF2}) is very promising in denoising, the applied wavelet thresholding provides geometrical details are well preserved.

(a) u by (P_{ROF2}) model only.



(b) u by wavelet 'à trous' only.

(c) u by ($\mathcal{P}$) model.

(d) v by (P_{ROF2}) model only.

(e) v by wavelet 'à trous' only.

(f) v by ($\mathcal{P}$) model.

Figure 7.3: Cerebellum data. One visual slice comparison between different components, cartoon component u and the oscillating component v observed by the model (P_{ROF2}) only, for $\lambda = 10$ (the first column), the wavelet 'à trous' algorithm only (scale=3) (the second column) and compared to the image decomposition model ($\mathcal{P}$), for $\lambda = 10$.

Consequently, in this chapter, we have tried to introduce the general framework which combines both previous approaches: second-order variational minimization

and wavelet thresholding scheme. The studied applied method can perform an image in which noise can be removed successfully and contours, textures are better preserved. It can be seen that this model is particularly well adapted to denoise textured images or even images with complicated structures inside (vessel of brain for example). Throughout the next chapter 8 we shall give a performance of this model applied to the vessel of mouse brain image, in which they give very interesting and promising results.

In conclusion, we have illustrated that the proposed method gives better results than those obtained by two previous models. And our approach seems very promising in many practical situations, where we want to split image into two or more components, each of them being characterized associated to a transform or reconstruction. This will be investigated in near future, follows by testing more types of images, more iterations enough (that depends on the CPU speed of process) etc..., and this proposed method can be also seen as a specific case of a more general approach in future work.

Chapter 8

Application to 3D biological image: mice vessels network

Contents

In chapter 1, we have given an introduction to the three-dimensional MRI images of mice brain : the first example deals with trisomic mice cerebellum and the second one with brain vessels network. We now focus on the second case: indeed, the quantitative measurement of three dimensional vessel network is now possible and MRI images provide a new method of evaluating many types of disease from the network geometry. We dedicate this special chapter 8 to the results description and give many issues concerning this model. These are particular images where the difficulties lie on the difference in the contrast between the different objects: for instance the contrast between vessels and the background is low, or inside vessel regions can vary from one region to another. The vessel's attributes include not only vessel number but also vessel morphological measures such as fibre structure, tortuosity and branching pattern etc. Moreover, the obtained MRI images contain a lot of noise. Therefore it is necessary to apply denoising methods while retain almost fiber structures of vessel.

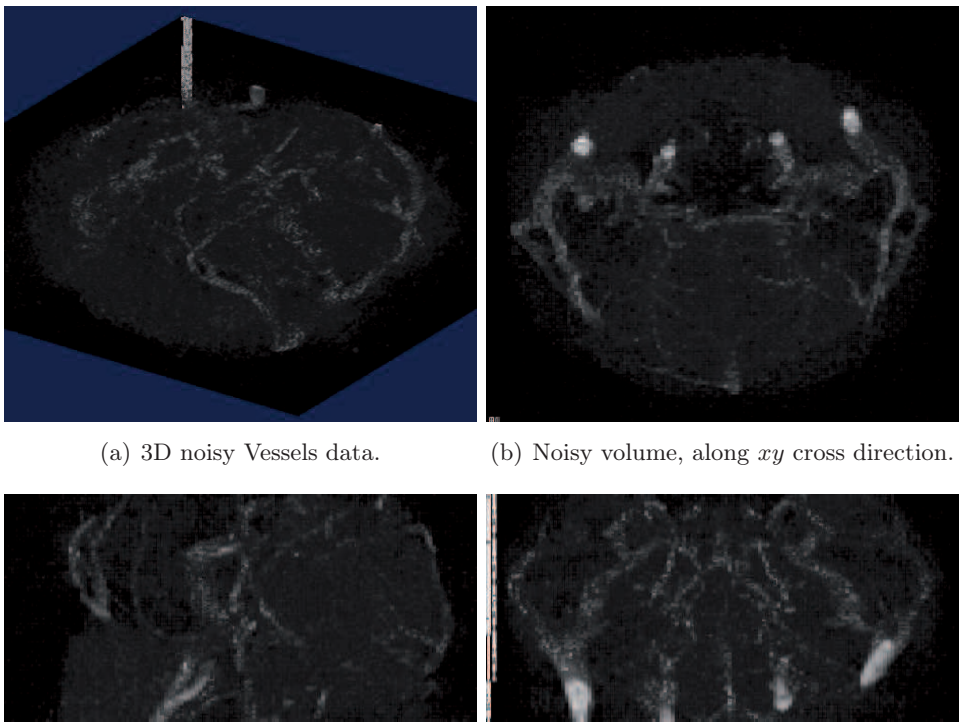
We thank J.C. Belœil, S. Même and F. Szeremeta, from CBM Laboratory[1] in Orléans, for using the data.

The vessels of mouse brain stack is composed of 51 two dimensional MRI slices. In this case, biologists want to recover the network of filament structures, especially the small ones inside a noisy volume. We want to to

[1] http://cbm.cnrs-orleans.fr/spip.php?rubrique48

differentiate the real network from the noise: the problem is to find where the noise is and where the interesting structures are. The images need to be positioned and oriented relative to one another and aligned exactly so that vessels are continuous through slices.

We here display again the 3D vessels volumes that have already been described in chapter 7 in Figure 8.1. The MRI images of vessels of Figure 8.1 are often affected by

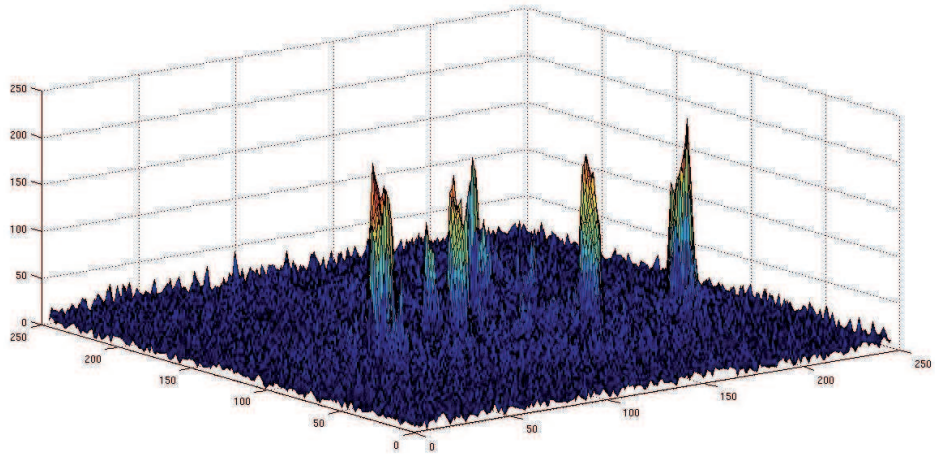


(a) 3D noisy Vessels data.

(b) Noisy volume, along xy cross direction.

(c) Noisy volume, along yz cross direction.

(d) Noisy volume, along xz cross direction.



(e) Surface plot of noisy Vessels model.

Figure 8.1: Original 3D vessel volume, MRI scans obtained from biologists

random noise arising in the image acquisition process. Because the noise of our image does not belong to any known classification noise (Gaussian, Salt and Pepper etc), it is a challenge to evaluate the results after noise reduction. In addition, we cannot compare our results to a ground truth image as it is the case for benchmark images as the famous "Lena", "Barbara" or "Cameraman" images in two-dimensions. The presence of noise does not only produce undesirable visual quality but also affects the visibility of low contrast objects. In such a situation it is very difficult for biomedical researchers to diagnosis a particular disease effect. Noise removal is essential in medical imaging applications to enhance and recover fine details that may be hidden in the data. This why reducing random noise is a very active research in medical image processing. Additionally, noise reduction methods developed in other research fields find their usage in biomedical applications. However, noise reduction must be carried out very carefully to avoid suppression of important image components. For the vessels data volume, it is a hard challenge to highlight structures such as mouse brain vessels that otherwise would be difficult to delineate their surroundings. Anyway, the results of biomedical image denoising has to be validated by experts.

The chapter sketches out the order of work as follows: first image denoising performance are considered. Then we apply the segmentation process to the vessels with the methods we have been previously considering. Since the data are very noisy, we first perform variational or wavelet based methods to reduce noise without staircasing and contour shapes are well indentified in numerical results. The schemes introduced in chapter 6 have been applied to detect and close contours to get the full vessels network. This makes the visualization easier by recovering volume prior to 3D reconstruction. Segmentation of vessels is essential for clinical assessment of human diseases, therefore it is challenging to perform image segmentation in angiography.

At last, the image decomposition model ($\mathcal{P}$) has been performed as well taking into account the theoretical studying of chapter 7. In the rest of this chapter, we present experimental results where the method gives very satisfactory results of separation into geometrical objects and the oscillating component. This process also confirms that our approach is well adapted to the 3D image of vessels data. Better numerical results will be obtained with more expensive computational work in the future.

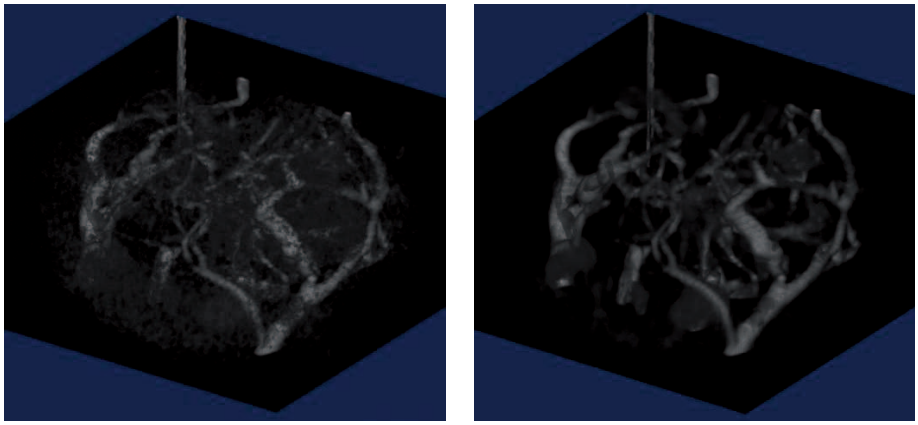
8.1 Application to Denoising

8.1.1 Denoising with variational method

In this section, we briefly recall how the variational second order model is applied to remove noise. This has been extensively described in chapters 2 and 3. We want to solve the following

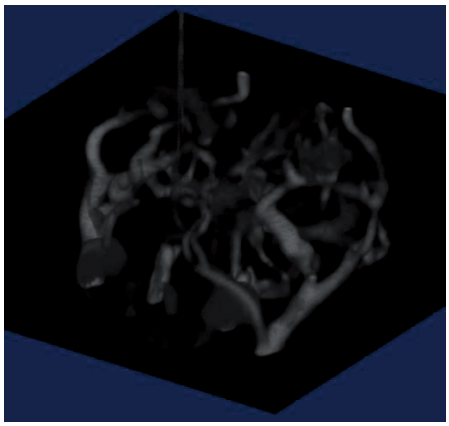
$$\inf_{u\in BV_0^2(\Omega)} \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV^2(u). \qquad (P_{ROF2})$$

The "isotropic" algorithm (withous any local anisotropy strategy) is first considered to computed the (denoised) solution to problem (P_{ROF2}). We use the Nesterov type algorithm (section 3.1.3 chapter 3) because its higher computation speed. We test different values of λ and the stopping criterion has been set to a maximal number of iterations can be chosen arbitrary large, here we have chosen $itmax = 5000$. It is of course possible to make the stopping criterion sharper, using for example the difference between two consecutive iterates of the minimizing sequence or the cost functional. However, we decided to work with the "original" Nesterov stopping criterion. The method is easy to implement and applicable to multidimensional images. It requires the user to view the resulting images and edit experimentally until obtaining the appropriate parameters. The automatic tuning of parameters is a hard task which is still widely opened. We first have to deeply understand the mathematical model behavior.

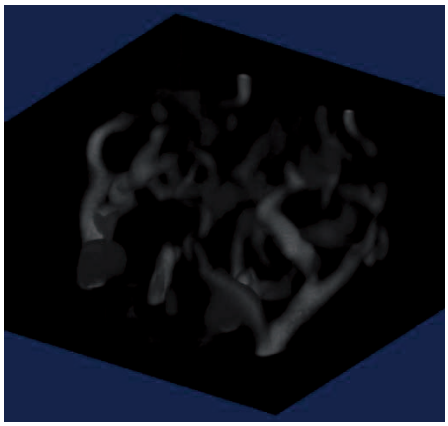


(a) $\lambda = 1$.

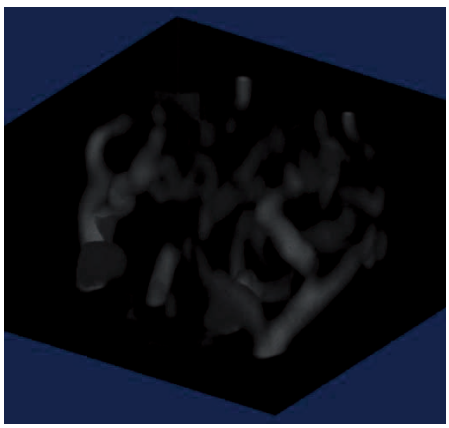
(b) $\lambda = 5$.



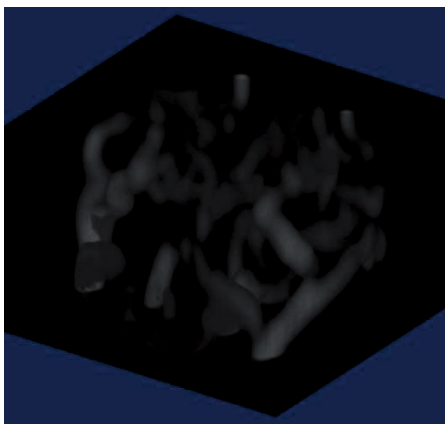
(c) $\lambda = 10$.



(d) $\lambda = 25$.



(e) $\lambda = 50$.



(f) $\lambda = 100$.

Figure 8.2: Denoised results performed by the isotropic algorithm for different value of λ.

From images of Figure 8.2, we see the influence of λ parameter. Image reconstruction is much more robust with respect to changes of the regularization pa-

rameter λ, which is an important condition when considering application in clinical practice. The larger λ gives us the smoother solution of vessel. Conversly, if λ is too small we may not remove enough noise. The value of $\lambda = 1$ is not the good choice, two next values $\lambda = 5$ and 10 give us the clear geometric information of vessel structure image without noise, the result is more satisfactory with λ is near to 10.

On the other hand, if λ is too large the scheme will remove too many features of the image. The larger next $\lambda \geq 10$ presented smooth results, but the visualized solutions give more and more blur feeling than expected, and the vessel filaments are very thick. Since there is no original image to compare to, tuning the parameter λ is time consuming since it often relies on experience and visual inspection. We do not have any automatic way for choosing λ as far as we know, so we choose λ in a reasonable range without being precise about the choice. Especially with the consideration of this vessel volume of medical images, this is a very significant issue. However, the choice of λ is dependent on the level noise. If we had the SNR information (from acquisition process for example) this would be helpful for the λ tuning.

Slices are displayed in Figure 8.3 to see the smoothing effect. Here we may conclude that the variational model (P_{ROF2}) removes noise significantly and almost contour shapes are well preserved in the smooth component. In addition, the TV^2 minimization scheme offers the best combination of noise removal and feature preservation. Moreover, surface viewing plots in Figures 8.3 show that stair-casing effect does not occur.

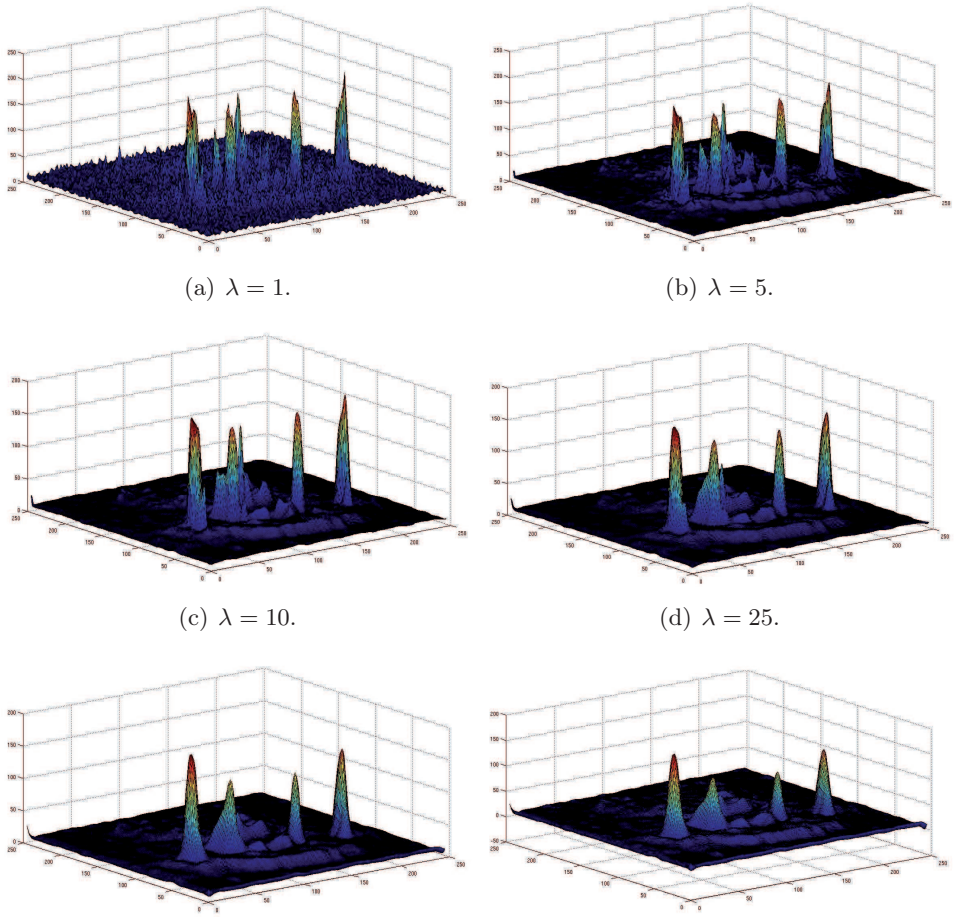
We give next the Peak-signal-to-noise ratio (PSNR), Mean Squared Error (MSE) or Mean Absolute Error (MAE). Table 8.1 performs comparison between these standard evaluations depending on λ. Consequently, as λ increases, the more blur we get and the bad quality of solutions gives a small value of PSNR. However, the PSNRs are near 30dB that could be acceptable, while MSEs and MAEs give the same conclusion.

In what follows, it is suggested to compute the ε, stands for the relative error between two consecutive iterates after $n = 5000$ iterations, in order to evaluate the results after our applied algorithm:

$$\varepsilon = \frac{\|u_{n+1} - u_n\|_{L^2}}{\|u_n\|_{L^2}} \tag{8.1}$$

λ	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$	$\lambda = 25$	$\lambda = 50$	$\lambda = 100$
PSNR	38.66	31.39	30.08	27.45	25.32	23.85
MSE	8.86	47.23	63.78	117.07	191.08	268.15
MAE	2.34	5.21	5.83	6.81	7.74	8.46
ε	8.27e-10	1.02e-06	2.32e-06	4.65e-06	7.77e-06	1.31e-05

Table 8.1: Comparisons of PSNR, MSE, MAE for denoising (P_{ROF2}), for different value of λ. The error values ε are included in.



(a) $\lambda = 1$. (b) $\lambda = 5$.

(c) $\lambda = 10$. (d) $\lambda = 25$.

(e) $\lambda = 50$. (f) $\lambda = 100$.

Figure 8.3: Surface plots of denoised results performed by the isotropic algorithm, test on the 3D MRI volume of vessel data, for different value of λ.

8.1.2 Denoising with wavelet based method

In the subsection we use the wavelet thresholding method described in section 5.2 chapter 5. We perform comparisons between different wavelet basis. The PSNR, MSE and MAE are reported in the following Table 8.2.

The numerical experimentation is performed to study the suitability of different wavelet basis. Results show that the wavelet scheme preserves details and offers a slightly high PSNR in the reconstruction. The proposed 3D discrete wavelet denoising algorithm has been evaluated on the noisy volume, by visual inspection and by computing quantitative measures of the similarity between the reference image and the denoised image The performance of the different wavelets is compared by computing the error criteria MSE, MAE and PSNR of the noisy image and the denoised image.

From Table 8.2 and the PSNR estimates we chose *Daubechies-5*, *Symlets-8* and *Coiflets-3* to display denoised results. Since the higher value of PSNR and small

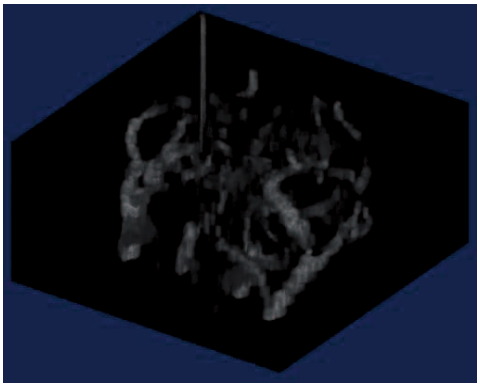
Wavelet basis	PSNR	MSE	MAE	Wavelet basis	PSNR	MSE	MAE
haar	30.53	57.53	4.90	*sym4*	30.86	53.30	4.93
db2	30.80	54.14	4.83	*sym5*	30.72	55.03	4.98
db3	30.78	54.39	4.91	*sym6*	30.86	53.30	4.84
db4	30.97	52.01	4.93	*sym7*	30.86	53.31	4.84
db5	**31.06**	50.99	4.83	*sym8*	**30.88**	53.04	4.94
db6	30.82	53.82	4.86	*sym9*	30.69	55.48	5.00
db7	30.73	54.91	5.05	*sym10*	30.87	53.24	4.88
db8	30.58	56.90	5.08	*coif1*	30.83	53.70	4.94
db9	30.76	54.64	4.93	*coif2*	30.93	52.47	4.92
db10	30.55	57.33	4.96	*coif3*	**30.96**	52.15	4.91
sym2	30.80	54.14	4.83	*coif4*	30.94	52.40	4.92
sym3	30.78	54.39	4.91	*coif5*	30.94	52.40	4.92

Table 8.2: Comparisons of PSNR, MSE and MAE, wavelet scheme testing on 3D MRI volume of Vessel model, for different wavelet basis.

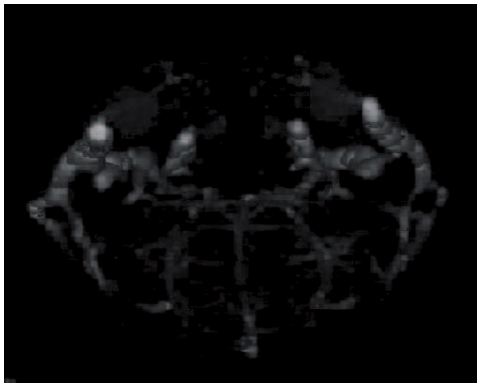
MSE indicate less noise power in the images, these wavelet basis are promising. In addition, most of wavelet basis reported in the Table 8.2 give a PSNR greater than 30dB which is considered as leading to a correctly reconstructed image. It is remarkable that for the vessel volume wavelet denoising strategy, the threshold parameter may be chosen by the histogram shape-based method (Appendix D), in which $T = 30$ for all wavelet basis implementation shown in Table 8.2.

Experimental results are displayed in Figures 8.4, 8.5, 8.6 and 8.7 respectively, where the *Haar, Daubechies-5, Symlets-8* and *Coiflets-3* are used. Denoised volumes are shown in detailed as three-dimensional viewing and through xy, yz and xz cross directions as well. Haar basis (Figure 8.4) gives the worst results: some piecewise effects appear in the results, that is not amazing because of the basis discontinuities. Moreover, the *Haar* gives lowest PSNR among all wavelet bases. The *db5*, *sym8* and *coif3* wavelet basis, gave satisfying denoised images. However, the *Daubechies-5* basis gives good results and less MSE values compared to others namely *Symlets-8* and *Coiflets-3*. We may conclude that Haar waveles arer not suitable. Higher order Daubechies basis have been tested and found to be more suitable.

The original image of vessel appears quite noisy. We cannot judge the performance by examining the PSNR as we do not have a noise-free image with which we can compare. However, by visual inspection it is evident that the denoised image, while removing a substantial amount of noise, suffers virtually no degradation in sharpness and details.



(a) The 3D Haar bases denoised data.

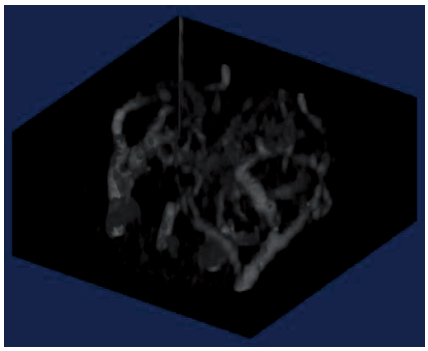


(b) One visual slice of xy cross direction.

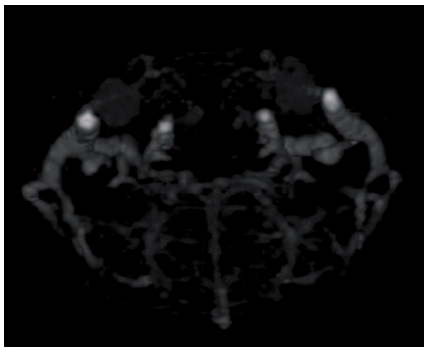
(c) One visual slice of yz cross direction.

(d) One visual slice of xz cross direction.

Figure 8.4: Denoising results using wavelet *Haar* basis.

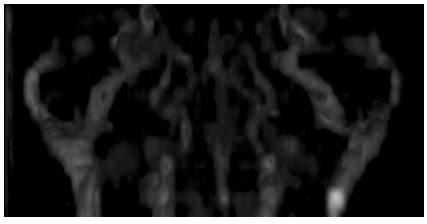


(a) The 3D *Daubechies* − 5 bases denoised data.



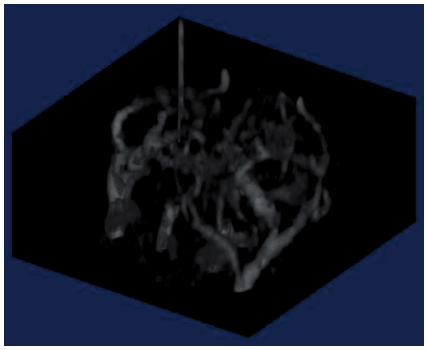
(b) One visual slice of xy cross direction.

(c) One visual slice of yz cross direction.

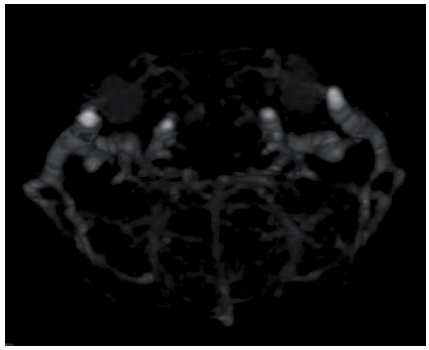


(d) One visual slice of xz cross direction.

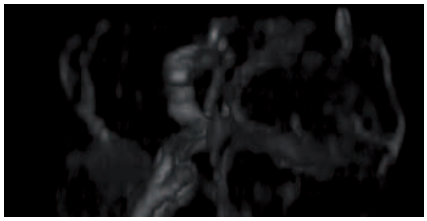
Figure 8.5: Denoising results using wavelet *Daubechies-5* basis.



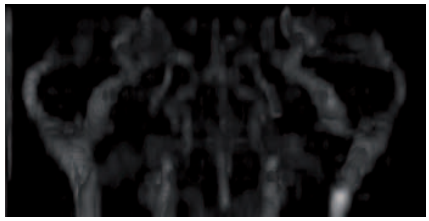
(a) The 3D $Symlets-8$ bases denoised data.



(b) One visual slice of xy cross direction.

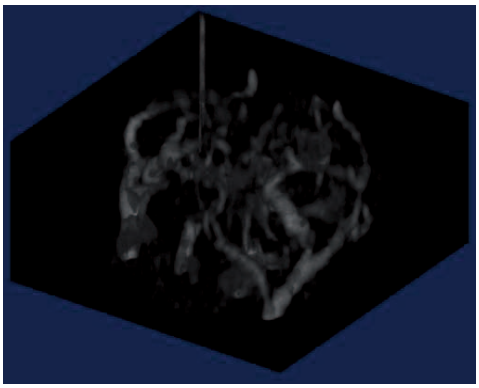


(c) One visual slice of yz cross direction.

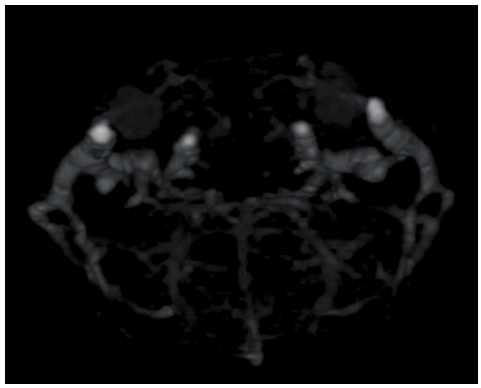


(d) One visual slice of xz cross direction.

Figure 8.6: Denoising results using wavelet *Symlets-8* basis.

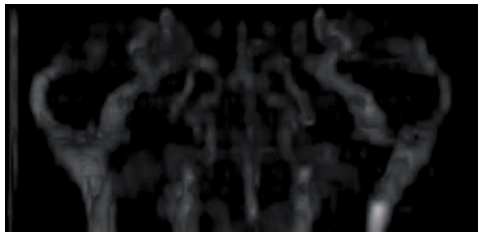


(a) The 3D $Coiflets-3$ bases denoised data.



(b) One visual slice of xy cross direction.

(c) One visual slice of yz cross direction.



(d) One visual slice of xz cross direction.

Figure 8.7: Denoising results using wavelet *Coiflets-3* basis.

8.2 Application to Texture Extraction

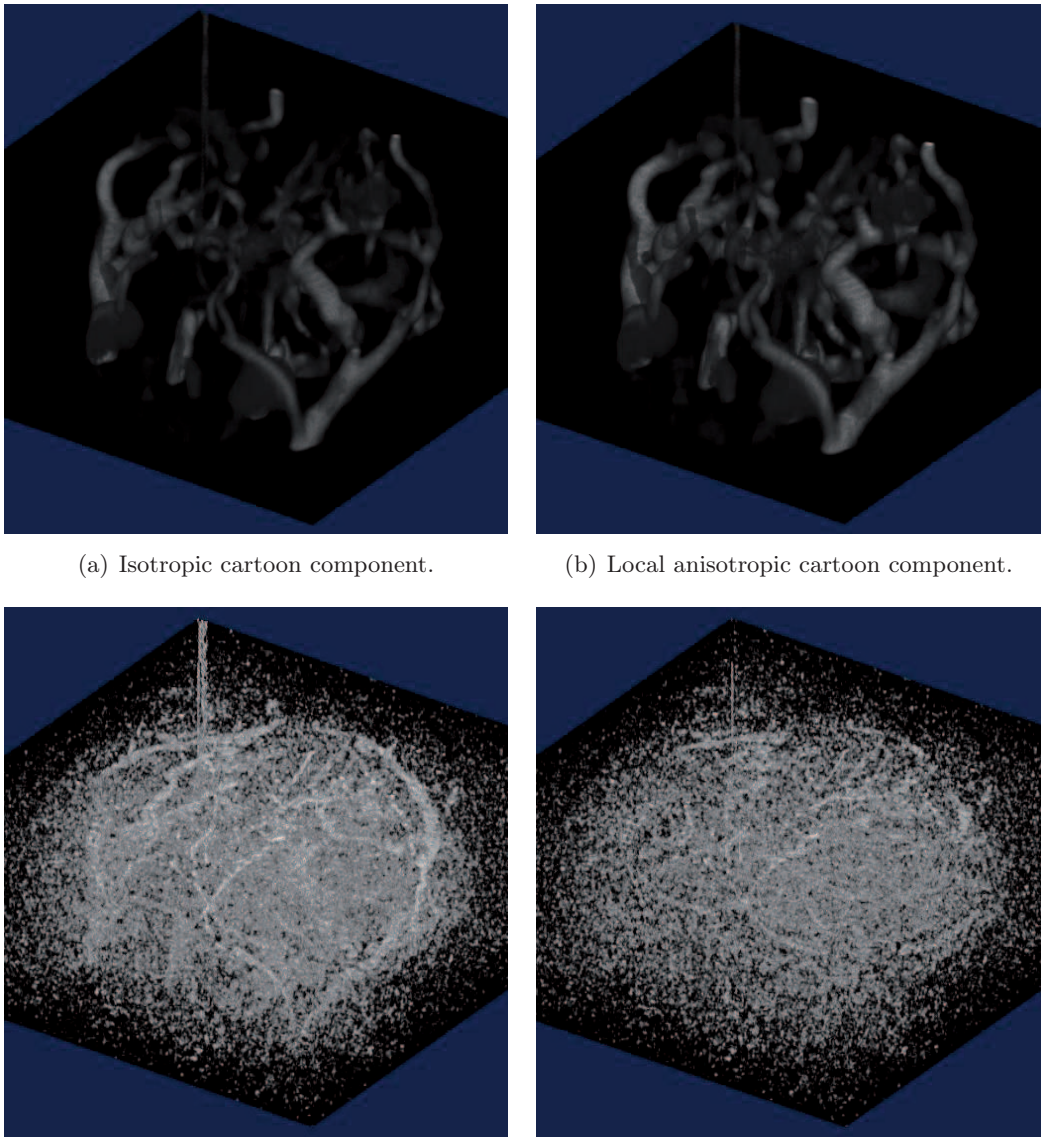
Observing the results derived from the variational model (P_{ROF2}) in Figure 8.2, we see that part of the geometrical information remains together with texture and noise in the oscillating component. In the sequel, our goal is to locate texture and/or noise so that we do not need to work with the cartoon part once it has been identified. We hope that we are able to keep most of texture and noise in the oscillating part while many contour shapes disappear. This the point we consider now, investigating the local anisotropic model for texture extraction.

In this section, we recall details from section 5.1.2 chapter 5 for the local anisotropic algorithm. An important feature of the (P_{ROF2}) local anisotropic model is that it takes the geometrical information of original images into account and keeps significant edges in the cartoon part. In fact significant contours are sharpened. Our consideration focuses on the volume of vessel data to compare with the previous isotropic algorithm efficiently. Inside the test, we set $\lambda = 10$. We report results in Figure 8.8. The examples running with $\lambda = 10$ gives us that 90% of vessel network is moved in the local anisotropic component, compared to the isotropic one. However, as we have tested with a limited number of iterations, there exists still (few) contour information inside oscillating part. This should be improved in the near future.

From the experimental results, we can see that a lot of geometrical information contour shapes of isotropic solution that remains in the oscillating part, while performing local anisotropic strategy let this geometrical information move to the cartoon part. Both schemes offer a goof combination of noise removal and feature preservation. In addition, oscillating views show that result after anisotropic algorithm is sharp whereas the fibre structures of vessel model are well located.

In Figures 8.9 and 8.10 we compare results for one visual slice of cartoon and oscillating components, respectively. It is clear that the oscillating part after isotropic method presents a lot of dynamic information remaining inside, meanwhile it has been removed after local texture modification of the second method.

As in the local anisotropic algorithm (section 3.1.3 chapter 3), thresholding process plays an important role. It decides if the method transfers well contour shapes or not and it may make us loose part of texture or keep some noise during determination contours process. The point is now to choose the optimal threshold parameter in this case, so that we may apply one of thresholding selection methods described in Appendix D.



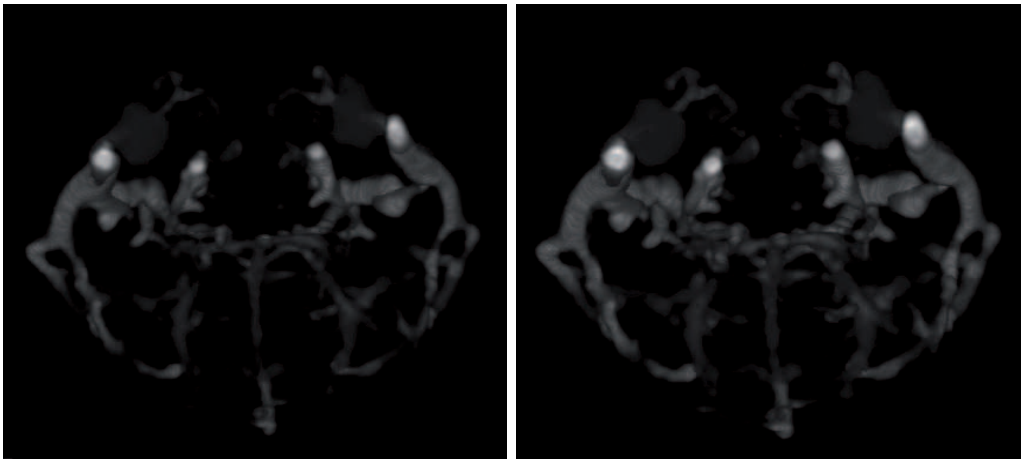
(a) Isotropic cartoon component.

(b) Local anisotropic cartoon component.

(c) Isotropic oscillating component.

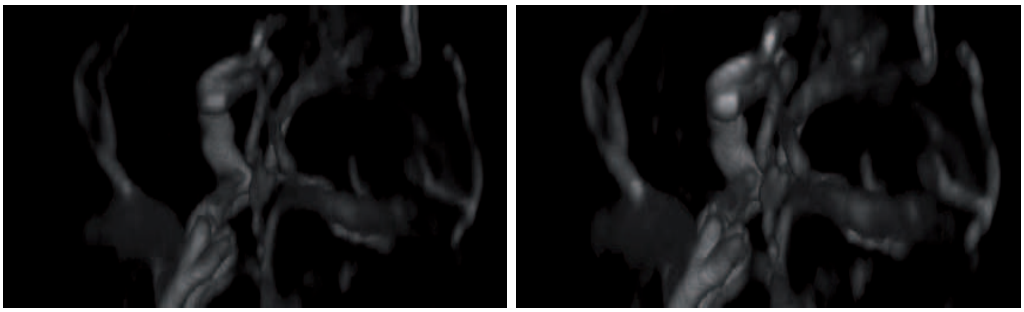
(d) Local anisotropic oscillating component.

Figure 8.8: Comparisons between the local anisotropic and isotropic algorithms, tested for a fixed $\lambda = 10$.



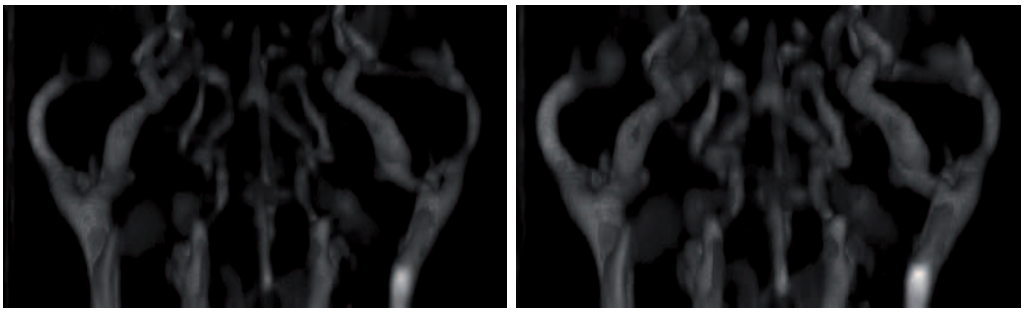
(a) Isotropic cartoon, xy cross direction.

(b) Local anisotropic cartoon, xy cross direction.



(c) Isotropic cartoon, yz cross direction.

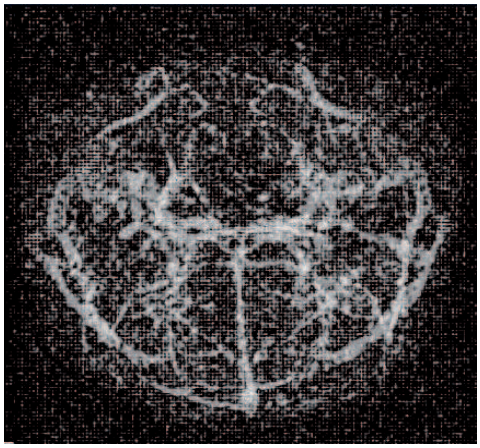
(d) Local anisotropic cartoon, yz cross direction.



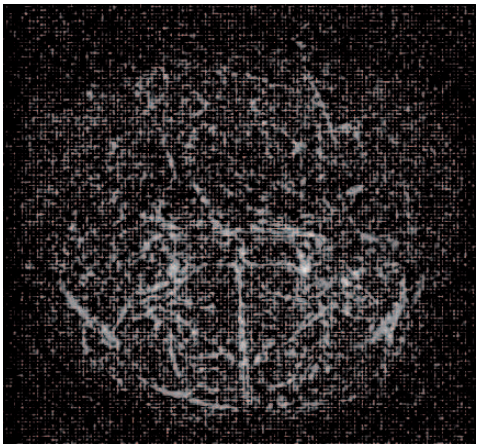
(e) Isotropic cartoon, xz cross direction.

(f) Local anisotropic cartoon, xz cross direction.

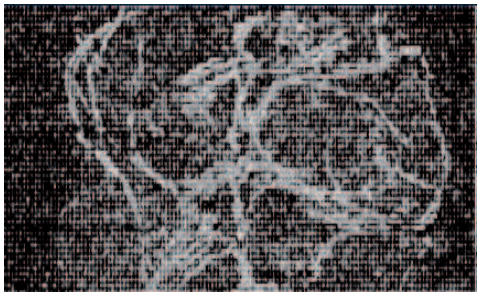
Figure 8.9: Comparison between isotropic and local anisotropic algorithms. Cartoon component- viewing in 3 crossing directions of a slice, $\lambda = 10$.



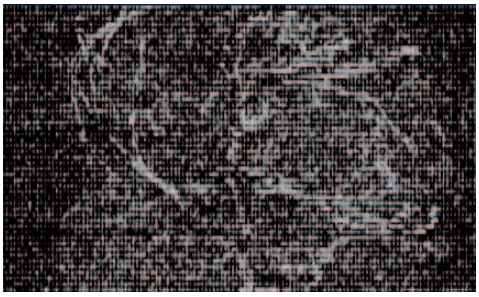
(a) Isotropic oscillating part, xy cross direction.



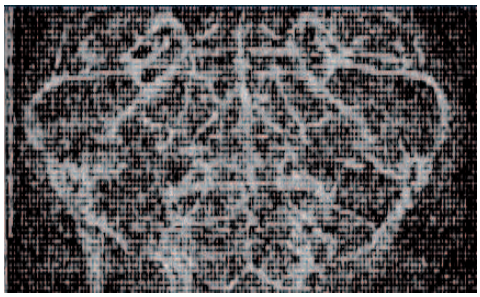
(b) Local anisotropic oscillating part, xy cross direction.



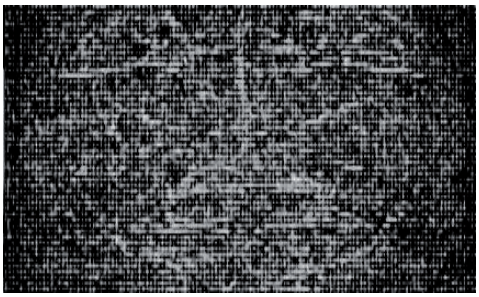
(c) Isotropic oscillating part, yz cross direction.



(d) Local anisotropic oscillating part, yz cross direction.



(e) Isotropic oscillating part, xz cross direction.



(f) Local anisotropic oscillating part, xz cross direction.

Figure 8.10: Comparison between isotropic and local anisotropic algorithms. Oscillating component- viewing in 3 crossing directions of a slice $\lambda = 10$.

8.3 Contour Detection

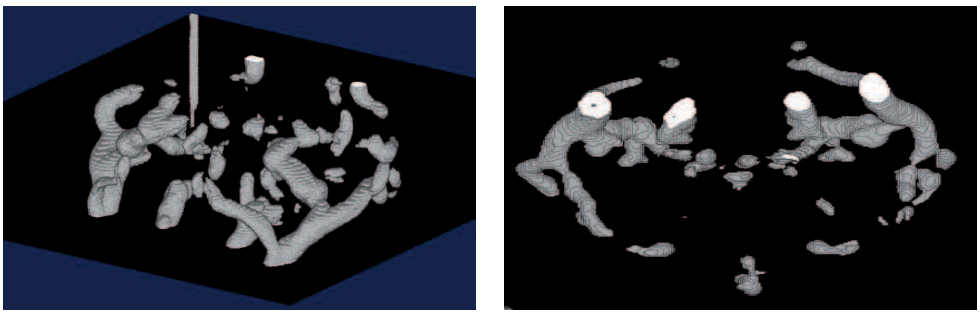
Biologists' interest to the vessel model after filtering is the structure of the network. Analysis of vessel networks in the brain is particularly important in improving our models and understanding of the relationships between blood structures. The segmentation process is "naturally" performed in the vessel system. So we expect to detect patterns, lines, edges, shapes based upon the visual information of our model. Many segmentation algorithms exist. However, we limit ourselves to using the variational model (P_{ROF2}) and wavelet based scheme in the segmentation procedure. These methods differ from previously known ones in that they use local and global vessel features cooperatively to segment the vessel network, [27], [14]...

We have already performed numerical tests in the previous chapter 6 on some 3D volumes,. In this section, we show that our methods are quite general and can be applied to any another type of medical images, e.g. MRI with different biological structures. Here we focus on the network of blood vessels structures that we want to recover. The network is reconstructed with segmentation from a huge three dimensional data set which can be very large (up to thousand pixels in each direction) to give a very good resolution in every two dimensional slice. The vessel tree is represented by a graph where each vessel keeps information about its relationship to other vessels. In addition, the images need to be positioned and oriented relative to one another and aligned exactly so that vessels are continuous through slices.

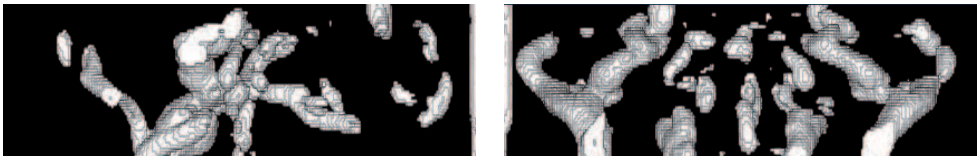
8.3.1 Contour Detection with variational methods

The contour shapes are computed from the difference between of two computed solutions: one comes from the variational isotropic algorithm and the other one from local anisotropic algorithm. Indeed, the local anisotropic improvement makes the contours moves from the oscillating component to the cartoon one. We note that contours are well identified and a threshold parameter is used to obtain a binary segmented image. Binary results are shown in Figure 8.11. They have been computed via a thresholding technique, where the histogram shape based method was applied to determine the threshold T.

The method has been tested on 3D vessels, with very good results. In the test, selected threshold is $T = 5$. For a small value of threshold, modified contours show well reconstructed vessel structure and topology inside. In Figure 8.11 we see that the vessel's network is almost recovered though the structures are not connected enough. We can observe that contour obtained with variational method are not continuous through slices. Vessels contrast is detected with a remarkable robustness, and most structures of vessel networks are preserved.



(a) 3D volume (b) xy cross direction

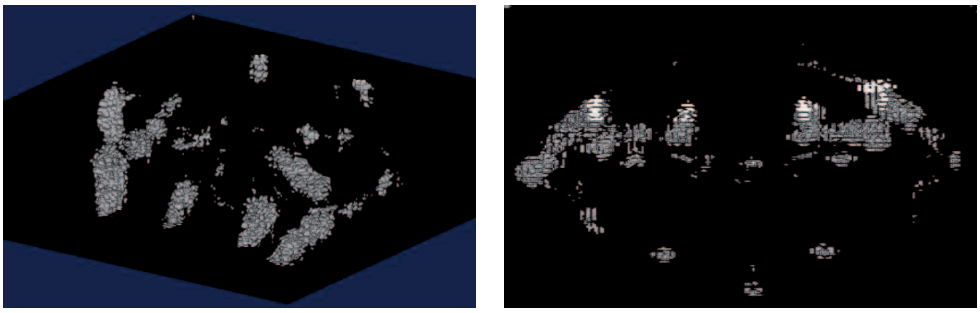


(c) yz cross direction (d) xz cross direction

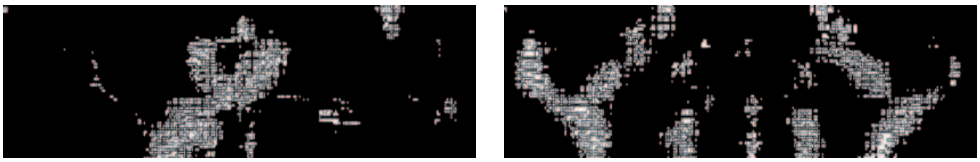
Figure 8.11: Contour shapes detected by the variational method, for $\lambda = 10$

8.3.2 Contour Detection with wavelet modification

8.3.2.1 Approximation coefficients replaced by zeros



(a) 3D volume (b) xy cross direction



(c) yz cross direction (d) xz cross direction

Figure 8.12: Contour shapes detected by the wavelet transform, approximation replaced by zeros, with *Symlets-8* basis

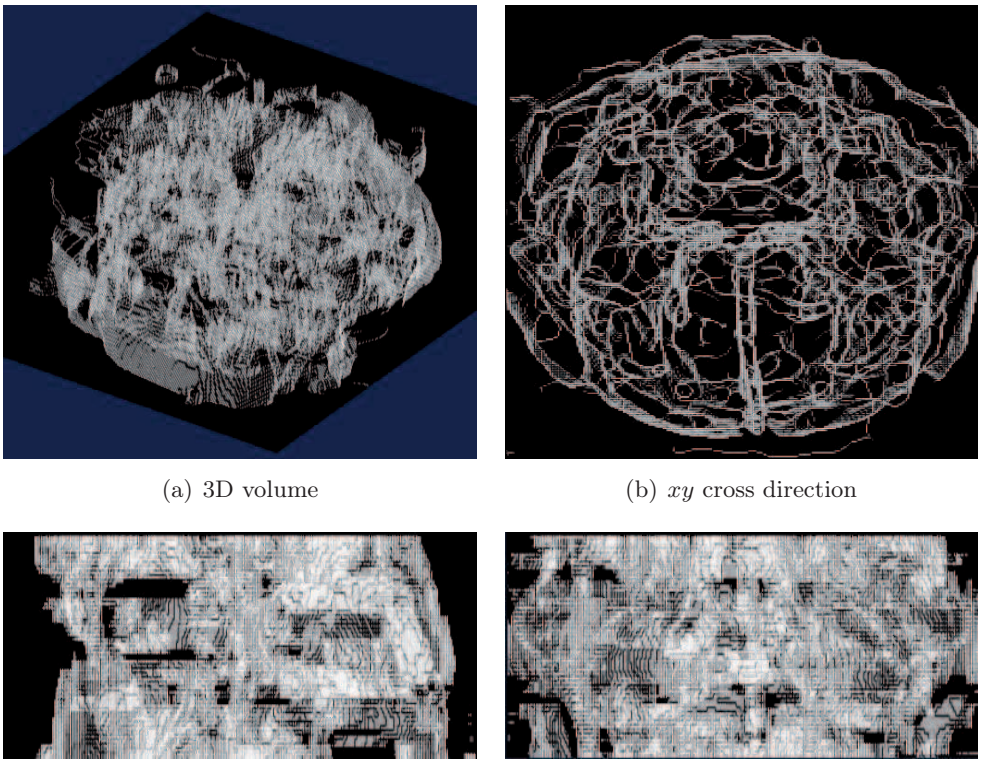
We present here the results we obtain with the method of section 6.1.2.1 chapter 6.We recall that the method is based on wavelet transform and that approximation coefficients are replaced by zeros before reconstruction. In this case, we use the

Symlets-8 wavelet basis and the result is displayed respectively in the Figure 8.12.

One sees that this zeros replacement can find only few contours. This detection method is not efficient though it is simple to implement. We may loose some contour details while getting rid of the approximation coefficients process. There are some small fragments and significant contour points cannot be found. In the Figure 8.11 we could see in their corresponding axial, sagittal and coronal views.

8.3.2.2 Modification of Approximation coefficients by Canny detector

As in the method described in section 6.1.2.2 chapter 6, the canny detector scheme can be used during the wavelet reconstruction process. This method provides good results: the use of Canny detector for approximation coefficients gives better results than in the previous algorithm, see Figure 8.13, where the orientation space represents at multiple directions.



(a) 3D volume (b) xy cross direction

(c) yz cross direction (d) xz cross direction

Figure 8.13: Contour shapes detected by the wavelet transform and modification of approximation coefficients by the Canny detector, test with *Haar* base.

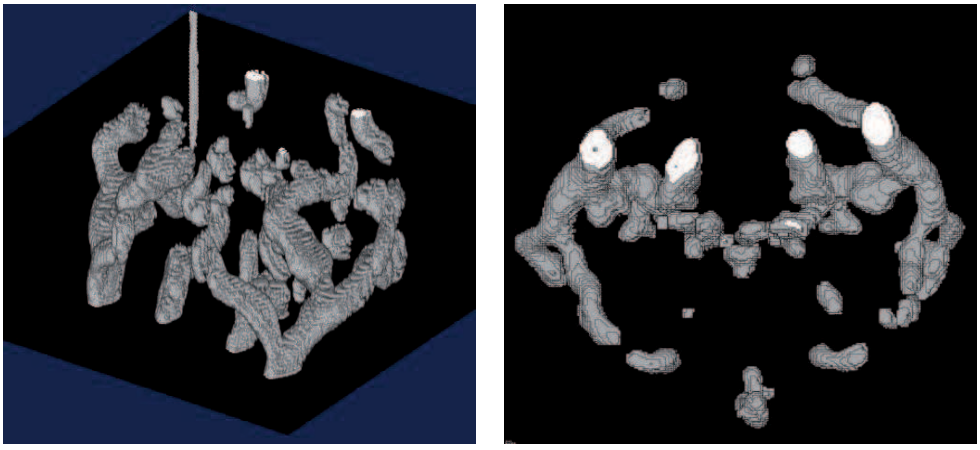
The algorithm has a good behavior but the obtained results are not acceptable. Some problems occur with corners and junctions, it seems that there are a lot of wrong branches that not belong to the vessel networks. The corner pixels look in the wrong directions with respect to its neighbors, leaving open ended edges, and missing junctions. It can be seen that, the vessel tree is represented from a lot of

branches. This makes the original vessel structure deformed. From Figure 8.13(b), main branches of vessels are not detected and distinguished from other tributaries. In addition, 3D viewer of segmented vessels in this figure does not give us a good visualisation.

8.4 Contour Closing

8.4.1 Hysteresis Thresholding

We have seen that once the binary transformation is performed contours are not clsoe. So we have to perform a contour closing method to get acceptable contours. Hysteresis procedures are described in section 6.2.1 of chapter 6. Here, thresholding is done using a double thresholding technique. We try to close available contours of vessel in Figure 8.11 obtained by the variational method. Therefore, the comparison is measured between vessel contour detected by variational model and proposed closing methods (see section 6.2.1). By this thresholding method, the connected pixels are selected, and then added to networks. In this case, the choice of two thresholds are established rather heuristically, and in this case their choice is $T_{low} = 2; T_{high} = 5$ respectively.



(a) 3D volume (b) xy cross direction

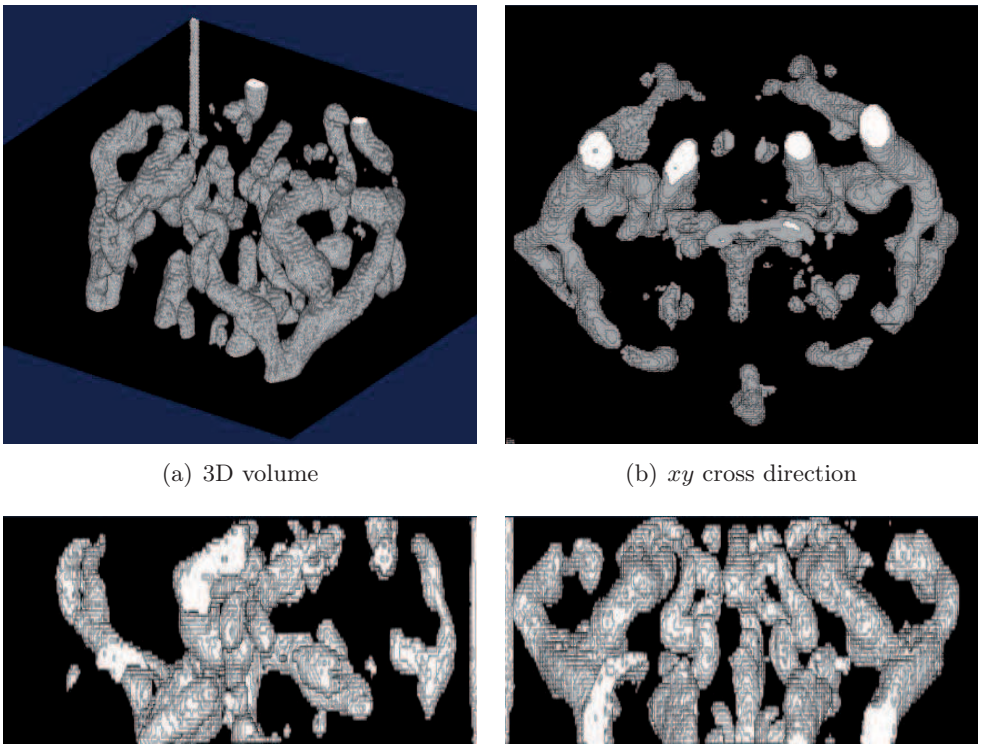
(c) yz cross direction (d) xz cross direction

Figure 8.14: Variational method closing contours by hysteresis thresholding, for $\lambda = 10$.

The algorithm is simple but requires improvements on the detection and threshold determination. Of course, in three dimensional visualization, this method does not give a good closed detected contour. Indeed, the vessel model is incorporated and some vessel properties such as the position and size of section and the curvature of the segment are shown in the formal structure model. However, this method describes a better closed visual measurement compared to the only one thresholding contours performance, see Figure 8.11. Consequently, results almost depend upon the two chosen thresholding. Threshold adaptation is governed by a learning algorithm that is based on the curve and consistency measurements around the pixels. That would be improved in the future further work with more accurate and more automatic techniques.

8.4.2 Local operator based on Chamfer Distance transformation

Depending on the success of our current strategy in two dimensions, we are optimistic that we will be able to achieve closed contours in three dimensions. In this section we test with the vessel contours that were detected in section 8.3. Figures 8.15 display closed contours of the local operator based on Chamfer distance transformation performance.



(a) 3D volume (b) xy cross direction

(c) yz cross direction (d) xz cross direction

Figure 8.15: Variational method closing contours based on Chamfer distance transformation, for $\lambda = 10$.

Numerical experiments show that the proposed frame based model performs successfully closed segmentations of vessels. Three crossing directions xy, yz and xz in Figure 8.15 show closed contours that make the vessel network continuous. It can be seen that the discontinuous contour shapes in Figure 8.11 are well resumed.

If we compare this method to the hysteresis thresholding procedure in Figure 8.14, some small vessel structures are prolonged what the hysteresis contour closing method did not provide. We conclude that such algorithm based on Chamfer distance transformation is able to extend the contours along the gradient norm with candidate belonging to the vessel path.

Throughout segmentation processes performed on vessels model, we did present a survey of vessel extraction techniques and algorithms. From that point, we can see the importance of vessel segmentation algorithms in some medical applications, such as radiological diagnostic systems, creating anatomical visualization, computer-aided surgery ... and other approaches. In the chapter, we have reviewed a few of segmentation methods focused on the vessels network.

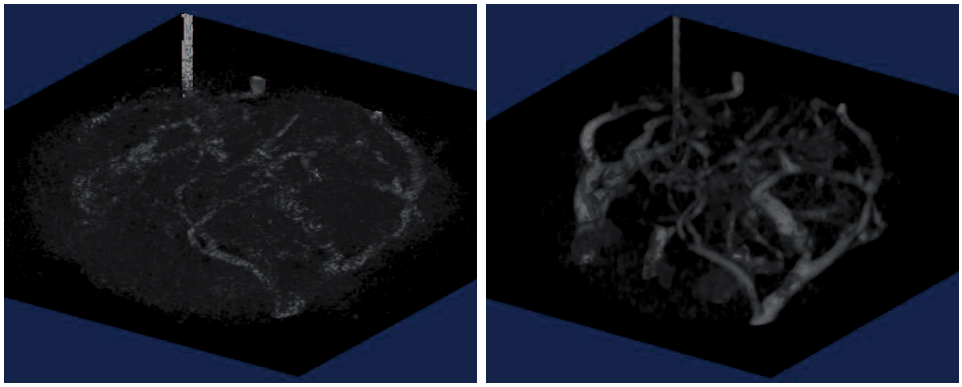
8.5 Image Decomposition model

In this section, we recall the decomposition that was proposed in chapter 7. We have to solve the problem:

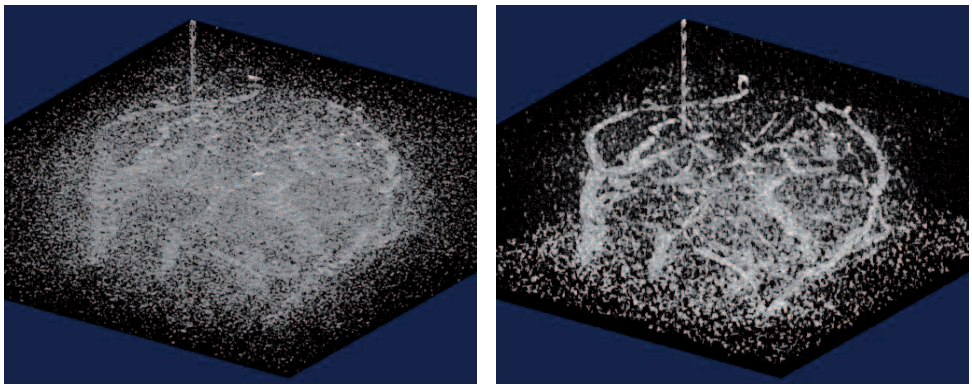
$$\inf_{(u,v)\in X^2} J_2(u) + B^*(v/\delta) + \frac{1}{2\lambda}\|f-u-v\|_X^2. \qquad (\mathcal{P})$$

The algorithm 6 is next applied to our exampe. It produces two components u (that contains geometrical information), v (that stands for the oscillating patterns) and the remaining term $\omega = f - u - v$. In Figure 8.16 below, we present results we obtained with $\lambda = 10$. Compared to numerical results obtained by the methods of previous sections 8.1.1 and 8.1.2, we note that the proposed new method produces better results. This was expected: the method gives very satisfactory results of separation into geometrical objects and the oscillating components.

We see in Figure 8.16 that the proposed model successfully decomposes the vessels. Considering the cartoon image u, it is evident that the geometric regions remain smooth. In that case, the component v contains most of the image noise. However, because of the limited number of iterations in the algorithm ($N = 500$) and the high computing cost with respect to CPU time and memory allocation for 3D data, we see that details of the texture that should be contained in ω still appear in v with noise component (see Figures 7.1 and 8.16). This will be further implemented in the near future, when the number of iterations is large enough to transfer all the details of the texture and geometrical information of the image need to be denoised.

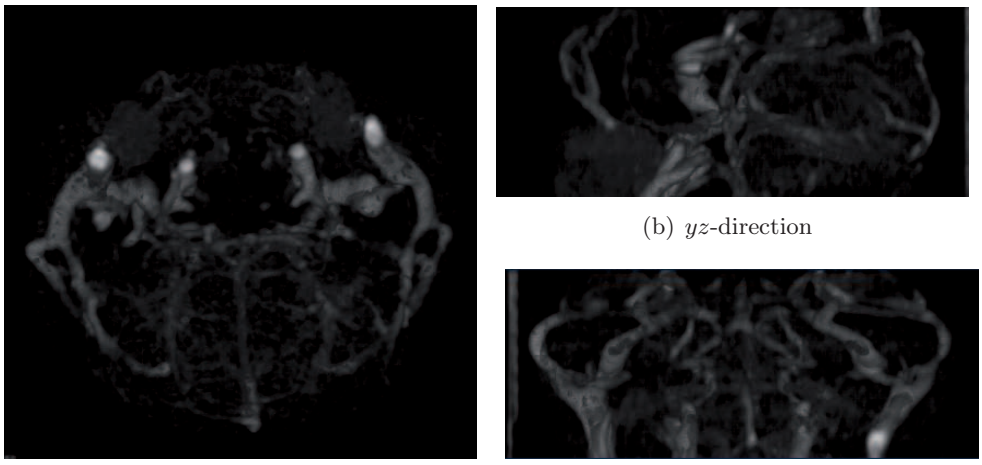


(a) Original image f (b) Component u



(c) Component v (d) Component $\omega = f - u - v$

Figure 8.16: Numerical test on the vessel of mouse brain, for $\lambda = 10$.



(a) xy-direction (b) yz-direction (c) xz-direction

Figure 8.17: Three cross-directions of the denoised image u, results from figure 8.16(b).

The image decomposition model separates the image into two components, one of them represents almost geometrical objects and the remaining component contains oscillating elements. The model automatically adapts to the local image information while preserving textured images. For instance, it is well adapted to our data (see figure 8.17 to evaluate the structure of denoised vessels). One can conclude that our approach seems very promising in many practical situations, where we want to separate two or more components. This will be investigated in the near future.

Chapter 9

Conclusions and Further Works

Conclusions

Ongoing work focuses on some examples of 3D medical MRI images that are dealt throughout the thesis: the mouse brain (cerebellum) volume and the vessels of mouse brain volume. As previously described, these obtained MRI models captured during scanning process contain a lot of noise, therefore it is necessary to apply some methods of denoising while almost dynamic geometrical information are preserved.

Work of segmenting the mouse brain 3D data (cerebellum) is a difficult issue since the contrast between different objects is low. On the other hand, for the vessels of mouse brain data, biologists want to recover the network of filament structures, especially the small ones inside a noisy volume. We want to know how to recover the real blood-vessels network (without noise). The problem is to identify the noise, since thin structures can be considered as noise as well. Images have to be positioned and oriented relative to one another and aligned exactly so that vessels are continuous through slices. For such volume data, we want to consider medical image processing from the mathematical point of view.

Our work is based on two methods: a second-order variational minimization model and the wavelet transform, which applications in image processing are image restoration, segmentation, decomposition strategies and so on. In addition, by considering these methods applied to our noisy MRI images, we can give some conclusions, comparisons, evaluations of advantages/disadvantages of each method. This helps to find the the most appropriate method for dedicated image processing. The principal contributions and conclusions of this work are summarized in the following paragraphs. We have included at the end of each chapter an extended summary and a short discussion of experimental results from the previous chapters is also detailed.

Usually, the considered variational model is the total-variation minimization model (that called the Rudin-Osher-Fatemi)(P_{ROF}). Because of the use of the total variation minimization the solution turns to be piecewise constant and geometrical elements, contours are not well preserved during the denoising process. It has been proposed to apply a second-order variational model (P_{ROF2}) that overcome these obstacles. In such situation, this promising model is highlighted in the thesis.

Observing many convincing results within the thesis, one can conclude that the second-order variational minimization model (P_{ROF2}) is quite efficient in image denoising. With the second-total regularization, the observed image will be divided into two components: one of them contains dynamic geometrical information and the other one is the oscillating component. Numerical solutions to problem (P_{ROF2})

show that the model keeps geometrical information and removes almost noise as well. Refer to [41], it is proved that this method keeps contour information in the cartoon componnent better than the classical model (P_{ROF}). However, this method also has some disadvantages in the implementation process. It strongly depends on the weighting parameter λ, which affects how much noise can be removed, and (as usual) implementation parameters (stopping criteria for example). In addition, the original "isotropic" algorithm is quite slow, especially in the context of 3D images with large size. In addition it is memory consuming and we have to develop a parallel implementation strategy.

Applying the "isotropic" algorithm of (P_{ROF2}) model, many contours remains together with the texture and noise in the oscillating part. That is the reason why we need a sharper variant called "local anisotropic" algorithm. With this new model we can keep most of texture and noise elements in the oscillating part while most of contour shapes are moved to the cartoon part. Based on s experimental results of video moving disk, cerebellum testing, it can be seen that the method helps us to save geometrical characteristics to the cartoon component. The method is very efficient to deal with the difficult vessels volume example: 90% of vessels network is reported in the anisotropic component (compared to the isotropic one). For instance, in section 8.2 of chapter 8, it is clear that the oscillating part after isotropic method present a lot of dynamic information remaining inside, while they disappear after the local texture modification of the second method. However, similar to the isotropic algorithm, the local anisotropic algorithm has also disadvantages since it depends on the choice of parameters. Besides that, the process of finding contour positions can affect our results. Indeed, the contour shapes depends on the thresholding process applying before the Hessian rotation step, specially in the case where the image is noisy. Another disadvantage of this local anisotropic algorithm in the 3D context lies in the fact that there are too many contour positions that can be found (maybe more than 1000.000 contour voxels for a 3D volume data). They should be performed one by one, this makes us have to choose very small number of iterations for every pixel processing. Thus, in the oscillating results there are some contour shapes which are not preserved successfully. This should be improved in the future, where each contour voxel is performed with enough increasing *itmax*, depending on the CPU time process.

Besides variational models, wavelet bases methods also play an important role in image processing today. Wavelets are used successfully in many applications such as multiresolution signal analysis, image analysis, communications systems, and other signal processing applications. The flexibility of wavelets makes them appropriate for many special purposes. This thesis deals with wavelet transform scheme and its several applications to image processing.

As we know, wavelet denoising uses multiresolution representation and noise separation features. By wavelet decomposition process, the wavelet coefficients are calculated. It is then possible to denoise by thresholding the coefficients, and the denoised image is reconstructed by the wavelet reconstruction process. Throughout this thesis, besides the study of subsampled wavelet transform (Mallat's algorithm),

we also give an understanding to undecimated wavelet tranform ("à trous" algorithm) therein. In chapter 5 and 8, some numerical results are performed with different wavelet bases: *Haar, Daubechies-p, Symlets-p* and *Coiflets-p*. We have successfully tested on the 3D cerebellum and vessels of brain models. The Wavelet package transform can be implemented further to achieve good performance. Ones can conclude that some wavelet applications in image denoising as follows:

- Based on the visualization image quality and Peak-Signal-to-Noise ratio (PSNR) measurements of cerebellum (in chapter 5) and the vessels model (chapter 8): The Mallat's algorithm gives better denoised results than the "à trous" algorithm.

- Among all types of wavelet basis, including *Haar, Daubechies-p, Symlets-p* and *Coiflets-p*, it can be seen that *Haar* gives us the worst denoised result compared to remaining basis.

- Based on the PSNR evaluation, we can conclude which wavelet basis is the best for image denoising, but this depends on the considered images. The higher value of PSNR gives better images. For the 2D+Time video of moving disk, the *symlets-10* gives better results than the other wavelet basis. For the cerebellum data denoising, the best wavelet basis is *symlets-6* and finally with the vessels model, we choose best basis as the *daubechies-5*.

- Although the wavelet hard-thresholding gives better PSNR performance, the wavelet soft-thresholding gives better image visual quality.

Using wavelet scheme to denoise image also has some disadvantages that can affect during denoising process. Wavelet procedure depends on thresholding selection step, a small threshold value will retain noisy coefficients whereas a large threshold values leads to the loss of coefficients that carry image details. Moreover, with our 3D large data, MATLAB wavelet toolbox need a long time to process results and compare them together. The work carried out here must be extended for the real time image denoising. Wavelet package transform can be implemented further to achieve good performance.

This thesis has focused on the studying of some applications to variational method of model (P_{ROF2}) and wavelet thresholding schemes separately. The main conclusions were that the (P_{ROF2}) model is outperformed by recent state of the art wavelet denoising methods but performs competitively with order wavelet methods. For the denoising criterion, outperformed results are well given but we have the blurry feeling of variational model (P_{ROF2}) more than the wavelet schemes. However, the approximation of wavelet performance keeps more features as well as eliminating most of the edges oscillations than the standard TV^2 denoising.

Each method has advantages and disadvantages: BV^2-variational method gives good denoised results meanwhile the wavelet shrinkage technique reduces well contour artifacts. In this situation we propose a new modified model that combines

these schemes. It permits us to improve another approach that employs variational framework, in particular the minimization model (P_{ROF2}) plus the wavelet thresholding to reduce oscillations, remove noise while maintain the sharpness of image. We have a reconstructed image has fewer oscillations near edges and noise is smoothed. Taking advantages of both two methods (variational method and wavelet based scheme) mentioned previously, we combine them in a new model. The numerical results show good denoising and preservation of most geometrical details. In chapter 7, we applied the method with images which noise can be removed successfully and contours, textures are well preserved. This model is particularly well adapted to denoise textured images or even images with complicated structures inside (vessel of brain for example). Throughout chapter 8 we gave a performance of this model applied to the vessel of mouse brain image, which give very interesting and promising results.

Second, in image segmentation, contour detection was also discussed in our work. We can perform it using these two methods. One concludes that the variational method gives us better results than the wavelet based method. The proposed contour detection scheme performs well since it is able to identify the curved and linear contours efficiently. However, the obtained contours can be either very thick or very narrow depending on the intensity across the contour and how much the original image was blurred. Moreover, some found contours are not closed and we may loose information of contours during thresholding process. This thesis contributes to extend some classical contour closing methods described in chapter 6 namely hysteresis thresholding and contour closing based on chamfer distance transform, in the 3D context. The chamfer based method is better than the hysteresis thresholding. This method can be used successfully to make preprocessing of the image segmentation operations.

Further Works

The work presented in this thesis proposes the following subjects for further research:

- Further study of the second-order total variation TV^2 and the second-order bounded variation space BV^2. It should be considered and found some relations between our proposed model with second-order total generalized variation TGV^2 and its corresponding space. In [59], this type of concept is also proposed in order to overcome the disadvantage of staircasing from problem (P_{ROF}). Also, the study of this new model increases the knowledge of variational minimization problems and application to image denoising.

- In the (P_{ROF2}) model in theoretical studying, there is a lot of open questions that have to be found: the existence an uniqueness solution without penalization terms has to be investigated together with a sharp analysis of the continuous model.

- Discuss again about algorithms applied to our variational model (P_{ROF2}) (the

fixed point and Nesterov-type algorithm), since our applied algorithms are still slow to implement, it is necessary to contribute another better algorithm, which provides a good solution, gives a acceptable speed of convergence and saves the CPU time performance.

- With the model of applying local anisotropic algorithm, each considered voxel has to be performed with a large number of iteration *itmax*. This depends on the CPU speed and time process with MATLAB software. In future, the work can be transfered into the language C or C++ to perform with larger data.
- On the other hand, in numerical point of view, our codes should be optimized in applied algorithms. We can choose another stopping criteria instead of the large number of iterations *itmax*. There are several ways to choose stopping criteria, one can focus on the choice of error ε between our solution at every step: $\|u_{n+1} - u_n\|$. That should be performed in the future.
- To compare the obtained experimental results together (in image denoising for example), we almost use visual inspection to compare the performance in preservation of geometrical information. Is is necessary to have other criteria to evaluate and compare these results which are not only reflected through the PSNR measurement.
- Besides the image denoising, considering image segmentation produces a significant amount of outliers. Therefore, the study of other segmented methods, should be considered and compared with our proposed methods.
- For representing the image data sets, we have used wavelet transform, a very basic and simple scheme. This possibilities can be further extended by other transforms, which is better than wavelet should be studied in future, such as: curvelets, beamlets transform and so on. This would be promising when considering our 3D medical image data.
- This thesis also describes a new 3D decomposition method which separates a 3D image into two components: the first one containing the geometrical structure of the image, the second one containing the noise. The proposed method is based on a second order variational model and an undecimated wavelet thresholding operator. The numerical implementation is described, and an experiment for denoising a 3D MRI image of cerebellum and vessels of mouse brain have been successfully performed in the rest of the thesis. It is promising in the future that we focus on extending this model to a more general approach, where image is divided into several components, in which each of them is characterized by a special functional space. Moreover, this will be investigated in near future, follows by testing more types of images, more iterations enough (that depends on the CPU speed of process) etc ...
- Moreover with the 3D data decomposition model, instead of using wavelet thresholding in the step of algorithm, we would like to try with another applying transform, such as: curvelet, beamlets. Because such kind of thresholding

transform is proved that represents edges/contours better than wavelet, specially in 3D, [16], therefore they should be performed for better results in the near future.

The research subject should be studied with the aim of improving proposed models, especially applied in medical imaging, one of the most promising and interesting fields of scientific investigation in image processing in future.

There still remains a lot of work to be done. It would be nice if the thesis has inspired its readers to contribute to the solution of some of the remaining open problems.

Appendix A

Wavelet Analysis

Throughout this Appendix, we shall recall the wavelet multiresolution representation, that mentioned by Mallat S., in [53], [54] since 1989. We recall and prove again some ideas about wavelet decomposition and representation theoretically.

A.1 Orthogonal Wavelet Bases

Definition 15. [53], [54] ***Multiresolution Analysis (MRA)***

Let $f(x) \in L^2(\mathbb{R}^N)$, for $N \geq 1$. A MRA *of $L^2(\mathbb{R}^N)$ is a pair $((V_j)_{j\in\mathbb{Z}}, \varphi)$, where V_j is a closed subspaces of $L^2(\mathbb{R}^N)$ and φ is a function of $L^2(\mathbb{R}^N)$, that satisfies the following properties:*

1. *$\forall j \in \mathbb{Z}$: $V_j \subset V_{j+1}$ (increasing),*
2. *$\forall j \in \mathbb{Z}$: $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$ (scaling),*
3. *$\bigcap_{j=-\infty}^{+\infty} V_j = \{0\}$ (separability),*
4. *$\overline{\bigcup_{j=-\infty}^{+\infty} V_j} = L^2\left(\mathbb{R}^N\right)$ (density),*
5. *The family $\left\{\varphi\left(.-k\right), k \in \mathbb{Z}^N\right\}$ is an orthonormal basis of V_0.*

Definition 16. [53], [54] ***Multiresolution Approximation of*** $L^2(\mathbb{R}^N)$

Let $A_{2^{-j}} f$ be the operator which approximates of function $f \in L^2(\mathbb{R}^N)$ at resolution 2^{-j}, for $N \geq 1$, which has properties as:

1. *$A_{2^{-j}} f(x)$ is the function which is the most similar to $f(x)$.*
2. *$A_{2^{-j}}$ is a linear function. The operator $A_{2^{-j}}$ is thus a projection operator on a vector space $V_j \subset L^2(\mathbb{R}^N)$. The vector space V_j - a set of all possible approximations at resolution 2^{-j} of functions in $L^2(\mathbb{R}^N)$.*

We call any set of vector spaces $(V_j)_{j\in\mathbb{Z}}$ which satisfies all properties of MRA a **multiresolution approximation of $L^2(\mathbb{R}^N)$.**

Remark 3. To avoid confusing between the resolution 2^{-j} and the scale 2^j, in the rest of this section from now, the notation of resolution is dropped and we denote simply $A_j f$ for the approximation of f at scale 2^j, also for space V_j the multiresolution approximation.

A.2 Wavelet Multiresolution transform for Signal Representation

Theorem 21. [53], [54] ***The existence of scaling function***

Let $(V_j)_{j\in\mathbb{Z}}$ be a multiresolution approximation of $L^2(\mathbb{R})$, $N = 1$ in this case. Then, there exists a function $\Phi \in L^2(\mathbb{R})$ is called a scaling function *and the family $\left\{\sqrt{2^{-j}}\Phi_{2^j}\left(x - 2^{-j}n\right)\right\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_j, where $\Phi_{2^j}(x) = 2^j\Phi(2^j x)$ is the dilation of $\Phi(x)$ by 2^j, for $j \in \mathbb{Z}$. Moreover, all translations of Φ, can be defined as $\Phi'(x) = \Phi(x - m)$, $m \in \mathbb{Z}$ also satisfy the properties of Φ above.*

This theorem is proved in [54]. By the second confirmation of theorem 21, the existence of scaling function Φ is not unique. It is simple to check that the function Φ' defined in theorem 21 verifies all properties of Φ. We point out this is true for $j = 0$, and we can prove similarily for another j. We then have the $\{\Phi_1(x - n)\}_{n\in\mathbb{Z}}$ is the orthonormal basis of V_0. As defined in theorem 21, for $m \in \mathbb{Z}$, the $\Phi'_1(x - n) = \Phi_1(x - n - m) = \Phi_1(x - (n + m))$. Therefore, the $\{\Phi'_1(x - n)\}_{n\in\mathbb{Z}}$ is also the orthonormal basis of V_0.

In Appendix C, we will point out different kinds of scaling function that are applied in Wavelets Transform scheme for image processing.

By theorem 21, the orthogonal projection on V_j can be computed by decomposing f on the orthonormal basis. We have:

$$\forall f(x) \in L^2(\mathbb{R}), A_j f(x) = 2^{-j} \sum_{\substack{n=-\infty \\ n\in\mathbb{Z}}}^{+\infty} \left\langle f(x), \Phi_{2^j}\left(u - 2^{-j}n\right)\right\rangle \Phi_{2^j}\left(x - 2^{-j}n\right) \quad \text{(A.1)}$$

where $n \in \mathbb{Z}$.

Let us denote

$$A_j^d f = \left(\left\langle f(u), \Phi_{2^j}\left(u - 2^{-j}n\right)\right\rangle\right)_{n\in\mathbb{Z}} \quad \text{(A.2)}$$

and we call $A_j^d f$ the *discrete approximation* of f at level j.

Remark 4. [54]

The inner product can be interpreted as a convolution as:

$$\left\langle f(u), \Phi_{2^j}\left(u - 2^{-j}n\right)\right\rangle = \int_{-\infty}^{+\infty} f(u)\,\Phi_{2^j}\left(u - 2^{-j}n\right)du = \left(f(u) * \Phi_{2^j}(-u)\right)\left(2^{-j}n\right)$$

and we can rewrite $A_j^d f$:

$$A_j^d f = \left(\left(f(u) * \Phi_{2^j}(-u)\right)\left(2^{-j}n\right)\right)_{n\in\mathbb{Z}} \quad \text{(A.3)}$$

A.2.1 How to compute the next $A_j^d f$ from $A_{j+1}^d f$, for $j < 0$?

Let $(V_j)_{j\in\mathbb{Z}}$ be a multiresolution approximation and $\Phi(x)$ the corresponding scaling function. As in theorem 21, we have family $\left\{\sqrt{2^{-j-1}}\Phi_{2^{j+1}}\left(x - 2^{-j-1}n\right)\right\}_{n\in\mathbb{Z}}$ is an

orthonormal basis of V_{j+1}, so:

$$\Phi_{2^j}\left(x-2^{-j}n\right)=2^{-j-1}\sum_{\substack{k=-\infty\\k\in\mathbb{Z}}}^{+\infty}\left\langle\Phi_{2^j}\left(u-2^{-j}n\right),\Phi_{2^{j+1}}\left(u-2^{-j-1}k\right)\right\rangle\Phi_{2^{j+1}}\left(x-2^{-j-1}k\right) \tag{A.4}$$

Hence,

$$\left\langle f\left(u\right),\Phi_{2^j}\left(x-2^{-j}n\right)\right\rangle=2^{-j-1}\sum_{\substack{k=-\infty\\k\in\mathbb{Z}}}^{+\infty}\begin{array}{l}\left\langle\Phi_{2^j}\left(u-2^{-j}n\right),\Phi_{2^{j+1}}\left(u-2^{-j-1}k\right)\right\rangle\\ \left\langle\,\Phi_{2^{j+1}}\left(x-2^{-j-1}k\right),f\left(u\right)\right\rangle\end{array} \tag{A.5}$$

We consider the inner product:

$$\begin{aligned}
&2^{-j-1}\left\langle\Phi_{2^j}\left(u-2^{-j}n\right),\Phi_{2^{j+1}}\left(u-2^{-j-1}k\right)\right\rangle\\
&=\int_{-\infty}^{+\infty}2^{-j-1}.2^{j}\Phi\left(2^{j}u-n\right)2^{j+1}\Phi\left(2^{j+1}u-k\right)du\\
&=\int_{-\infty}^{+\infty}2^{j}\Phi\left(2^{j}u-n\right)\Phi\left(2^{j+1}u-k\right)du\\
&=\int_{-\infty}^{+\infty}\Phi\left(v\right)\Phi\left(2v+2n-k\right)dv\\
&=\int_{-\infty}^{+\infty}\frac{1}{2}\Phi\left(\frac{u}{2}\right)\Phi\left(u+2n-k\right)du\\
&=\left\langle\Phi_{2^{-1}}\left(u\right),\Phi\left(u-\left(k-2n\right)\right)\right\rangle
\end{aligned}$$

By changing variables in the inner product integral, one can show that:

$$\left\langle f\left(u\right),\Phi_{2^j}\left(x-2^{-j}n\right)\right\rangle=\sum_{\substack{k=-\infty\\k\in\mathbb{Z}}}^{+\infty}\begin{array}{l}\left\langle\Phi_{2^{-1}}\left(u\right),\Phi\left(u-\left(k-2n\right)\right)\right\rangle\\ \left\langle\,f\left(u\right),\Phi_{2^{j+1}}\left(u-2^{-j-1}k\right)\right\rangle\end{array} \tag{A.6}$$

Therefore, by (A.6) we can easily calculate the approximation $A_j^d f$, for $j<0$ from the previous one $A_{j+1}^d f$.

Let H be the discrete filter whose impulse response is:

$$\forall n\in\mathbb{Z},h(n)=\left\langle\Phi_{2^{-1}}\left(u\right),\Phi\left(u-n\right)\right\rangle$$

and $\widetilde{h}(n)=h(-n)$. We will have:

$$A_j^d f=\left\{\left\langle f\left(u\right),\Phi_{2^j}\left(u-2^{-j}n\right)\right\rangle\right\}_{n\in\mathbb{Z}}=\left\{\alpha_{j,n}\right\}_{n\in\mathbb{Z}} \tag{A.7}$$

where

$$\alpha_{j,n}=\sum_{\substack{k\\k\in\mathbb{Z}}}\widetilde{h}\left(2n-k\right)\left\langle f\left(u\right),\Phi_{2^{j+1}}\left(u-2^{-j-1}k\right)\right\rangle$$

Theorem 22. [53], [54]

Let $\Phi(x)$ a scaling function and let h be a discrete filter with impulse response $h(n) = \langle \Phi_{2^{-1}}(u), \Phi(u-n) \rangle$. Let $H(\omega)$ be the Fourier series defined by:

$$H(\omega) = \sum_{\substack{n=-\infty \\ n\in\mathbb{Z}}}^{+\infty} h(n)\, e^{-in\omega}$$

$H(\omega)$ satisfies:

$$|H(0)| = 1 \tag{A.8}$$

$$|H(\omega)|^2 + |H(\omega+\pi)|^2 = 1. \tag{A.9}$$

Conversely, let $H(\omega)$ be a Fourier series satisfies (A.8) and (A.9) and such that $H(\omega) \neq 0$ for $\omega \in [0, \pi/2]$. Then, the function $\hat{\Phi}(\omega) = \prod_{p=1}^{+\infty} H(2^{-p}\omega)$ is the Fourier transform of a scaling function.

A.2.2 How to extract the difference of information between $A_{j+1}f$ and $A_j f$?

As in [54], the approximation of f at level $j+1$ and j are respectively equal to its *orthogonal projection* on V_{j+1} and V_j. The difference of information is called *detail signal* at level j. We point out that this difference is given by the orthogonal projection of the original signal on the *orthogonal complement* of V_j in V_{j+1}, which is denoted by O_j:

$$O_j \oplus V_j = V_{j+1}. \tag{A.10}$$

In order to compute the orthogonal projection of function f on O_j, we also need to construct an orthonormal basis of O_j.

Theorem 23. [53], [54]

Let $(V_j)_{j\in\mathbb{Z}}$ the multiresolution approximation, Φ the scaling function and H the corresponding conjugate filter, we define another filter G by $G(\omega) = e^{-\omega}\overline{H(\omega+\pi)}$. Let Ψ is the function whose Fourier Transform is given by:

$$\widehat{\Psi}(\omega) = G\left(\frac{\omega}{2}\right)\widehat{\Phi}\left(\frac{\omega}{2}\right) \tag{A.11}$$

and let $\Psi_{2^j} = 2^j\Psi(2^j x)$ denote the dilation of $\Psi(x)$ by 2^j. Then, $\left\{\sqrt{2^{-j}}\Psi_{2^j}\left(x - 2^{-j}n\right)\right\}_{n\in\mathbb{Z}}$ is an orthonormal basis O_j. Moreover, we have $\left\{\sqrt{2^{-j}}\Psi_{2^j}\left(x - 2^{-j}n\right)\right\}_{(n,j)\in\mathbb{Z}^2}$ is an orthonormal basis of $L^2(\mathbb{R})$. The function Ψ is called orthogonal wavelet function.

Then, we define P_{O_j} is the orthogonal projection on space O_j. By theorem 23 this term can be written:

$$P_{O_j} f(x) = 2^{-j} \sum_{\substack{n=-\infty \\ n\in\mathbb{Z}}}^{+\infty} \left\langle f(u), \Psi_{2^j}\left(u - 2^{-j}n\right)\right\rangle \Psi_{2^j}\left(x - 2^{-j}n\right) \tag{A.12}$$

As in the previous section, we define $D_j f$ as the detail of f at the scale j, as follows:

$$D_j f = \left(\left\langle f(u), \Psi_{2^j}\left(u - 2^{-j}n\right)\right\rangle\right)_{n\in\mathbb{Z}} = \left(\left(f(u) * \Psi_{2^j}(-u)\right)\left(2^{-j}n\right)\right)_{n\in\mathbb{Z}} \quad (A.13)$$

A.2.3 How to compute the next $D_j^i f$ from the $A_{j+1}^d f$, for $j < 0$?

For any $n \in \mathbb{Z}$, the function $\Psi_{2^j}(x - 2^{-j}n) \in O_j \subset V_{j+1}$, it can be expanded in orthonormal basis of V_{j+1} as:

$$\Psi_{2^j}\left(x - 2^{-j}n\right) = 2^{-j-1} \sum_{\substack{k=-\infty \\ k\in\mathbb{Z}}}^{+\infty} \left\langle \Psi_{2^j}\left(u - 2^{-j}n\right), \Phi_{2^j}\left(u - 2^{-j-1}k\right)\right\rangle \Phi_{2^{j+1}}\left(x - 2^{-j-1}k\right) \quad (A.14)$$

As in (A.6), by changing variables we obtain:

$$2^{-j-1}\left\langle \Psi_{2^j}\left(u - 2^{-j}n\right), \Phi_{2^j}\left(u - 2^{-j-1}k\right)\right\rangle = \left\langle \Psi_{2^{-1}}(u), \Phi\left(u - (k-2n)\right)\right\rangle \quad (A.15)$$

Taking the inner product of $f(u)$ both sides in (A.14) gives,

$$\left\langle f(u), \Psi_{2^j}\left(u - 2^{-j}n\right)\right\rangle = \sum_{\substack{k=-\infty \\ k\in\mathbb{Z}}}^{+\infty} \left\langle \Psi_{2^{-1}}^i(u), \Phi\left(u - (k-2n)\right)\right\rangle \left\langle f(u), \Phi_{2^{j+1}}\left(u - 2^{-j-1}k\right)\right\rangle \quad (A.16)$$

Let G be the filter with impulse response:

$$g(n) = \left\langle \Psi_{2^j}(u), \Phi(u - n)\right\rangle \quad (A.17)$$

and $\widetilde{G}$ be the symmetric filter with impulse response $\tilde{g}(n) = g(-n)$, we get:

$$D_j f = \left\{\left\langle f(u), \Psi_{2^j}\left(u - 2^{-j}n\right)\right\rangle\right\}_{n\in\mathbb{Z}} = \{\beta_{j,n}\}_{n\in\mathbb{Z}} \quad (A.18)$$

where

$$\beta_{j,n} = \sum_{\substack{k=-\infty \\ k\in\mathbb{Z}}}^{+\infty} \widetilde{g}(2n-k)\left\langle f(u), \Phi_{2^{j+1}}\left(u - 2^{-j-1}k\right)\right\rangle. \quad (A.19)$$

Therefore, we can easily compute $D_j f$ by convolving $A_{j+1}^d f$, $j < 0$. The orthogonal wavelet representation of a discrete approximation $A_0^d f$ can be computed by successively decomposing $A_{j+1}^d f$ into $A_j^d f$ and $D_j f$.

Finally, we summarize as follows:

$$A_J^d f : j = -J \leftarrow j = -J+1 \leftarrow \cdots \leftarrow j = -2 \leftarrow j = -1 \leftarrow j = 0 : A_0^d f.$$

Appendix B

Staircasing Effect

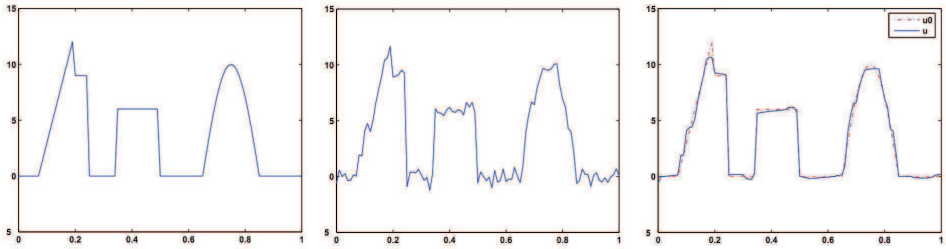
The *Staricasing effect* is a concept that appeared first in [20]. It is also observed on images denoised by TV minimization, for example the Rudin-Osher-Fatemi (ROF) model. The corresponding differential equation (PDE) of ROF problem is a second-order equation. Although the ROF model does an excellent job in removing noise, it often causes undesirable staircase effects since it favors solutions that are piecewise constant. Let us cite the definition of this *staircasing* from [11].

Let Ω be a measurable subset of $\mathbb{R}^n$, $n \geq 1$. We define the connectedness of a set $\Omega_1 \subset \Omega$ by the 4-pixel neighborhoods. One speaks of *staircasing effect* in the denoised version $\hat{u}$ of a noisy image u if there exists at least one connected region $\Omega_1 \subset \Omega$ containing at least two pixels, such that:

$$\forall x \in \Omega_1, \forall x' \in \mathcal{N}_x, x' \in \Omega_1 \Rightarrow \hat{u}(x) = \hat{u}(x') \tag{B.1}$$

where $\mathcal{N}_x$ denotes a neighborhood of the pixel x (both in the discrete and continuous settings).

The *staircasing effect* can be shown in figures in one and two-dimensions as follows:

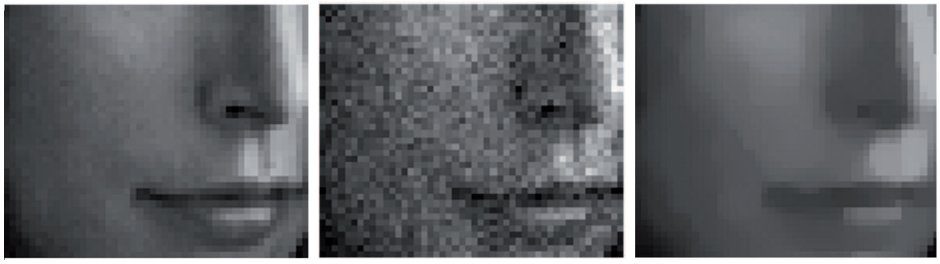


(a) Initial clean signal. (b) Gaussian noise with standard deviation 0.5, mean 0. (c) Recover from ROF model, SNR= 41.52db.

Figure B.1: One-dimensional ROF denoising, signal reproduced from [10].

Here, with staircase effect, smooth regions with noise are processed into piecewise constant regions after using the ROF model. ROF's solution fails to satisfy the eye and they can develop *false edges* that do not exist in the true image. Therefore, it is believed to develop another better approximation to a natural image than ROF variational model. We may get more examples and more details about "staircasing effect" as in [41], [11], [10].

In order to overcome this difficulty, some works have been proposed to reduce the staircasing effect seen in image denoised during TV regularization. In most cases



(a) Original image. (b) Noisy image. (c) Restoration by ROF model.

Figure B.2: Two-dimensional ROF denoising. A part of Lena image is emphasized to compare, reproduced from [10].

the emphasis is on designing new suitable regularization functionals which reduce staircasing as well as recover edges via retaining some form of TV regularization. A natural approach is to make the ROF model more convex in regions of moderate gradient (away from the edges), [7]. Another popular way to reduce staircasing is to introduce in some way higher order derivatives into the regularization term [37]...

Appendix C

Some types of Wavelet Bases

In this Appendix, we shall present some classical Wavelet bases that we used in Wavelet Denoising and Segmentation applications: Haar, Daubechies, Symlets and Coiflets, their scaling and mother wavelet functions in one-dimensional context.

C.1 Haar

Any discussion on wavelets begins with Haar wavelet, the first oldest and simplest one. Suppose that $\Phi(t)$ is a "hat" function satisfying:

$$\Phi(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

If we define the function $\Psi(t) = \Phi(2t) - \Phi(2t-1)$, we can obtain the following function:

$$\Psi(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq 1/2, \\ -1 & \text{if } 1/2 \leq t \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

The function $\Phi(t)$ is the Haar scaling function, and $\Psi(t)$ is the Haar mother wavelet. Then, the family:

$$\Psi_{j,n}(t) = 2^{j/2}\Psi(2^j t - n), \quad j, n \in \mathbb{Z}. \tag{C.1}$$

constitutes an orthonormal basis for $L^2(\mathbb{R})$. It means that, for any function $f \in L^2(\mathbb{R})$, we have:

$$f(t) = \sum_{\substack{n=-\infty \\ n\in\mathbb{Z}}}^{+\infty} \langle f, \Psi_{j,n} \rangle \Psi_{j,n}(t) \tag{C.2}$$

where the inner product in (C.2) is defined on interval $[0, 1]$.

To let the Haar series converge, the total number of data points is a power of 2. The basis functions are given by:

$$\Psi_{2k}[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 2k;\ 2k+1, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad \Psi_{2k+1}[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 2k, \\ -\frac{1}{\sqrt{2}} & n = 2k+1, \\ 0 & \text{otherwise.} \end{cases} \tag{C.3}$$

The Haar wavelet is not continuous and looks like a step function. Moreover it has the shortest support among all orthogonal wavelets. It is not well adapted to approximating smooth functions because it has only one vanishing moment. It represents the same wavelet as Daubechies *db1*.

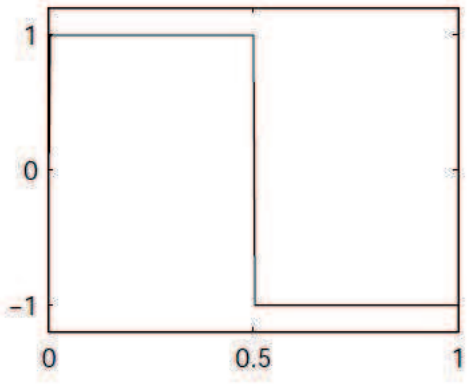


Figure C.1: Haar Wavelet.

C.2 Daubechies Wavelet

Ingrid Daubechies, in [68], invented what are called compactly supported orthonormal wavelets, thus making discrete wavelet analysis practicable. Daubechies wavelets are the most popular wavelets. They represent the foundations of wavelets and are used in numerous applications. Except for Haar basis, all of example of orthonormal wavelet bases consist of infinitely supported functions. Ingrid Daubechies constructed an orthonormal wavelet in which Ψ is compactly supported. The way to ensure compact support for the wavelet Ψ is to choose a scaling function Φ with compact support. Daubechies wavelets have a support of minimum size for any given number p of vanishing moments.

The names of the Daubechies family wavelets are written *dbp*, where p is the order of the wavelet. The *db*1 wavelet, as mentioned above, is the same as Haar wavelet.

First of all, find a sequence $\{\alpha_k : k \in \mathbb{Z}\}$ satisfying the following four conditions for all integer $N \geq 2$:

$$\alpha_k = 0 \quad \text{if} \quad \text{k} < 0 \quad \text{or} \quad \text{k} > 2\text{N} \tag{C.4}$$

$$\sum_{k=-\infty}^{\infty} \alpha_k \alpha_{k+2m} = \delta_{0m} \quad \text{for all integer m} \tag{C.5}$$

$$\sum_{k=-\infty}^{\infty} \alpha_k = \sqrt{2} \tag{C.6}$$

$$\sum_{k=-\infty}^{\infty} \beta_k k^m = 0, \quad 0 \leq m \leq N-1, \tag{C.7}$$

where $\beta_k = (-1)^k \alpha_{-k+1}$.

If $N = 1$, then $\alpha_0 = \alpha_1 = 1$, corresponding to the Haar basis. Therefore, the Haar basis is a simple case of Daubechies wavelet.

We can find a compactly supported scaling function $\Phi(t)$ from the above progression $\{\alpha_k\}$. The function $\Phi(t)$ is one solution of a functional equation:

$$\Phi(t) = \sum_{k=-\infty}^{\infty} \alpha_k \sqrt{2} \Phi(2t-k). \tag{C.8}$$

It is continuous and compactly supported and satisfies $\int \Phi(t)\, dt = 1$ for any integer N and the corresponding sequence $\{\alpha_k\}$. The support of $\Phi(t)$ is $[0, 2N-1]$.

Furthermore, if β_k is defined in condition (C.7), the function $\Psi(t)$ satisfies the functional equation:

$$\Psi(t) = \sum_{k=-\infty}^{\infty} \beta_k \sqrt{2} \Phi(2t-k). \tag{C.9}$$

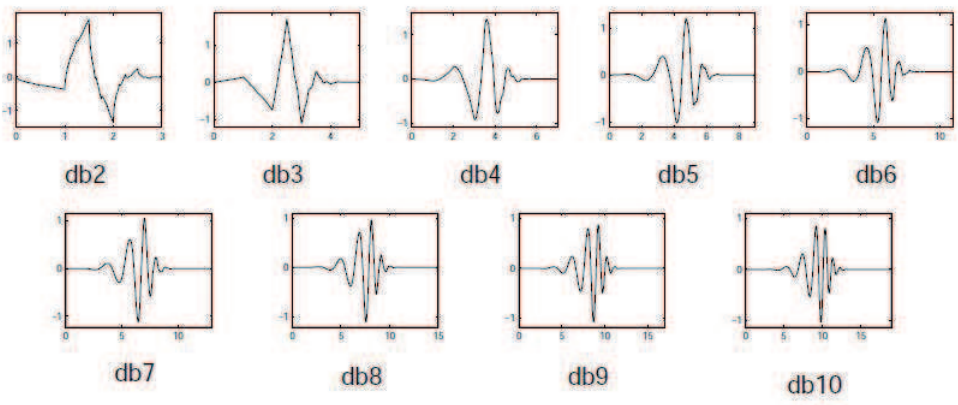


Figure C.2: Daubechies-p Wavelet.

C.3 Symlets

The Symlets are symmetrical, orthogonal and biorthogonal wavelets proposed by Daubechies as modifications to the *db* family. The resulting wavelets still have a minimum support $[-p+1, p]$ with p vanishing moments but they are more symmetric. And the properties of the two wavelet families Symlets and Daubechies are similar.

C.4 Coiflets

In 1989, R. Coifmann suggested the designed of orthonormal wavelet systems with vanishing moments for both the scaling and wavelet functions. They were first constructed by I. Daubechies in [26] and named *Coiflets*. A family of wavelets Ψ

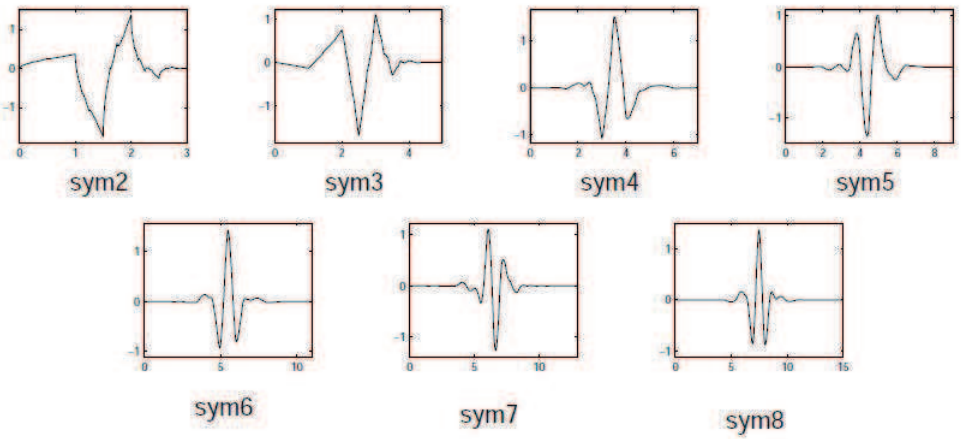


Figure C.3: Symlets-p Wavelet.

have p vanishing moments and a minimum size support, but the scaling function also satisfy:

$$\int_{-\infty}^{+\infty} \Phi(t)\, dt = 1$$

and

$$\int_{-\infty}^{+\infty} t^k \Phi(t)\, dt = 0, \forall 1 \leq k < p.$$

The wavelet function has $2p$ moments equal to 0 and the scaling function has $2p-1$ moments equal to 0.

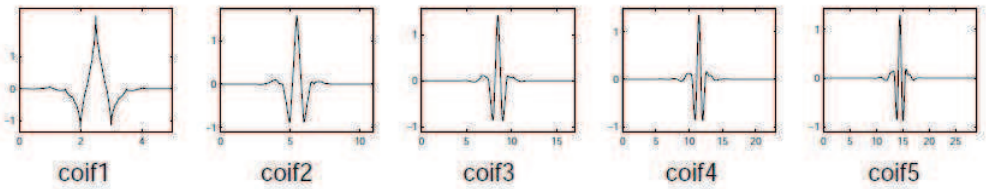


Figure C.4: Coiflets-p Wavelet.

The interesting property of this kind of wavelets is the interpolating characteristic of their associated scaling functions. This characteristic is due to the fact that both Coiflets and their scaling functions moments vanish. We can find more details about Coiflets in [65].

The wavelet equation produces different wavelet families, such as Morlet, Meyer, Shannon wavelets etc... Wavelets are classified into a family by the number of vanishing moments p. Within each family of wavelets there are wavelet subclasses distinguished by the number of coefficients and by the level of iterations. The filter

lengths and the number of vanishing moments for the four different wavelet families are tabulated in table C.1.

Wavelet family	Filter length	Number of vanishing moments p
Haar	2	1
Daubechies p	$2p$	p
Coiflets p	$6p$	$2p-1$
Symlets p	$2p$	p

Table C.1: Wavelet families and their properties.

Appendix D

Threshold Determination methods

Throughout this context, we keep a 3D volume in mind to define an image f over its domain $\Omega \subset \mathbb{R}^3$:

$$\begin{aligned} f : \Omega &\to \mathbb{R} \\ x &\mapsto f(x) \end{aligned}$$

where Ω is a discrete three dimensional space, indexing the grid points (voxels) on which gray levels $f(x)$ are observed.

Thresholding is an operation that converts a gray-scale image into a binary image where the two levels are assigned to pixels that are below or above the specified threshold value. Finding the correct threshold value to separate an image into desirable parts is a very important step in image processing. Thresholding method gives us the binary images, for they are restricted to values 0 and 1, indicating image background and object, respectively.

There exists a large number of gray level based segmentation methods using either global or local image information. In both cases, we can consider only one threshold for the whole image (global thresholding) or on the contrary we can establish different thresholds for each sub-region of the original image.

In thresholding, one assumes that the object can be characterized by its brightness, which is often a valid assumption for 3D datasets.

The most widely used of all possible segmentation method is the choice of value T such that $\min_x f(x) \leq T \leq \max_x f(x)$, and the segmented image is denoted g on Ω, can be defined:

$$g(x) = \begin{cases} 1 & ; \text{ if } f(x) \geq T, \\ 0 & ; \text{ otherwise.} \end{cases}$$

The problem (question) in this context is, how to select the value T, and whether there exists an optimal threshold value. There are a lot of different solutions to this threshold selection problem, each being based on different model assumptions. They have been studied extensively and a large number of thresholding methods have been published. Generally, with the wavelet denoising scheme, for images with Gaussian noise, the threshold can be approximated to $T = \sigma\sqrt{2\log_2(MNP)}$ where $M \times N \times P$ is the image's size, [2]. In our consideration of medical datasets (cerebellum and vessels of mouse brain), we do not have the value of σ, so the thresholding determination process is non-trivial step. In the works of Aujol [2], the value of σ can be estimated directly on the wavelet coefficients at the final scale. However, this estimation process should be studied and applied in the further works. In the scope of this Appendix section, we discuss about some these methods in thresholding selection. For instance, we mention two threshold choosing methods which are *Iterative method* and the *Optimal method from histogram analyze*.

D.1 Histogram shape-based method

Histogram based algorithms have been studied extensively. Regions with uniform intensity give rise strong peaks in the histogram.

Histogram shape can be useful in locating the threshold. It is the most direct and meaningful statistics of an image. In bi-level thresholding, the histogram of the image is usually assumed to have one valley between two peaks, the peaks representing background and objects, respectively.

In general, a good threshold can be selected if the histogram peaks are tall, narrow, symmetric and separated by deep valleys. Choosing a threshold in the valley between two overlapping peaks and inevitably some pixels will be incorrectly classified by the thresholding. A refinement of this technique is to recursively apply

Figure D.1: Threshold histogram choice.

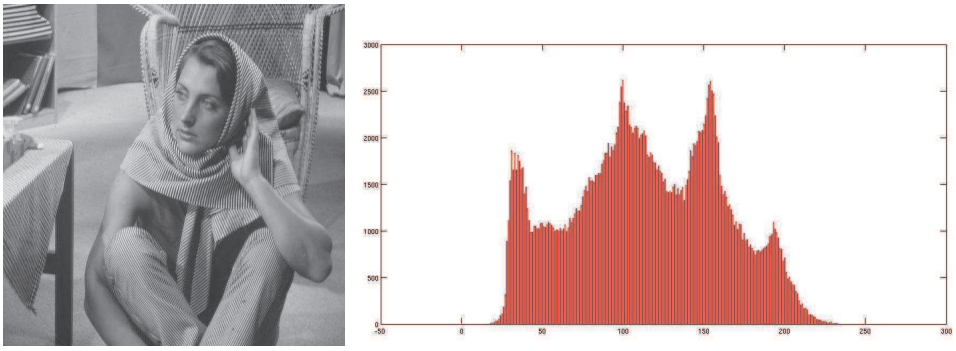
the histogram-seeking method to clusters in the image in order to divide them into smaller clusters. This is repeated with smaller and smaller clusters until no more clusters are formed. One disadvantage of the histogram-seeking method is that it may be difficult to identify significant peaks and valleys in the image. In this technique of image classifications distance metric and integrated region matching are familiar.

Histogram-based approaches can also be quickly adapted to occur over multiple frames, while maintaining their single pass efficiently. The histogram can be done in multiple fashions when multiple frames are considered. The same approach that is taken with one frame can be applied to multiple, and after the results are merged, peaks and valleys that were previously difficult to identify are more likely to be distinguishable.

Histogram can also be applied on a per pixel basis where the information results are used to determine the most frequent color for the pixel location. This approach segments based on active objects and a static environment, resulting in a different type of segmentation is useful in video tracking.

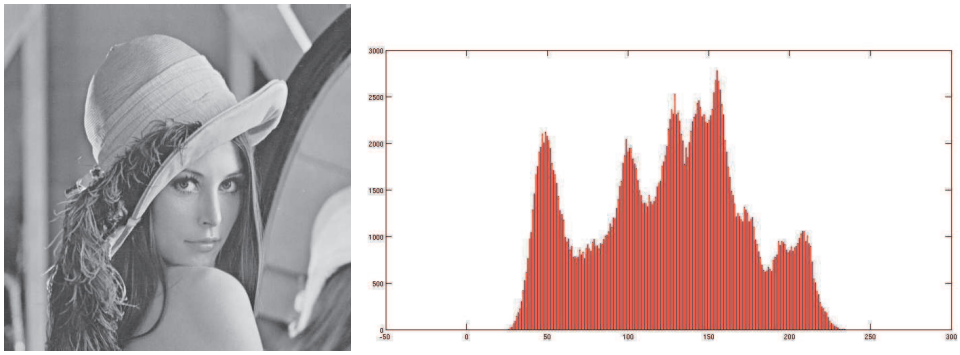
Histogram shape can be useful in locating the threshold. However, it is not reliable for threshold selection when peaks are not clearly resolved, a "flat" object with no discernible surface texture, and no color variation will give rise to a relatively narrow histogram peak.

The determination of peaks and valleys is non-trivial problem, then in 3D situation we could estimate the threshold values dependently on histogram shapes. In figures D.2, D.3, D.4 following we show some examples that choosing threshold values T by histogram shape-based.



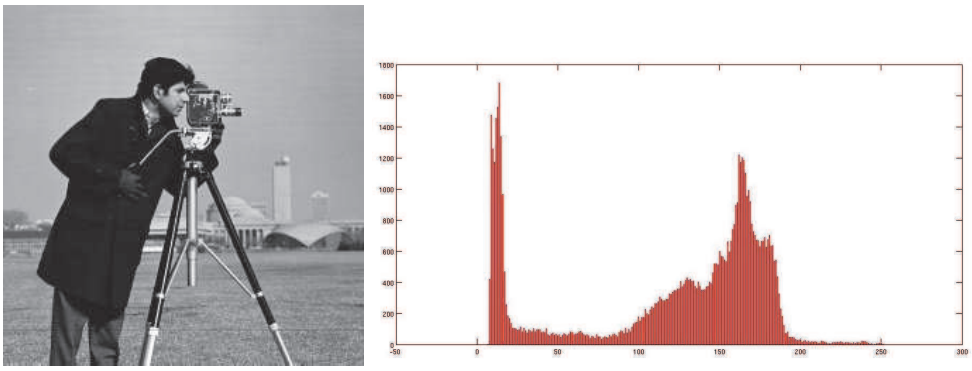
(a) (b)

Figure D.2: (a) Barbara 512×512, (b) The histogram of Barara.



(a) (b)

Figure D.3: (a) Lena 512×512. (b) The histogram of Lena.



(a) (b)

Figure D.4: (a) Cameraman 256×256. (b) The histogram of Cameraman.

D.2 Automatic Thresholding method (Iterative Threshold selection)

Method for automatic thresholding is the iterative isodata method, which is actually an application of the more general isodata clustering algorithm to the gray values of an image. This simple method would be to choose the mean or median value, the rational being that if the object pixels are brighter than the background, they should also the brighter than the average. In a noiseless image with uniform background and object values, the mean or median will work well as the threshold, however this will generally not be the case.

Given an initial threshold T^0 for example half of the maximum gray value or the average intensity of the image, the isodata algorithm can be stated as follows:

1. At the iteration i, generate binary image g^i from f using T^i.
2. Calculate the mean grey value μ_0^i and μ_1^i of the object and background voxels, respectively, we get the partitions R_1 and R_2.
3. Partition the image into two groups R_1, R_2 using threshold T^i.
4. Select a new threshold:
$$T^{i+1} = \frac{1}{2}\left(\mu_0^i + \mu_1^i\right). \tag{D.1}$$
5. Repeat step 2-4 until the mean values μ_0 and μ_1 in successive iterations do not change; we got the convergence.

In the table D.1 we present some thresholding values by applying this algorithm with 5000 iterations.

Images	Size	Threshold
Lena	512×512	$T = 21.7810$
Cameraman	256×256	$T = 51.3943$
Barbara	512×512	$T = 32.0270$
3D Mouse Brain	$341 \times 110 \times 100$	$T = 14.8756$
3D Small Cerebellum	$45 \times 87 \times 56$	$T = 40.2392$
3D Vessel	$256 \times 256 \times 64$	$T = 18.5776$

Table D.1: Thresholding values by iterative methods.

Appendix E

Non-Maximum Suppression and 3D Canny Detector

E.1 Non-Maximum Suppression

Non-maximum suppression is a process for marking all pixels whose intensity is not maximal as zero within a certain local neighborhood. Non-maximum suppression is often used with contour detection. After applying some contour detection methods, the obtained contour shapes can be either very thick or very thin depending on the intensity across the contour and how much the image was previously blurred.

Images are scanned along image gradient direction, and if pixels are not part of local maxima they are set to zero. This removes all image information that is not part of local maxima. For given estimates of image gradients, a search is carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction.

With a computed gradient magnitude at every pixel in 3D image, the local neighborhood can be performed that depends on the gradient angle (or pair of angles in 3D case; see Figure E.1). This neighborhood can be a linear window at different directions. Figure E.2 shows examples of linear windows at angles of $0^o, 45^o, 90^o$ and 135^o in two-dimensional space. Additional directions, such as $22.5^o, 67.5^o, 112.5^o$ and 157.5^o could be rounded to one of four directions above. Obviously, $180^o := 0^o$, $225^o := 45^o$, etc. This means angle θ in the ranges $[-22.5^o...22.5^o]$ and $[157.5^o...202.5^o]$ would be rounded to $\theta = 0^o$, and so on..., we could see more information in [63] and [15].

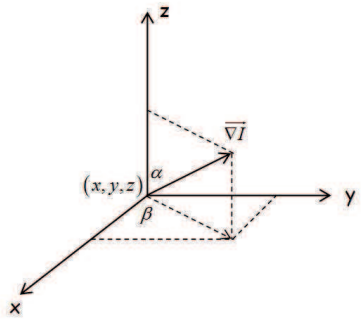


Figure E.1: A pair of angles (α, β) in three-dimensional space.

On the other hand, for 3D images the linear windows need to be oriented as in

figure E.3. A pair of angles (α, β) in figure E.1 could be computed and value of $0^o, 45^o, 90^o$ and 135^o are kept; 3D case can be extended from 2D case. We detail in figure E.3 the union of different linear windows.

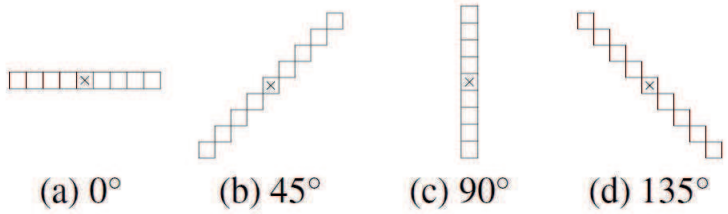


Figure E.2: 2D linear orientations, the $\times$ marked the pixel (x, y).

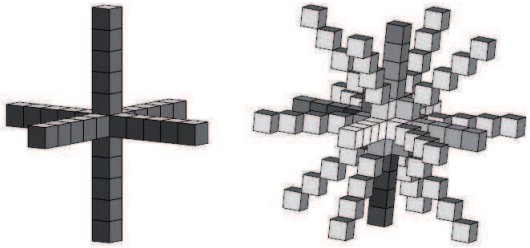


Figure E.3: 3D linear orientations.

The *non-maximum suppression* step keeps only pixels with the highest gradient magnitude. These maximal magnitudes should occur right at the contour boundary, and the gradient magnitude should fall off with distance from the contour shape.

In the 3D case, two angles of direction α (angle between gradient direction and its projection on plane Oxy), β (angle between this projection and Ox-axis) are described. As in figure E.3 we take the three pixels in a $3 \times 3 \times 3$ around pixel (x, y, z), and the principle for choosing neighbor pixels of (x, y, z) as follows:

- for $\alpha(x, y, z) = 0^o$, $\forall \beta(x, y, z) \in \{0^o, 45^o, 90^o, 135^o\}$: the pixels $(x, y, z - 1); (x, y, z); (x, y, z + 1)$ are examined.
- for $\alpha(x, y, z) = 45^o; \beta(x, y, z) = 0^o$: the pixels $(x - 1, y, z - 1); (x, y, z); (x + 1, y, z + 1)$ are examined.
- for $\alpha(x, y, z) = 45^o; \beta(x, y, z) = 45^o$: the pixels $(x-1, y-1, z-1); (x, y, z); (x+1, y + 1, z + 1)$ are examined.
- for $\alpha(x, y, z) = 45^o; \beta(x, y, z) = 90^o$: the pixels $(x, y-1, z-1); (x, y, z); (x, y+1, z + 1)$ are examined.
- for $\alpha(x, y, z) = 45^o; \beta(x, y, z) = 135^o$: the pixels $(x-1, y+1, z+1); (x, y, z); (x+1, y - 1, z - 1)$ are examined.

- for $\alpha(x,y,z) = 90^{o}; \beta(x,y,z) = 0^{o}$: the pixels $(x-1,y,z);(x,y,z);(x+1,y,z)$ are examined.
- for $\alpha(x,y,z) = 90^{o}; \beta(x,y,z) = 45^{o}$: the pixels $(x-1,y-1,z);(x,y,z);(x+1,y+1,z)$ are examined.
- for $\alpha(x,y,z) = 90^{o}; \beta(x,y,z) = 90^{o}$: the pixels $(x,y-1,z);(x,y,z);(x,y+1,z)$ are examined.
- for $\alpha(x,y,z) = 90^{o}; \beta(x,y,z) = 135^{o}$: the pixels $(x-1,y+1,z);(x,y,z);(x+1,y-1,z)$ are examined.
- for $\alpha(x,y,z) = 135^{o}; \beta(x,y,z) = 0^{o}$: the pixels $(x-1,y,z+1);(x,y,z);(x+1,y,z-1)$ are examined.
- for $\alpha(x,y,z) = 135^{o}; \beta(x,y,z) = 45^{o}$: the pixels $(x-1,y-1,z-1);(x,y,z);(x+1,y+1,z+1)$ are examined.
- for $\alpha(x,y,z) = 135^{o}; \beta(x,y,z) = 90^{o}$: the pixels $(x,y-1,z+1);(x,y,z);(x,y+1,z-1)$ are examined.
- for $\alpha(x,y,z) = 135^{o}; \beta(x,y,z) = 135^{o}$: the pixels $(x-1,y+1,z-1);(x,y,z);(x+1,y-1,z+1)$ are examined.

Then, if the pixel (x,y,z) has the highest gradient magnitude of the three pixels examined, it is kept as a contour. If one of the two other pixels but pixel (x,y,z) has a higher gradient magnitude, the pixel (x,y,z) is not on the center of contour and should not be classified as a contour pixel.

Consequently, non-maximal suppression method reduce thick contour responses to thin shapes. We could apply this step after the contour detection strategies to require a fixed number of comparisons per pixel regardless on suppression neighborhood size.

E.2 The 3D Canny Detector

In the past two decades several algorithms were developed to extract the contour in image, a lot of attention is focused to contour detection. Classically, the first stage of edge detection (the gradient operator, Robert, Sobel and the Prewitt operator) is the evaluation of derivatives of the image intensity). Smoothing filter and surface fitting are used as regularization techniques to make differentiation more robust to noise. Canny detector had the best performance and the best robustness in convergence. It is one of the faster executing detectors. It is a multi step detector which performs smoothing and filtering, non-maxima suppression, followed by a connected-component analysis stage to detect edges, while suppressing non edge filter responses.

The Canny edge detector was deviced to be an optimal edge detector, which satisfies all of the three performance criteria. The first and most obvious is low

error rate. It is important that edges occurring in images should not be missed and that there should be no responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. The third criterion is to have only one response to a single edge. This was implemented because the first two were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criteria, the Canny detector first smoothes the image to eliminate noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non maximum suppression). The gradient array is now further reduced by hysteresis (as in the chapter 6). Hysteresis is used to track along the remaining pixels that not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to be zeros (not contours), else if the magnitude is above the high threshold, it is kept an contour. And if the magnitude is between 2 thresholds, then it is set to zero unless there is a path from this pixel with a gradient above the low threshold.

The Canny edge detection algorithms is typically used as a two-dimensional edge detector, but has been generalized to three dimensions to detect surfaces. The Canny algorithm involves first a noise-reducing and edge-detecting operator to the image, then locally eliminates all but the maximal pixel or voxel outputs, then performs a hysteresis function to continue strong edges through areas of weak response.

A typical implementation of the Canny detector follows the steps below:

1. *Smoothing*: Canny algorithm first convolves an image with a filter that smooths the image or blurs image to eliminate noise. There are many possible variations of filters which effectively accomplish both tasks, with an appropriate Gaussian filter to reduce desired image details;

2. *Finding gradients*: Determine gradient magnitude and gradient direction at each pixel;

3. *Non-maximum suppression*: Once the appropriate operator is applied to the image, voxels with locally non-maximal intensity are eliminated. Only local maxima should be used to link edge points and deal out non-edge points, i.e., if the gradient magnitude at a pixel is larger than those at its two neighbors in the gradient detection, mark the pixel as an contour, otherwise, mark the pixel as the background.

4. *Contour tracking by hysteresis*: remove the weak contours by hysteresis thresholding, namely, final contours are determined by suppressing all contours that are not connected to a very certain (strong) contour.

Each step can be described more details in the following subsections.

E.2.1 Smoothing - Noise reduction

It is inevitable that all images taken from a camera will contain some amount of noise. To present that noise is mistaken for edges, noise must be reduced. The first step is to filter out any noise in the original image before trying to locate and detect any edges. Because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm.

This step will give us image reduced desirable amount of image details and noise, it optimizes the trad-off between noise filtering and contour localization. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating one square of pixels at time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected contours also increases slightly as the Gaussian width is increased.

E.2.2 Finding Gradient Magnitude and Angles

After smoothing the image and eliminating the noise, the next step is to find the contour strenth by taking the gradient of image at each pixel. We compute the derivatives $D_x(x, y, z)$, $D_y(x, y, z)$ and $D_z(x, y, z)$ of the image in the x, y and z directions.

Then, compute the gradient magnitude:

$$D(x, y, z) = \sqrt{D_x^2(x, y, z) + D_y^2(x, y, z) + D_z^2(x, y, z)}. \tag{E.1}$$

The direction of the contour shapes is computed using the gradient in x, y and z directions, the angles α and β in three-dimensional case, which are desbribed in figure E.1, can be calculated at every pixel:

$$\alpha(x, y, z) = arctan\left(\frac{D_z(x, y, z)}{\sqrt{D_x^2(x, y, z) + D_y^2(x, y, z)}}\right) \times \frac{180}{\pi}, \tag{E.2}$$

$$\beta(x, y, z) = arctan\left(\frac{D_y(x, y, z)}{D_x(x, y, z)}\right) \times \frac{180}{\pi}. \tag{E.3}$$

We next perform these angles into their rounded values that correspond to 8 possible directions when describing the surrounding pixels, which are well illustrated in figure E.3, then our angles $\alpha, \beta \in \{0^o, 45^o, 90^o, 135^o\}$.

E.2.3 Non-Maximum Suppression

After the contour directions are known, non-maximum suppression now has to be applied. Non-maximum suppression is used to trace along the contour direction and suppress any pixel value that is not considered to be a contour shape; this principle has been studied as in section E.1 before. Here we could summarize how to implement. If the gradient magnitude at a pixel is larger than those at its two

neighbors in the gradient derection, mark the pixel as a major contour. If the gradient magnitude at the pixel is larger than those pixels adjacent to it in any direction, mark the pixel as a minor contour. Otherwise, mark the pixel as the background.

This step partition the minor contours at the branch points, remove all branches that do not contain a major contour. Then rename as major contours the portions of minor contours that are deliminated by major contours. Combine newly obtained major contours with previously obtained major contours.

E.2.4 Contour tracking by Hysteresis thresholding

Some of the contour shapes detected by step $1-3$ will not be actually valid, but will just be the noise. We would like to filter this noise out. Eliminating pixels whose gradient magnitude D falls below some threshold removes the worst of this problem, but it introduces a new problem. Because, selecting a good value of threshold T is difficult, some false contour shapes will remain if T is too low, some contours will disappear if T is too high.

A simple threshold then may actually remove valid parts of a connected contour leaving a disconnected final contour image. This happens in regions where the gradient magnitude fluctuates between just above and just below the threshold. Hysteresis is one way of solving this problem.

Instead of choosing a single threshold, two thresholds T_{high} and T_{low} are used ($T_{low} < T_{high}$). Pixels with a gradient magnitude $D < T_{low}$ are discarded immediately. However, pixels with $T_{low} \leq D < T_{high}$ are only kept if they form a continuous contour shapes with high gradient magnitude (i.e. above T_{high}). We could apply the hysteresis thresholding algorithm in section 6.2.1 for this step.

Appendix F

Publication 1

Informations:

Bergounioux M., Tran M.P.,

A second-order model for 3D texture extraction,

Mathematical image processing, Springer Proc. Math.,

Volume 5, Pages 41-57,

Springer Berlin Heidelberg, 2010.

A second order model for 3D-texture extraction

Maïtine Bergounioux and Minh Phuong Tran

Abstract In this paper we present the 3D-implementation of a second-order model for texture extraction that has been fully described in [4]. Numerical experimentation has been performed for 2D-images. We generalize the discrete model to the 3D case. In particular we describe the whole discretization process. In addition, we add an algorithmic modification that improves texture extraction using a modified Hessian matrix. We end with numerical examples arising in biomedical imaging

1 Introduction

In this paper we present the 3D- implementation of a second-order model for texture extraction that has been fully described in [4]. Numerical experimentation was performed for 2D-images. We generalize the discrete model to the 3D case. In particular we describe the complete discretization scheme. In addition, we add an algorithmic modification that improves texture extraction significantly using a modified Hessian matrix. This is also a generalization of the 2D-case (see Piffet [10, 11]). First, we recall the main definitions and present the generic second order variational model. Section 2 is devoted to the 3D-discretization and implementation. Then we present an "anisotropic" improvement of the algorithm which takes into account the (local) contours to compute the second-order derivative. We end with numerical examples arising in biomedical imaging, namely angiography MRI images.

Université d'Orléans, Laboratoire MAPMO - UMR 6628, Fédération Denis-Poisson, BP 6759, F-45067 Orléans Cedex 2, France e-mail: maitine.bergounioux@univ-orleans.fr

1.1 Bounded Variation Spaces of first and second order

Let $\Omega \subset \mathbb{R}^n$ $(n \geq 2)$ be an open bounded set. The space of functions of bounded variation, $BV(\Omega)$ is well known. We refer to [1, 2, 4] for example. We denote by $TV(u)$ the total variation of $u \in BV(\Omega)$:

$$TV(u) = \sup \left\{ \int_\Omega u \operatorname{div} \varphi \, \mathrm{d}x : \varphi \in \mathscr{C}_0^1(\Omega), \|\varphi\|_\infty \leq 1 \right\} \tag{1}$$

Following Demengel [8] and Piffet [10] we may define the space of functions of bounded second-order variation (or hessian bounded) as

$$BV^2(\Omega) := \{ u \in W^{1,1}(\Omega) \mid TV2(u) < +\infty \} .$$

Here the second-order total variation is defined as

$$TV2(u) := \sup \left\{ \int_\Omega \langle \nabla u, \operatorname{div}(\varphi) \rangle_{\mathbb{R}^n} \mid \varphi \in \mathscr{C}_c^2(\Omega, \mathbb{R}^{n \times n}), \|\varphi\|_\infty \leq 1 \right\} \tag{2}$$

where

$$\operatorname{div}(\varphi) = (\operatorname{div}(\varphi_1), \operatorname{div}(\varphi_2), \cdots, \operatorname{div}(\varphi_n)),$$

with

$$\forall i, \ \varphi_i = (\varphi_i^1, \varphi_i^2, \ldots, \varphi_i^n) \in \mathbb{R}^n \ \text{ and } \ \operatorname{div}(\varphi_i) = \sum_{k=1}^{n} \frac{\partial \varphi_i^k}{\partial x_k}.$$

The space $BV^2(\Omega)$ endowed with the norm

$$\|u\|_{BV^2(\Omega)} = \|u\|_{W^{1,1}(\Omega)} + TV2(\Omega)$$

is a Banach space. Moreover, it has been proved in [10] that

$$BV^2(\Omega) = \left\{ u \in W^{1,1}(\Omega) \mid \forall i \in \{1, 2, \cdots, n\} : \frac{\partial u}{\partial x_i} \in BV(\Omega) \right\} .$$

1.2 The abstract second-order model

We recall the variational model described in [4]. We refer to this paper for a precise motivation of this second-order model. Let $\Omega \subset \mathbb{R}^n$ be an open bounded set (smooth enough, for example with Lipschitz boundary). We consider the following functional:

$$\begin{array}{c} F : BV^2(\Omega) \to \mathbb{R}^+ \\ (v) \mapsto F(v) \end{array}$$

$$F(v) = \frac{1}{2} \|u_d - v\|^2_{L^2(\Omega)} + \lambda TV2(v) + \delta \|v\|_{W^{1,1}(\Omega)}$$

where $u_d \in L^2(\Omega)$ and $\lambda, \delta \geq 0$ and we are looking for a solution to the optimization problem:

$$\inf_{v \in BV^2(\Omega)} F(v) \tag{3}$$

It has been proved in [4] that problem (3) has a unique solution for $\lambda > 0$ and $\delta > 0$. However, this result is still true for the discretized problem even with $\delta = 0$. Moreover, in [4] we prove that the existence result still holds true for the infinite dimensional problem if the function v satisfies $\dfrac{\partial v}{\partial n}_{|\partial\Omega} = 0$ and $\Omega = \prod_{i=1}^{n}]a_i, b_i[$ is a square subset of $\mathbb{R}^n$. In what follows, we investigate the finite-dimensional problem, so we assume that $\delta = 0$.

1.3 Discretization of the 3D - problem

In [4] the problem has been discretized in the case of 2D images and numerical tests have been performed. Here we generalize this work to the 3D-case and extend the anisotropic correction of the algorithm of [11] . In the sequel, $n = 3$ and the image size is $N_1 \times N_2 \times N_3$. The generic component of u is $u_{i,j,k}$ and we denote similarly the continuous function (previous section) and the corresponding (discretized) tensor. We denote $X = \mathbb{R}^{N_1 \times N_2 \times N_3}$ endowed with inner product and norm

$$\langle u, v \rangle_X = \sum_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2 \\ 1 \leq k \leq N_3}} u_{i,j,k} v_{i,j,k} \text{ and } \|u\|_X = \sqrt{\sum_{\substack{1 \leq i \leq N_1 \\ 1 \leq j \leq N_2 \\ 1 \leq k \leq N_3}} u_{i,j,k}^2}$$

and set $Y = X \times X \times X$.

(a) We first compute the discrete gradient $\nabla u \in Y$ of the image $u \in X$:

$$(\nabla u_{i,j,k}) = (\nabla u^1_{i,j,k}, \nabla u^2_{i,j,k}, \nabla u^3_{i,j,k})$$

where

$$\nabla u^1_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} & i < N_1 \\ 0 & i = N_1 \end{cases}$$

$$\nabla u^2_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k} & j < N_2 \\ 0 & j = N_2 \end{cases}$$

$$\nabla u^3_{i,j,k} = \begin{cases} u_{i,j,k+1} - u_{i,j,k} & k < N_3 \\ 0 & k = N_3 \end{cases}$$

(b) Discretization of the term $TV2(v)$.

We have

$$\langle \nabla u, \operatorname{div} \phi \rangle = - \left\langle \phi, \nabla^2 u \right\rangle .$$

Then,

$$TV2(v) \simeq \sum_{\substack{1\le i\le N_1\\ 1\le j\le N_2\\ 1\le k\le N_3}} \left\|(Hv)_{i,j,k}\right\|_{\mathbb{R}^9}$$

where

$$\begin{aligned}(Hv)_{i,j,k} =&(Hv^{11}_{i,j,k}, Hv^{12}_{i,j,k}, Hv^{13}_{i,j,k}, Hv^{21,}_{i,j,k}\\ &Hv^{22}_{i,j,k}, Hv^{23}_{i,j,k}, Hv^{31}_{i,j,k}, Hv^{32}_{i,j,k}, Hv^{33}_{i,j,k}).\end{aligned}$$

For every $i = 1,...,N_1$, $j = 1,...,N_2$ and $k = 1,...,N_3$, the computation of Hv gives

$$(Hv)^{11}_{i,j,k} = \begin{cases} v_{i+1,j,k} - v_{i,j,k} + v_{i-1,j,k} & 1 < i < N_1\\ v_{i+1,j,k} - v_{i,j,k} & i = 1\\ v_{i,j,k} - v_{i-1,j,k} & i = N_1 \end{cases}$$

$$(Hv)^{12}_{i,j,k} = \begin{cases} v_{i,j+1,k} - v_{i,j,k} - v_{i-1,j+1,k} + v_{i-1,j,k} & 1 < i \le N_1\\ & 1 \le j < N_2\\ 0 & j = N_2\\ 0 & i = 1 \end{cases}$$

$$(Hv)^{13}_{i,j,k} = \begin{cases} v_{i,j,k+1} - v_{i,j,k} - v_{i-1,j,k+1} + v_{i-1,j,k} & 1 < i \le N_1\\ & 1 \le k < N_3\\ 0 & i = 1\\ 0 & k = N_3 \end{cases}$$

$$(Hv)^{21}_{i,j,k} = \begin{cases} v_{i+1,j,k} - v_{i,j,k} - v_{i+1,j-1,k} + v_{i,j-1,k} & 1 \le i < N_1\\ & 1 < k \le N_3\\ 0 & i = N_1\\ 0 & k = 1 \end{cases}$$

$$(Hv)^{22}_{i,j,k} = \begin{cases} v_{i,j+1,k} - v_{i,j,k} + v_{i,j-1,k} & 1 < j < N_2\\ v_{i,j+1,k} - v_{i,j,k} & j = 1\\ v_{i,j,k} - v_{i,j-1,k} & j = N_2 \end{cases}$$

$$(Hv)^{23}_{i,j,k} = \begin{cases} v_{i,j,k+1} - v_{i,j,k} - v_{i,j-1,k+1} + v_{i,j-1,k} & 1 < j \le N\\ & 1 \le k < N_3\\ 0 & j = 1\\ 0 & k = N_3 \end{cases}$$

$$(Hv)^{31}_{i,j,k} = \begin{cases} v_{i+1,j,k} - v_{i,j,k} - v_{i+1,j,k-1} + v_{i,j,k-1} & 1 < k \le N_3\\ & 1 \le i < N_1\\ 0 & k = 1\\ 0 & i = N_1 \end{cases}$$

$$(Hv)^{32}_{i,j,k} = \begin{cases} v_{i,j+1,k} - v_{i,j,k} - v_{i+,j+1,k-1} + v_{i,j,k-1} & 1 \le j < N\\ & 1 < k \le N_3\\ 0 & j = N_2\\ 0 & k = 1 \end{cases}$$

$$(Hv)^{33}_{i,j,k} = \begin{cases} v_{i,j,k+1} - v_{i,j,k} + v_{i,j,k-1} & 1 < k < N_3 \\ v_{i,j,k+1} - v_{i,j,k} & k = 1 \\ v_{i,j,k} - v_{i,j,k-1} & k = N_3 \end{cases}$$

1.4 Numerical computation of the solution of (3)

Let us consider $H^* : X^9 \to X$ defined as follows (H^* is the adjoint of operator H): for every $p = (p^{11}, p^{12}, p^{13}, p^{21}, p^{22}, p^{23}, p^{31}, p^{32}, p^{33}) \in X^9$,

$$(H^* p)_{i,j,k} = \sigma^{11}_{i,j,k} + \sigma^{12}_{i,j,k} + \sigma^{13}_{i,j,k} + \sigma^{21}_{i,j,k} + \sigma^{22}_{i,j,k} + \sigma^{23}_{i,j,k} + \sigma^{31}_{i,j,k} + \sigma^{32}_{i,j,k} + \sigma^{33}_{i,j,k}$$

where

$$\sigma^{11}_{i,j,k} = \begin{cases} p^{11}_{i+1,j,k} - 2p^{11}_{i,j,k} + p^{11}_{i-1,j,k} & 1 < i < N_1 \\ p^{11}_{i+1,j,k} - p^{11}_{i,j,k} & i = 1 \\ p^{11}_{i-1,j,k} - p^{11}_{i,j,k} & i = N_1 \end{cases}$$

$$\sigma^{22}_{i,j,k} = \begin{cases} p^{22}_{i,j+1,k} - 2p^{22}_{i,j,k} + p^{22}_{i,j-1,k} & 1 < j < N_2 \\ p^{22}_{i,j+1,k} - p^{22}_{i,j,k} & j = 1 \\ p^{22}_{i,j-1,k} - p^{22}_{i,j,k} & j = N_2 \end{cases}$$

$$\sigma^{33}_{i,j,k} = \begin{cases} p^{33}_{i,j,k+1} - 2p^{33}_{i,j,k} + p^{33}_{i,j,k-1} & 1 < k < N_3 \\ p^{33}_{i,j,k+1} - p^{33}_{i,j,k} & k = 1 \\ p^{33}_{i,j,k-1} - p^{33}_{i,j,k} & k = N_3 \end{cases}$$

$$\sigma^{12}_{i,j,k} = \begin{cases} p^{12}_{i+1,j,k} & i = 1, j = 1 \\ -p^{12}_{i+1,j-1,k} & i = 1, j = N_2 \\ p^{12}_{i+1,j,k} - p^{12}_{i+1,j-1,k} & i = 1, 1 < j < N_2 \\ -p^{12}_{i,j,k} & i = N_1, j = 1 \\ p^{12}_{i,j-1,k} & i = N_1, j = N_2 \\ p^{12}_{i,j-1,k} - p^{12}_{i,j,k} & i = N_1, 1 < j < N_2 \\ p^{12}_{i+1,j,k} - p^{12}_{i,j,k} & 1 < i < N_1, j = 1 \\ p^{12}_{i,j-1,k} - p^{12}_{i+1,j-1,k} & 1 < i < N_1, j = N_2 \\ p^{12}_{i,j-1,k} - p^{12}_{i,j,k} - p^{12}_{i+1,j-1,k} + p^{12}_{i+1,j,k} & 1 < i < N_1, 1 < j < N_2 \end{cases}$$

$$\sigma^{13}_{i,j,k} = \begin{cases} p^{13}_{i+1,j,k} & i=1,k=1 \\ -p^{13}_{i+1,j,k-1} & i=1,k=N_3 \\ p^{13}_{i+1,j,k} - p^{13}_{i+1,j,k-1} & i=1,1<j<N_3 \\ -p^{13}_{i,j,k} & i=N_1,k=1 \\ p^{13}_{i,j,k-1} & i=N_1,k=N_3 \\ p^{13}_{i,j,k-1} - p^{13}_{i,j,k} & i=N_1,1<k<N_3 \\ p^{13}_{i+1,j,k} - p^{13}_{i,j,k} & 1<i<N_1,k=1 \\ p^{13}_{i,j,k-1} - p^{13}_{i+1,j,k-1} & 1<i<N_1,k=N_3 \\ p^{13}_{i,j,k-1} - p^{13}_{i,j,k} - p^{13}_{i+1,j,k-1} + p^{13}_{i+1,j,k} & 1<i<N_1,1<k<N_3 \end{cases}$$

$$\sigma^{21}_{i,j,k} = \begin{cases} p^{21}_{i,j+1,k} & j=1,i=1 \\ -p^{21}_{i-1,j+1,k} & j=1,i=N_1 \\ p^{21}_{i,j+1,k} - p^{21}_{i-1,j+1,k} & j=1,1<i<N_1 \\ -p^{21}_{i,j,k} & j=N_2,i=1 \\ p^{21}_{i-1,j,k} & j=N_2,i=N_1 \\ p^{21}_{i-1,j,k} - p^{21}_{i,j,k} & j=N_2,1<i<N_1 \\ p^{21}_{i,j+1,k} - p^{21}_{i,j,k} & 1<j<N_2,i=1 \\ p^{21}_{i-1,j,k} - p^{21}_{i-1,j+1,k} & 1<j<N_2,i=N_1 \\ p^{21}_{i-1,j,k} - p^{21}_{i,j,k} - p^{21}_{i-1,j+1,k} + p^{21}_{i,j+1,k} & 1<j<N_2,1<i<N_1 \end{cases}$$

$$\sigma^{23}_{i,j,k} = \begin{cases} p^{23}_{i,j+1,k} & j=1,k=1 \\ -p^{23}_{i,j+1,k-1} & j=1,k=N_3 \\ p^{23}_{i,j+1,k} - p^{23}_{i,j+1,k-1} & j=1,1<k<N_3 \\ -p^{23}_{i,j,k} & j=N_2,k=1 \\ p^{23}_{i,j,k-1} & j=N_2,k=N_3 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j,k} & j=N_2,1<k<N_3 \\ p^{23}_{i,j+1,k} - p^{23}_{i,j,k} & 1<j<N_2,k=1 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j+1,k-1} & 1<j<N_2,k=N_3 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j,k} - p^{23}_{i,j+1,k-1} + p^{23}_{i,j+1,k} & 1<j<N_2,1<k<N_3 \end{cases}$$

$$\sigma^{31}_{i,j,k}=\begin{cases} p^{31}_{i,j,k+1} & k=1,i=1\\ -p^{31}_{i-1,j,k+1} & k=1,i=N_1\\ p^{31}_{i,j,k+1}-p^{31}_{i-1,j,k+1} & k=1,1<i<N_1\\ -p^{31}_{i,j,k} & k=N_3,i=1\\ p^{31}_{i-1,j,k} & k=N_3,i=N_1\\ p^{31}_{i-1,j,k}-p^{31}_{i,j,k} & k=N_3,1<i<N_1\\ p^{31}_{i,j,k+1}-p^{31}_{i,j,k} & 1<k<N_3,i=1\\ p^{31}_{i-1,j,k}-p^{31}_{i-1,j,k+1} & 1<k<N_3,i=N_1\\ p^{31}_{i-1,j,k}-p^{31}_{i,j,k}-p^{31}_{i-1,j,k+1}+p^{31}_{i,j,k+1} & 1<k<N_3,1<i<N_1 \end{cases}$$

$$\sigma^{32}_{i,j,k}=\begin{cases} p^{32}_{i,j,k+1} & k=1,1=1\\ -p^{32}_{i,j-1,k+1} & k=1,j=N_2\\ p^{32}_{i,j,k+1}-p^{32}_{i,j-1,k+1} & k=1,1<j<N_2\\ -p^{32}_{i,j,k} & k=N_3,j=1\\ p^{32}_{i,j-1,k} & k=N_3,j=N_2\\ p^{32}_{i,j-1,k}-p^{32}_{i,j,k} & k=N_3,1<j<N_2\\ p^{32}_{i,j,k+1}-p^{32}_{i,j,k} & 1<k<N_3,j=1\\ p^{32}_{i,j-1,k}-p^{32}_{i,j-1,k+1} & 1<k<N_3,j=N_2\\ p^{32}_{i,j-1,k}-p^{32}_{i,j,k}-p^{32}_{i,j-1,k+1}+p^{32}_{i,j,k+1} & 1<k<N_3,1<j<N_2 \end{cases}$$

It is straightforward to prove that

Theorem 1. *The solution to problem (3) verifies:*

$$v = ud - P_{\lambda K}(u_d)$$

where $P_{\lambda K}$ is the orthogonal projector operator on λK and

$$K := \{H^* p \mid p \in X^9,\ \|p_{i,j,k}\|_{\mathbb{R}^9} \leq 1,\ 1 \leq i \leq N_1, 1 \leq j \leq N, 1 \leq k \leq N_3\}.$$

Proof. It is quite similar to the 2D-case proof. We refer to [4].

To compute $P_{\lambda K}(u_d)$ we have to solve the following problem:

$$\begin{cases} \min \|\lambda H^* p - u_d\|_X^2 \\ p \in X^9 \\ \|p_{i,j,k}\|_{\mathbb{R}^9}^2 \leq 1, 1 \leq i \leq N_1, 1 \leq j \leq N_2, 1 \leq k \leq N_3 \end{cases}$$

Following [6] and [4] we use the following algorithm to compute $P_{\lambda K}(u_d)$

Algorithm

Choose $\tau > 0$

1. Let $p^0 = 0, n = 0$.
2. Suppose p^n is known, we compute p^{n+1} as follows:

$$p^n_{i,j,k} = p^{n+1}_{i,j,k} + \tau \left[(H \left[H^* p - \frac{u_d}{\lambda} \right])_{i,j,k} + \left\| (H \left[H^* p^n - \frac{u_d}{\lambda} \right])_{i,j,k} \right\|_{R^9} p^{n+1}_{i,j,k} \right]$$

which implies:

$$p^{n+1}_{i,j,k} = \frac{p^n_{i,j,k} - \tau (H \left[H^* p^n - \frac{u_d}{\lambda} \right])_{i,j,k}}{1 + \tau \left\| (H \left[H^* p^n - \frac{u_d}{\lambda} \right])_{i,j,k} \right\|_{\mathbb{R}^9}}$$

Theorem 2. *Let $\tau \leq 1/8^3$, then $\lambda (H^* p^n)_n$ converges to $P_{\lambda K_2}(u_d)$ as $n \to \infty$.*

Proof. Once again the proof is quite technical but similar to the 2D-case proof ([4]).

2 Introducing anisotropy

L. Piffet [8, 10, 11] has observed (in the 2D-case) that cancelling one or more coefficients of the Hessian matrix permits to get rid of the contours along the corresponding direction.

(a) Original image (Barbara)

(b) Texture part without anisotropic strategy

(c) Texture part without horizontal and vertical contours

Fig. 1 Effects of anisotropic improvement strategy

We give a 2D-example in Figure 1 : here the coefficients $(Hv)^{1,1}$ and $(Hv)^{2,2} = 0$ have been globally set to 0. We can see that horizontal and vertical contours are not involved in the texture part any longer. This method has been improved since there were two major inconveniences :

- First, the same transform is performed at every pixel, so that the image is globally treated. All the vertical and horizontal lines are removed;
- Second, the transform is depended on the chosen (fixed) cartesian axis and it is not possible to remove contours that are not horizontal, vertical or diagonal.

Therefore, the Hessian matrix is now locally computed at every pixel. First , a rotation is performed so that the gradient direction is the new y-axis (or x-axis). The

corresponding Hessian matrix is computed and suitable coefficients are canceled. Then the inverse rotation is performed. For more details on can refer to [10, 11].

We compute the (local) 3D- Hessian matrix at a voxel (i, j, k) using this technique. We have to perform two rotations r_α and r_β to compute an modified hessian matrix H'. More precisely, we perform a change of variables (with the rotations) to compute the Hessian matrix and the adjoint matrix as in the previous section: the local axis (with the gradient vector as z-axis) are considered instead of the original fixed cartesian axis. Then, we may cancel the Hessian matrix terms corresponding to the gradient direction (for example), to get rid of the corresponding contour (if it is significant) in the extracted texture. Finally we go back to the original axis with the inverse rotations. Let us detail the process :

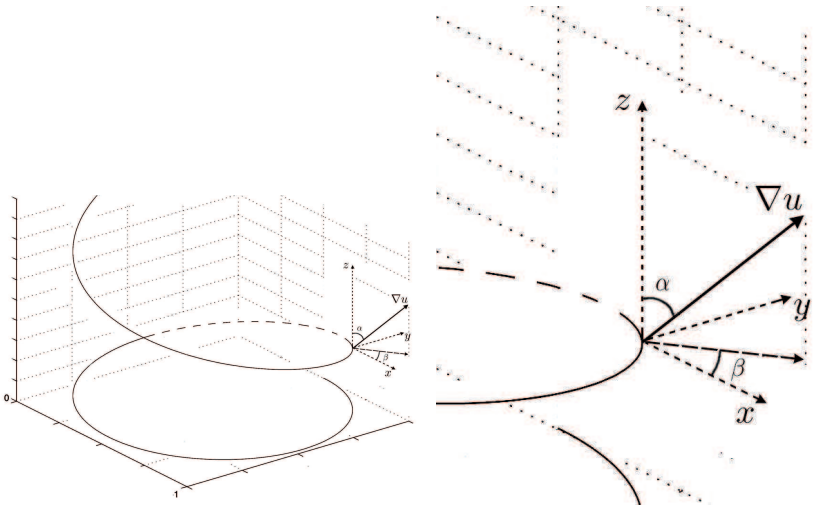


Fig. 2 Definition of local axis and angles α and β

The angles α and β are defined at point $X_o = (x_o, y_o, z_o)$ as follows : α is the (azimuthal) angle between the gradient $\nabla u(x_o, y_o, z_o)$ and the z-axis . β is the angle between the orthogonal projection of

$$\nabla u(x_o, y_o, z_o) := \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} (x_o, y_o, z_o)$$

(on the xOy plane) and the x -axis. Note that we can perform this transformation with axis Ox or Oy instead of Oz . Let us define the two rotations : r_α and r_β which matrices are :

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \text{ and } R_\beta = \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} ,$$

with

$$\alpha = \text{atan}\left(\frac{u_z}{\sqrt{u_x^2+u_y^2}}\right)(X_o), \quad \beta = \text{atan}\left(\frac{u_y}{u_x}\right)(X_o)\ .$$

The change of variables from the fixed basis to the local one is given par

$$\tilde{X} = R_\beta R_\alpha X, \ \text{ with } X=(x,y,z)\in\mathbb{R}^3\ .$$

Moreover

$$X = (R_\beta R_\alpha)^{-1}\tilde{X} = R_\alpha^{-1}R_\beta^{-1}\tilde{X} = R_{-\alpha}R_{-\beta}\tilde{X}\ .$$

In the sequel, we set $\tilde{u}(\tilde{X}) := u(X)$ and $R_{\alpha,\beta} \stackrel{def}{:=} R_{-\alpha}R_{-\beta}$ and we compute the first and second order derivative of $\tilde{u}$:

$$\nabla\tilde{u} = \begin{pmatrix} \frac{\partial\tilde{u}}{\partial\tilde{x}} \\ \frac{\partial\tilde{u}}{\partial\tilde{y}} \\ \frac{\partial\tilde{u}}{\partial\tilde{z}} \end{pmatrix} \text{ and } \tilde{H} := \begin{pmatrix} \frac{\partial^2\tilde{u}}{\partial\tilde{x}^2} & \frac{\partial^2\tilde{u}}{\partial\tilde{x}\partial\tilde{y}} & \frac{\partial^2\tilde{u}}{\partial\tilde{x}\partial\tilde{z}} \\ \frac{\partial^2\tilde{u}}{\partial\tilde{x}\partial\tilde{y}} & \frac{\partial^2\tilde{u}}{\partial\tilde{y}^2} & \frac{\partial^2\tilde{u}}{\partial\tilde{y}\partial\tilde{z}} \\ \frac{\partial^2\tilde{u}}{\partial\tilde{x}\partial\tilde{z}} & \frac{\partial^2\tilde{u}}{\partial\tilde{y}\partial\tilde{z}} & \frac{\partial^2\tilde{u}}{\partial\tilde{z}^2} \end{pmatrix}\ .$$

A short computation gives

$$\frac{\partial\tilde{u}}{\partial\tilde{x}} = \frac{\partial u}{\partial x}\frac{\partial\tilde{x}}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial\tilde{y}}{\partial x} + \frac{\partial u}{\partial z}\frac{\partial\tilde{z}}{\partial x} = \nabla u\cdot\frac{\partial\tilde{X}}{\partial x} = \nabla u\cdot R(:,1)\ ,$$

where $\cdot$ denotes the $\mathbb{R}^3$ scalar product and $R(:,1)$ is the first column of R. Finally, we get

$$\nabla\tilde{u} = R_{\alpha,\beta}\nabla u\ . \tag{4}$$

Now we compute $\tilde{H}$; we set $\tilde{v} = \dfrac{\partial\tilde{u}}{\partial\tilde{x}}$ and estimate $\nabla\tilde{v}$ as above : this will be the first column of $\tilde{H}$.

$$\nabla\tilde{v} = R_{\alpha,\beta}\nabla v = R_{\alpha,\beta}\begin{pmatrix} \frac{\partial^2 u}{\partial x^2} \\ \frac{\partial^2 u}{\partial y\partial x} \\ \frac{\partial^2 u}{\partial z\partial x} \end{pmatrix}\ .$$

Finally

$$\tilde{H} = R_{\alpha,\beta}H\ . \tag{5}$$

As already mentioned, the idea is to cancel some terms of the Hessian matrix to get rid of (or to keep) the contours. However, without performing the rotations, there would be only few possible directions, for example vertical, horizontal and diagonal

in the 2D-case so that many contours are not considered. Performing the change of variables allows to identify the gradient direction (that is the contour direction if the gradient is large enough) with the z-axis and then cancel corresponding terms of the matrix $\tilde{H}$. Of course, we have to get back to the original situation. Let us denote by $\mathscr{L}$ the (linear) transformation that assigns 0 to some coefficients of $\tilde{H}$ (this is a projection). The whole process is described by

$$H \to \tilde{H} = R_{-\alpha}R_{-\beta}H \to \mathscr{L}(\tilde{H}) := \tilde{H}' \to [R_{\alpha,\beta}]^{-1}\mathscr{L}(\tilde{H}) = R_\beta R_\alpha \mathscr{L}(\tilde{H}) \ ,$$

that is

$$H \to \left[R_\beta R_\alpha \mathscr{L} R_{-\alpha}R_{-\beta}\right] H \ . \tag{6}$$

So, algorithm p.48 is modified as follows

Algorithm

Choose $\tau > 0, \square > 0$ and compute ∇u. Use a threshold process to identify the contours ($\|\nabla u\| \geq \square$) . Set $I_\square$ the set of voxels corresponding to these"significant contours".

1. Let $p^0 = 0, n = 0$.
   For voxels in $I_\square$, modify H with the following rule

   $$H \to \tilde{H} = R_{-\alpha}R_{-\beta}H \to \mathscr{L}(\tilde{H}) = [\mathscr{L}R_{-\alpha}R_{-\beta}]H := H'$$

   and compute $(H')^*$
2. Same as before p.48 with H' instead of H.

3 Numerical examples

Numerical experimentation has been done in the context of biomedical imaging. We consider a stack of 50 MRI images of the vessel network of brain mice.[1] The challenge is to identify the network to get structural informations. Using 2D segmentation and interpolation methods is not possible, since the slices are not exploitable (see Figure 3.)

[1] We thank J.C. Belœil, S. Même and F. Szeremeta, from CBM Laboratory in Orléans, for the use of these images, `http://cbm.cnrs-orleans.fr/`.

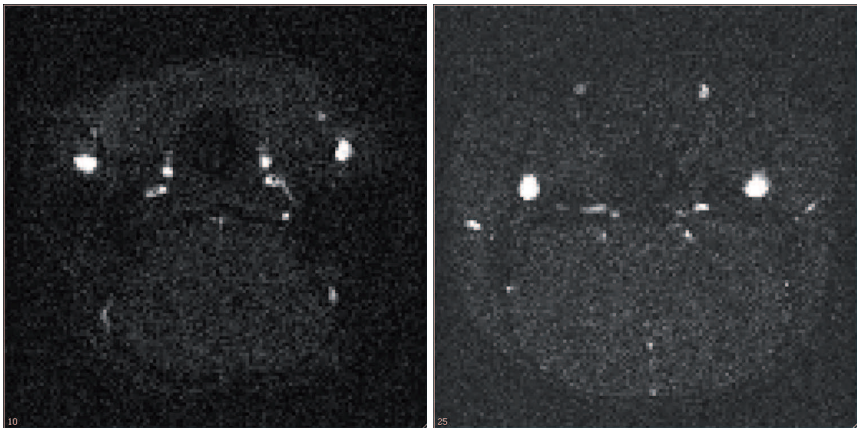
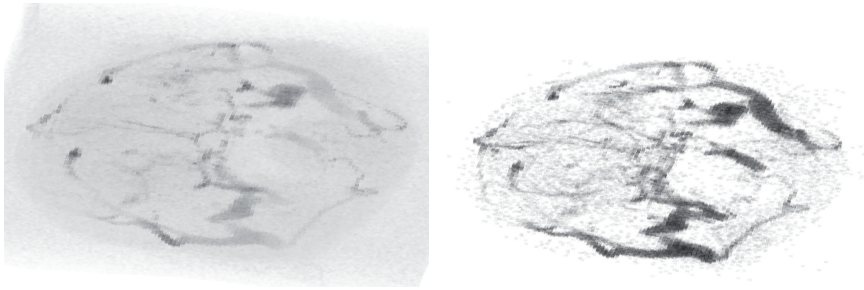


Fig. 3 2D slices example (slices 10 and 25)

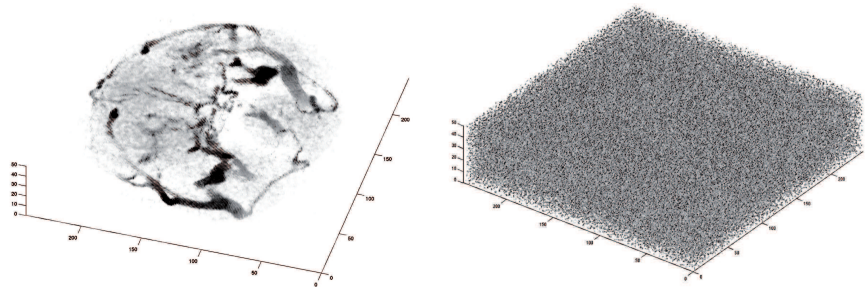
Therefore we have to deal with the complete 3D information. We consider that noise and very small vessels effect is texture. Extracting texture gives the remainder part, the so-called "cartoon" (smooth part). We expect that the contours are kept in the cartoon part which in the cleaned image in some sense. Then classical segmentation methods (as threshold for example) can be used. The following results have been obtained without any anisotropic strategy. Indeed, computational time is large and we still have to improve the speed of algorithm. However, we present a comparison between the two methods with and without anisotropy strategy. The results show that the anisotropy technique is quite efficient and we have good hope to keep the whole contour information contour in the cartoon part.

We have tested many values for λ and the maximum number of iterations. We present some results to show the influence of λ (images have been contrasted). We shall speed up the method in the future using (for example) Nesterov algorithms as in [12] .



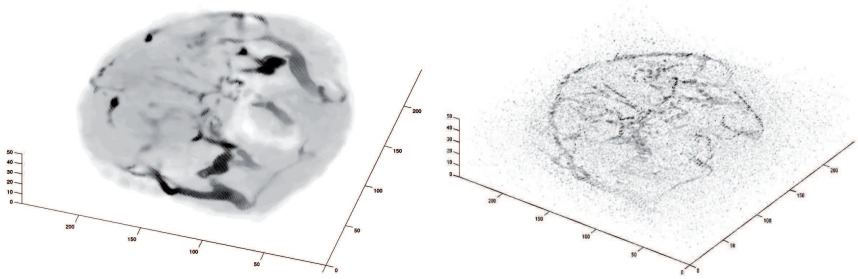
(a) Original Image (b) Image with threshold at grey value = 210

Fig. 4 3D angiography image



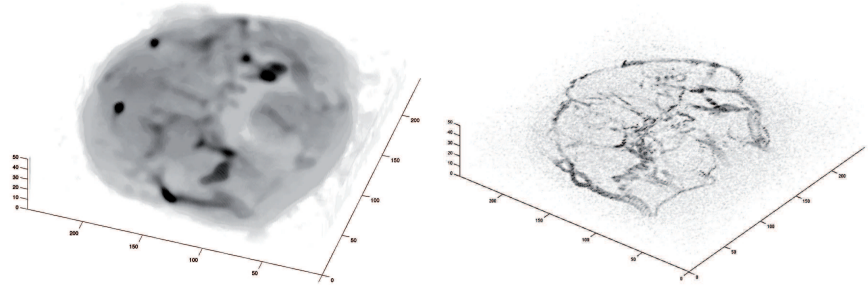
(a) Cartoon (b) Texture

Fig. 5 No anisotropy strategy : $\lambda = 1$ and 5 000 iterations - The choice of small λ allows to denoising the image quite efficiently : here the texture is the noise and the cartoon the denoised image



(a) Cartoon (b) Texture

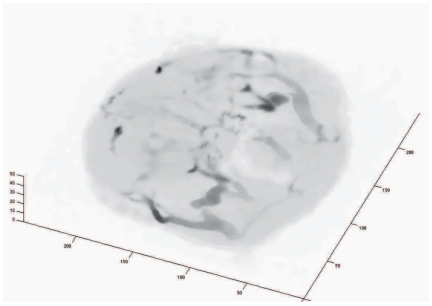
Fig. 6 No anisotropy strategy :$\lambda = 10$ and 5 000 iterations



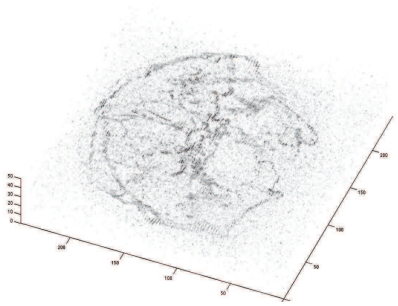
(a) Cartoon (b) Texture

Fig. 7 No anisotropy strategy : $\lambda = 50$ and 10 000 iterations -The contours and the vessel network are recovered in the texture.

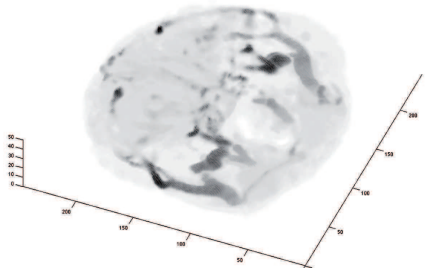
We have tested the algorithm with and without anisotropy strategy. We give below results for $\lambda = 10$ and 5000 iterations. As the 3D cartoon and texture pictures are not easy to compare we give pictures of the difference as well.



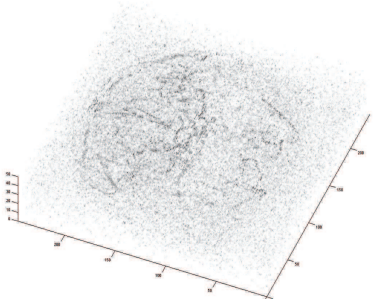
(a) Cartoon without anisotropy strategy



(b) Texture without anisotropy strategy : contours are involved in the texture

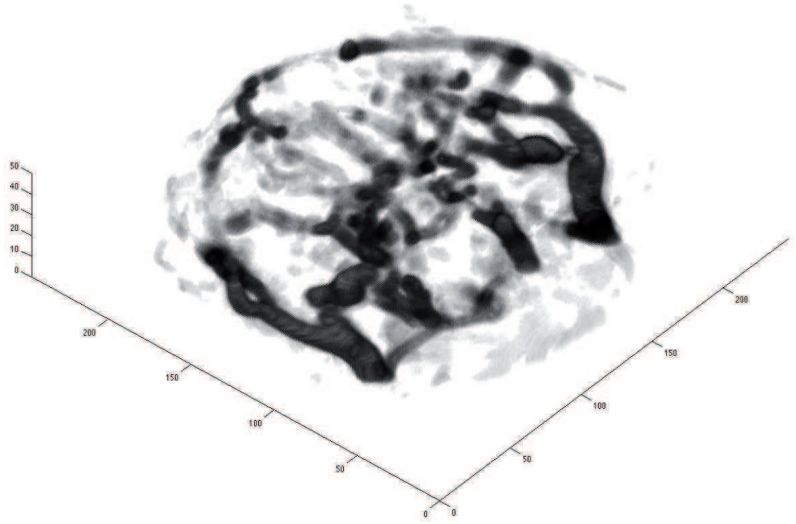


(c) Cartoon with anisotropy strategy

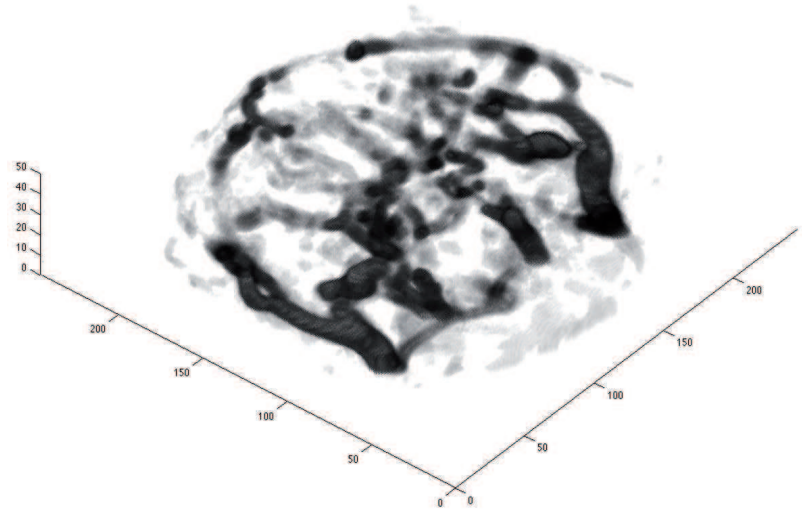


(d) Texture with anisotropy strategy : contours are not involved in the texture any longer

Fig. 8 Comparison between the two strategies for $\lambda = 10$ and 5 000 iterations



(a) Difference between cartoons



(b) Difference between textures

Fig. 9 Absolute value of the difference for $\lambda = 10$ and 5 000 iterations . The vessel-network which is alternatively included in the cartoon (when no anisotropy strategy is performed) or in the texture in the other case, so that the respective differences of cartoons and textures are the same and give the vessel network.

References

1. Ambrosio L., Fusco N. , Pallara D. , Functions of bounded variation and free discontinuity problems, Oxford mathematical monographs, Oxford University Press, 2000.
2. Attouch H., Buttazzo G., Michaille G., Variational analysis in Sobolev and BV spaces : applications to PDEs and optimization, MPS-SIAM series on optimization, 2006.
3. Bergounioux M., Piffet L. , A second-order model for image denoising , Set-Valued Analysis and Variational Analysis, Vol. 18, 3-4, pp. 277-306, 2010.
4. Bergounioux M. , On Poincaré-Wirtinger inequalities in spaces of functions of bounded variation, preprint 2010, `http://hal.archives-ouvertes.fr/hal-00515451/fr/`
5. Bredies K., Kunisch K. , Pock T., Total Generalized Variation, preprint 2009.
6. Chambolle A., An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision, **20**, 89–97, 2004.
7. Demengel F., Fonctions à hessien borné, Annales de l'institut Fourier, **34**, no 2, 155–190, 1984.
8. Echegut, R., Piffet, L., A variational model for image texture identification, Recent Advances in Optimization and its Applications in Engineering, Diehl, M.; Glineur, F.; Jarlebring, E.; Michiels, W. (Eds.), Springer, 2010.
9. L.C. Evans, R. Gariepy, Measure theory and fine properties of functions, CRC Press, 1992.
10. L. Piffet, Modèles variationnels du second ordre pour l'extraction de textures 2D, PhD Thesis, Orléans, 2010.
11. L. Piffet, this book, 2011.
12. Weiss P., Blanc-Fraud L., Aubert, G., Efficient schemes for total variation minimization under constraints in image processing. SIAM journal on Scientific Computing, **31**, no 3, 2047–2080, 2009.

Appendix G

Publication 2

Informations:

Tran M.P., Péteri R., Bergounioux M.,

Denoising 3D medical images using a second-order variational model and wavelet shrinkage,

Image Analysis and Recognition, Lecture Notes in Computer Science,

Volume 7325, Pages 138-145,

Springer Berlin Heidelberg, 2012.

Denoising 3D medical images using a second order variational model and wavelet shrinkage

Minh-Phuong Tran[1], Renaud Péteri[2], and Maitine Bergounioux[1]

[1] Université d'Orléans, Laboratoire MAPMO
UMR 6628, Fédération Denis-Poisson
BP 6759, F-45067 Orléans Cedex 2, France
minh-phuong.tran@etu.univ-orleans.fr, Maitine.Bergounioux@univ-orleans.fr
[2] Université de La Rochelle,
Laboratoire Mathématiques, Image et Applications
EA 3165
F-17000 La Rochelle, France
renaud.peteri@univ-lr.fr

Abstract. The aim of this paper is to construct a model which decomposes a 3D image into two components: the first one containing the geometrical structure of the image, the second one containing the noise. The proposed method is based on a second order variational model and an undecimated wavelet thresholding operator. The numerical implementation is described, and some experiments for denoising a 3D MRI image are successfully performed. Future prospects are finally exposed.

Keywords: Image Decomposition, Image Denoising, Undecimated wavelet Shrinkage, Second order variational model, 3D medical image

1 Introduction

Medical images obtained from MRI (Magnetic-Resonance-Imaging) are now a very common tool for diagnosing human diseases. These images are often affected by random noise arising during the acquisition process. Moreover, medical images constituted of low-contrast objects are a major challenge for biomedical researchers. The noise highly affects the visual interpretation of medical images, but also most of the segmentation or clustering algorithms. Therefore, denoising medical images is an important pre-step for medical image analysis.

Image denoising is one of the classical problems in image processing, and has been studied for several years due to its important role in various applications. Its goal is to remove noise and/or spurious details from a given corrupted image while maintaining its important features. Many denoising methods have been developed, such as methods based on variational methods, rank filters, frequency domain filters or sparse representations (curvelets, beamlets,...).

The general idea behind variational denoising methods is to considered an observed image f as a corrupted version of a noiseless image u. In denoising models, image u is then the solution of an inverse problem. One of the most successful variational algorithms is the Rudin-Osher-Fatemi (ROF) model ([2, 4, 5]) which uses Total-Variation regularization. The observed image to recover/denoise f is

split into two components u and v, giving $f = u + v$, where u is the cartoon part (the smooth component), the remaining term $v := f - u$ being the noise. The functional energy F on bounded variation space is:

$$F(u) = \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV(u), u \in BV(\Omega) \tag{1}$$

where $TV(u)$ represents the total variation of $u \in BV(\Omega)$ [10], and $\lambda \geq 0$ is a regularization parameter. Solving this problem leads to the minimisation of the following expression:

$$\inf_{u \in BV(\Omega)} F(u) \tag{$\mathcal{P}_{ROF}$}$$

It has been shown that this problem has a unique solution in $BV(\Omega)$ ([9, 1]). However, the use of the BV-norm in the ROF model favours piecewise constant solutions, causing unsatisfying *'staircasing effects'* [6]. This variational model has been improved by using different functional spaces. In [9] it has been proposed to use the second order functional space of bounded variation - the BV^2 space. This model leads to the minimisation of the following expression:

$$\inf_{u \in BV^2(\Omega)} F_2(u) \tag{$\mathcal{P}_{ROF2}$}$$

where

$$F_2(u) = \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV^2(u), u \in BV^2(\Omega) \tag{2}$$

In the following section, we generalize the model ROF to the new functional space BV^2 for 3D signals using second order total variation TV^2 [10]. The problem is considered in the BV^2 discrete space.

2 Three-dimensional ROF2 model

2.1 Functional framework

Let $\Omega \subset \mathbb{R}^3$ be an open bounded set, we consider the finite-dimensional problem where function F_{ROF2}:

$$\begin{aligned} F_{ROF2} : BV^2(\Omega) &\to \mathbb{R}^+ \\ u &\mapsto F_{ROF2}(u) \end{aligned}$$

is defined by:

$$F_{ROF2}(u) = \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV^2(u)$$

Solving the second order model (ROF2) leads to the minimisation of the following expression:

$$\inf_{u \in BV^2(\Omega)} \frac{1}{2}\|f-u\|^2_{L^2(\Omega)} + \lambda TV^2(u) \tag{$\mathcal{P}_{ROF2}$}$$

Theorem 1. [9] *If $\lambda > 0$, it has been shown that the problem has an unique solution.*

2.2 Discretization of the ROF2 model

In the sequel, we denote by X the Euclidean space $\mathbb{R}^{N_1 \times N_2 \times N_3}$ and $Y = X \times X \times X$. The space X is endowed with the inner product:

$$\langle u, v \rangle_X = \sum_{\substack{1 \le i \le N_1 \\ 1 \le j \le N_2 \\ 1 \le k \le N_3}} u_{i,j,k} v_{i,j,k}$$

In the case of the ROF2 model, the second order total variation term $TV^2(u)$ can be discretized to $J(u)$ (more details can be found in [9, 10]). The discretization of the ROF2 model ($\mathcal{P}_{ROF2}$) can be then defined as:

$$\inf_{u \in X} J(u) + \frac{1}{2\lambda} \|f - u\|_X^2 \qquad \text{(d-}\mathcal{P}_{ROF2}\text{)}$$

where $J(u)$ stands for the discrete TV^2. The following theorem comes from the convex duality theory [7], and gives the approximated solution:

Theorem 2. *The solution to problem ROF2 verifies:*

$$u = f - P_{\lambda K}(f)$$

where $P_{\lambda K}$ is the orthogonal projector operator on λK, and

$$K := \{H^* p \mid p \in X^9,\ \|p_{i,j,k}\|_{\mathbb{R}^9} \le 1;\ 1 \le i, j, k \le N_1, N_2, N_3\}.$$

H is the Hessian operator and H^* its adjoint. We refer to [9, 2] for the proof of this theorem. Moreover, in order to approximate the projection term $P_{\lambda K}(f)$ of theorem 2, the following problem has to be solved [2] :

$$\begin{cases} \min \|\lambda H^* p - f\|_X^2 \\ p \in X^9 \\ \|p_{i,j,k}\|_{\mathbb{R}^9}^2 \le 1;\ 1 \le i, j, k \le N_1, N_2, N_3 \end{cases} \tag{3}$$

This problem can be solved by a fixed point method with an iterative scheme on the solution p: $p^0 = 0$ and

$$p_{i,j,k}^{n+1} = \frac{p_{i,j,k}^n - \tau \left(H \left[H^* p^n - \frac{f}{\lambda} \right] \right)_{i,j,k}}{1 + \tau \left\| \left(H \left[H^* p^n - \frac{f}{\lambda} \right] \right)_{i,j,k} \right\|_{\mathbb{R}^9}} \tag{4}$$

The discretization of the three-dimensional Hessian operator H and its adjoint operator H^* as well as a sufficient condition ensuring the convergence of the algorithm can be found in [10].

Theorem 3. [10] *Let $\tau \le 1/12^2$, then $\lambda(H^* p^n)_n$ converges to $P_{\lambda K}(f)$ as $n \to \infty$.*

3 3D Image Decomposition Model using Undecimated Wavelet Shrinkage

In this section, a decomposition model based on the second order variational model ROF2 is presented. Following the work of [15, 8], an undecimated wavelet transform (the 'a trous' algorithm) is introduced in order to better separate geometry from noise during the iteration process.

3D decomposition model. The proposed method aims at dividing a 3D image f into two components: the first component $u \in BV^2$ represents the geometrical information (smooth part) while the second component v contains the noise, with $f = u + v$. This decomposition model has been proposed in [8] and is computed by minimizing a convex functional which depends on two variables (u, v) as following:

$$\inf_{(u,v)\in X^2} J(u) + B^*(v/\delta) + \frac{1}{2\lambda}\|f - u - v\|_X^2 \qquad (\mathcal{P})$$

where $B^*(v/\delta)$ is the Legendre-Fenchel transform of B of the noise component v, [8]. Furthermore, let us denote $\delta B_E = \{z/\|z\|_E \leq \delta\}$. In order to solve the problem $(\mathcal{P})$, one considers to solve the two following problems:

1. v being fixed, we find u as solution of problem:

$$\inf_{u\in X} J(u) + \frac{1}{2\lambda}\|f - u - v\|_X^2 \qquad (5)$$

2. u being then fixed, we search for v as the solution of:

$$\inf_{v\in \delta B_E} \|f - u - v\|_X^2 \qquad (6)$$

The solution of problem (5) is given by $u^* = f - v - P_{\lambda K}(f - v)$.
Solution of (6) is obtained using the universal threshold T during the iteration process [8] on an undecimated wavelet transform, the 'à trous' algorithm. Solution can be written $v^* = f - u - UWT(f - u, T)$, where UWT denotes the undecimated wavelet thresholding operator that is detailed in the next section.

The " à trous" algorithm . The 'à trous' algorithm [3] is a fast dyadic wavelet transform and is implemented with filter banks. It is similar to a fast biorthogonal wavelet transform but without subsampling. In our 3D implementation, the scaling and wavelet functions ϕ and ψ are a cubic B-splines that enable a nearly isotropic analysis of the 3D image, and filters are separable 1D filters.
For any resolution level $j \geq 0$, the approximation a_j and the details d_j (wavelet coefficients) are:

$$a_j[n, m, l] =< f(x, y, z), \phi_{2^j}(x - n)\phi_{2^j}(y - m)\phi_{2^j}(z - l) > \qquad (7)$$
$$d_j[n, m, l] =< f(x, y, z), \psi_{2^j}(x - n)\psi_{2^j}(y - m)\psi_{2^j}(z - l) > \qquad (8)$$

and discrete image values are assimilated to $a_0[n, m, l]$.
A filter $x[n]$ is dilated to make the filter $x_j[n]$ by inserting 2^{j-1} zeros ('trous') between each sample. Let us denote $\bar{x}_j[n] = x_j[-n]$ and $\delta[n]$ the discrete Dirac. $\bar{h}$ is a low-pass filter associated with the scaling function ϕ and $\bar{g}$ is a high-pass filter associated with the mother wavelet ψ.
The "à trous" algorithm then enables to compute the fast dyadic wavelet transform in the following way:

$$a_{j+1}[n, m, l] = (\bar{h}_j \bar{h}_j \bar{h}_j * a_j)[n, m, l], \tag{9}$$
$$d_{j+1}[n, m, l] = ([\bar{h}_j \bar{h}_j \bar{h}_j - \delta\delta\delta] * a_j)[n, m, l] \tag{10}$$

where $\bar{h}_j \bar{h}_j \bar{h}$ and $\delta\delta\delta$ are 3D filters obtained from $\bar{h}$ and δ by tensor products.

As there is no downsampling of the original image, all the approximation and wavelet images have the same size. The undecimated wavelet thresholding operator UWT used for computing v^* perform the 3D 'à trous' decomposition of the image, applies the universal threshold T on each 3D wavelet images and reconstructed the 3D thresholded image by summing the details and the last approximation.

Proposed Algorithm. Consequently, our decomposition model is solved by the following iterative algorithm:

1. Initialization: $u_0 = v_0 = 0$,
2. Iterations on n:
$$u_{n+1} = f - v_n - P_{\lambda K}(f - v_n) \tag{11}$$
$$v_{n+1} = f - u_{n+1} - UWT(f - u_{n+1}, T) \tag{12}$$
3. Stopping test: if the following condition is fullfilled:
$$\max\left(|u_{n+1} - u_n|, |v_{n+1} - v_n|\right) \leq \epsilon \tag{13}$$

4 Application to 3D medical image denoising

The proposed method has been applied on the MRI of a trisomic mouse (Fig. 1). The mouse brain volume is the stack of 104 MRI images. This is a difficult case because the contrast between different objects in the brain is low, and there is moreover some acquisition noise (see top image of figure 3).

Our 3D image decomposition method has been applied to this data, for different values of regularizing parameter λ (see figure 2). Since in practice there is no denoised volume to compare to, tuning of parameter λ often relies on visual inspection. The stopping criterion has been set to a maximal number of iterations which can be chosen arbitrary large.

One can observe that the algorithm is able to separate the initial MRI image into a component u that contains the regularized (denoised) image, and a component v that contains mostly noise with some texture and contours information.

Fig. 1. Original 3D MRI of a Mouse Brain.

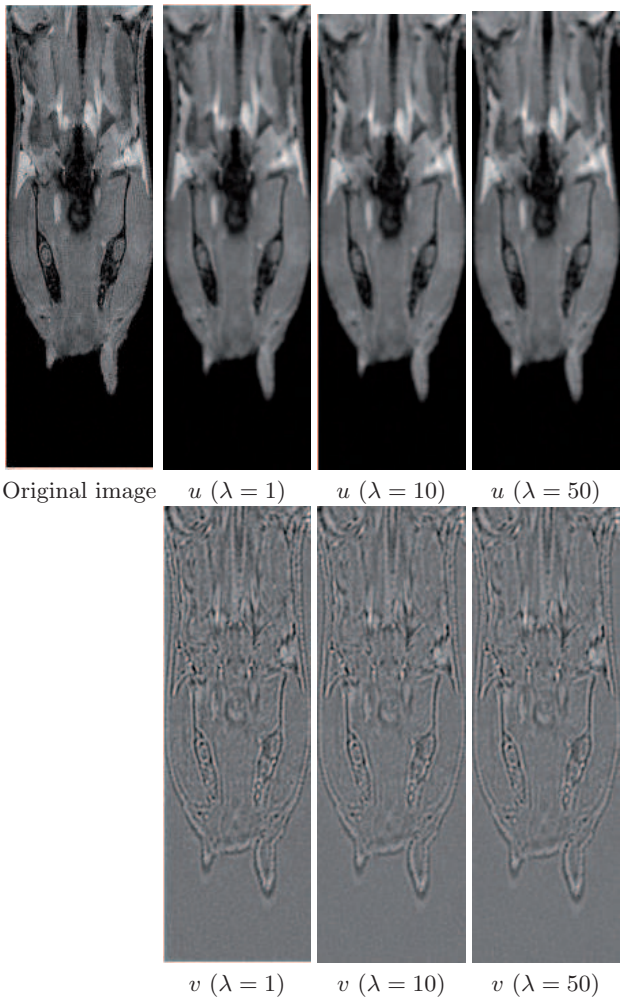


Fig. 2. Comparison of the $u + v$ decomposition for different value of regulizer λ.

The good ability to denoise the initial 3D image is confirmed on figure 3, which shows one slice on the 3D image represented as a 2D surface, its regularized component u and its noise component v ($\lambda = 10$). In figure 3, component v can

be viewed as a very highly oscillating function. In addition, one can notice in the denoise part that edges are not oversmoothed. Moreover, its behaviour is quite stable with respect to λ (for a large value of $\lambda = 100$, geometric details appear in the noise component v).

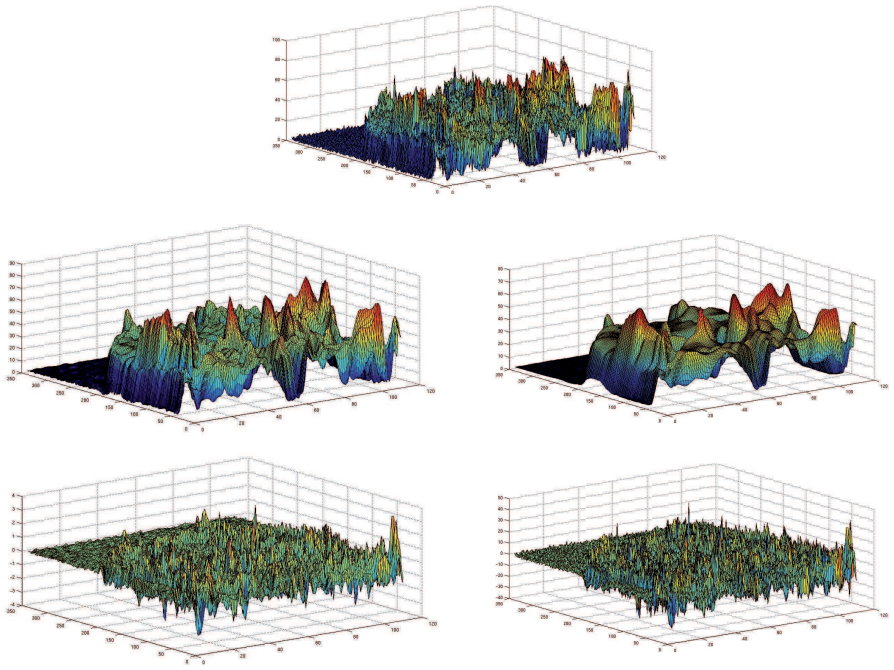


Fig. 3. Surface representation of one slice of the original 3D volume (top). The u component (middle row) and its v component (bottom row). The proposed decomposition model with undecimated wavelet shrinkage (left column) and a comparison with no wavelet shrinkage (right column).

A comparison using the same decomposition model without undecimated wavelet shrinkage has also been performed (using the same value for $\lambda = 10$). It can be noticed on Fig. 3 (right column), that the u component is a bit oversmoothed and thus region borders are blurred.

5 Conclusion

This article describes a new 3D decomposition method which separates a 3D image into two components: the first one containing the geometrical structure of the image, the second one containing the noise. The proposed method is based on a second order variational model and an undecimated wavelet thresholding

operator. The numerical implementation is described, and an experiment for denoising a 3D MRI image of a mouse brain has been successfully performed. In future works, we shall focus on extending this model to a three component model $f = u + v + w$, which could discriminate between geometrical structures (u), textures (v) and noise (w). Application of this method to video is also under consideration.

References

1. Bergounioux M. On Poincare-Wirtinger inequalities in spaces of functions of bounded variation. *[hal-00515451] version 2, 10 June 2011.*
2. Chambolle A. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision Volume 20* (2004), 89–97.
3. Holschneider, M., R. Kronland-Martinet, J. Morlet, and P. Tchamitchian (1989). A real time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets, Time-Frequency Methods and Phase Space*, pp. pages 286–297. Springer-Verlag. Berlin, Allemagne.
4. Chambolle A., Lions P.L. Image recovery via total variation minimization and related problems Numerische Mathematik. *Journal of Mathematical Imaging and Vision* 167-188, volume77, 1997.
5. Chan T. Esedoglu S., Park F., Yip A. *Recent Developments in total Variation Image Restoration.* CAM Report 05-01, Department of Mathematics, UCLA 2004.
6. Louchet C. Variational and Bayesian models for image denoising: from total variation towards non-local means. *Universite Paris Descartes, Ecole Doctoranle Mathematiques Paris-Centre, December 10,2008.*
7. Ekeland I., Remam R. Analyse convex et problemes variationnels. *Etudes Mathematiques. Dunod, 1974.*
8. Jean-Francois Aujol and Antonin Chambolle Dual norms and image decomposition models. *IJCV, volume 63, number 1, pages 85-104, June 2005.*
9. Piffet L. Décomposition d'image par modèles variationnels - Débruitage et extraction de texture. Université d'Orléans, Pôle Universités Centre Val de Loire, Novembre 23,2010.
10. Bergounioux M., Tran M.P. A second order model for 3D texture extraction. Mathematical Image Processing. Springer proceedings in Mathematics 5. Universite d'Orleans, 2011.
11. Jean-Luc Starck, Emmanuel J. Candes and David L. Donoho The Curvelet transform for Image Denoising. *IEEE Transactions on image processing, Vol.11, No.6, June 2002*
12. Mallat S. A wavelet tour of signal processing. *Academic Press Inc., 1998.*
13. Yinpeng Jin, Elsa Angelini, Andrew Laine Wavelets in medical image processing: denoising, sementation and registration. *Department of Biomedical Engineering, Columbia University, New York, USA*
14. Yves Meyer Oscillating patterns in image processing and in some nonlinear evolution equations. *The 15th Dean Jacquelines B. Lewis Memorial Lectures, March 2001.*
15. G. Steidl, J. Weickert, T. Brox, P. Mrzek, and M. Welk, *On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides*, Tech. Rep. 26, Department of Mathematics, University of Bremen, Germany, 2003.

Bibliography

[1] Aujol J.F.; Aubert G.; Blanc-Féraud L.; Chambolle A. Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vision*, 22(1):71–88, 2005. (Cited on pages 33 and 36.)

[2] Aujol J.F.; Chambolle A. Dual norms and image decomposition models. *International Journal of Computer Vision*, 63(1):85–104, June 2005. (Cited on pages 22, 23, 114, 115 and 163.)

[3] Bultheel A. Wavelets with applications in signal and image processing. October 26, 2006. (Cited on pages 9 and 47.)

[4] Chambolle A. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1-2):89–97, 2004. Special issue on mathematics and image analysis. (Cited on pages 7, 9, 18, 22, 27, 33, 34 and 36.)

[5] Chan T.; Ezsedoglu S. ;Park F.; Yip A. Recent developments in total variation image restoration. *CAM Report 05-01, Department of Mathematics, UCLA*, 2004. (Cited on pages 7 and 22.)

[6] Thiel E.; Montanvert A. Discrete approximation of the euclidean distance for image analysis: improvement of chamfer distances. (Cited on pages 11 and 89.)

[7] Qiang C.; Philippe M.; Quan S.S.; Peng A.H. and De Shen X. Adaptive total variation denoising based on difference curvature. *Image and Vision Computing*, 28 (2010). (Cited on page 156.)

[8] Yakovlev A.S. Window fourier and wavelet transforms. properties and applications of wavelets. *Department of Computational Physics, St Petersburg State University, Russia.* (Cited on pages 9 and 47.)

[9] Goldlucke B. Saddle point problems: Definition, properties and the $tv - \mathcal{L}^2$ model. *Variational methods in Computer Vision II*, 2010. (Cited on pages 11, 89, 105 and 106.)

[10] Li F.; Shen C.; Fan J.; Shen C. Image restoration combining a total variation filter and a fourth-order flter. *ScienceDirect, J. Vis. Commun. Image R.*, 18:322–330, 2007. (Cited on pages 155 and 156.)

[11] Louchet C. *Variational and Bayesian models for image denoising: from total variation towards non-local means.* PhD thesis, Universite Paris Descartes, Ecole Doctoranle Mathematiques Paris-Centre, December 10,2008. (Cited on pages 8 and 155.)

[12] Ambrosio L.; Fusco N.; Pallara D. *Functions of bounded variation and free discontinuity problems.* Oxford Mathematical Monographs. The Clarendon Press Oxford University Press, New York, 2000. (Cited on pages 15, 16, 17 and 18.)

[13] Dr. Boreham D. *Computed Tomography.* PhD thesis, Medical Physics 779, Radiation Health Risks and Benefits. (Cited on page 2.)

[14] Erhan B.; Ghadarghadar N.; Erdogmus D. Automated extraction of blood vessel netwroksfrom 3d microscopy image stacks via multi-scale principal curve tracing. pages 1–4. (Cited on page 135.)

[15] Frederic D. A non-maxima suppression method for edge detection with sub-pixel accuracy. *INRIA Rapport de recherche*, 2724. (Cited on page 167.)

[16] Starck J.L.; Candes E.J.; Donoho D.L. The curvelet transform for image denosing. *IEEE Trans Image Processing*, 11:131–141, 2002. (Cited on page 148.)

[17] Petrova J.; Hostalkova E. Edge detection in medical image using the wavelet transform. *Report of Research, Department of Computing and Control Engineering, Czech Public.* (Cited on pages 11 and 89.)

[18] Rudin L.; Osher S.; Fatemi E. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992. (Cited on pages 21 and 22.)

[19] Almhdie A.; Lopes-Pereira P.; Même S. et al. Chan-vese based method to segment mouse brain mri images: Application to cerebral malformation analysis in trisomy 21. *17th European Signal Processing Conference, EUSIPCO 2009*, 24-28 August 2009. (Cited on pages 3 and 5.)

[20] Dobson D.C.; Santosa F. Recovery of blocky images from noisy and blurred data. *SIAM J. Appl. Math.*, 56(4):1181–1198, 1996. (Cited on page 155.)

[21] Borgefors G. Distance transformations in arbitrary dimensions. *Computer vision, Graphics and Image Processing*, 27:321 – 345, 1984. (Cited on pages 101, 102 and 104.)

[22] Dougherty G. *Digital Image Processing for Medical Applications.* California State University, Channel Islands, April 2009. (Cited on page 1.)

[23] Weiss P.; Blanc-Féraud L.; Aubert G. Efficient schemes for total variation minimization under constraints in image processing. *SIAM J. Sci. Comput.*, 31(3):2047–2080, 2009. (Cited on pages 9, 27, 36, 37 and 38.)

[24] Chan T.F.; Zhou H.M. Total variation wavelet thresholding. *J. Sci. Comput.*, 32(2):315–341, 2007. (Cited on pages 11, 87 and 113.)

[25] Wang Y.; Zhou H.M. Total variation wavelet-based medical image denoising. *International Journal of Biomedical Imaging*, 2006:1–6, 2006. (Cited on pages 11, 87 and 113.)

[26] Daubechies I. Orthonormal bases of compactly supported wavelets. II. Variations on a theme. *SIAM J. Math. Anal.*, 24(2):499–519, 1993. (Cited on pages 9, 47 and 159.)

[27] Joshi G.D.; Sivaswamy J. A simple scheme for contour detection. *VISAPP 2006, Image Analysis*, pages 236–242. (Cited on pages 90 and 135.)

[28] Aujol J.F. Some first-order algorithms for total variation based restoration. *J Math Imaging Vis*, 34:307–327, 2009. (Cited on page 22.)

[29] Breen J.F. Imaging of pericardium. *Journal Thorac Imaging*, 2001. (Cited on page 4.)

[30] Robert A.A.; Fournier J.J.F. *Sobolev spaces*, volume 140 of *Pure and Applied Mathematics (Amsterdam)*. Elsevier/Academic Press, Amsterdam, second edition, 2003. (Cited on page 23.)

[31] Alexandre E.; Guermond J.L. *Theory and practice of finite elements*, volume 159 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004. (Cited on page 17.)

[32] Chang A.E.; Matory Y.L.; Dawyer A.J.; Hill S.C.; Girton M.E.; Steinberg S.M.; Knop R.H.; Frank J.A.; Hyams D.; Doppman J.L. Magnetic resonance imaging versus computed tomography in the valuation of soft tissue of the extremities. *Ann Surg*, 205(4):340–348, April 1987. (Cited on page 3.)

[33] Misiti M.; Misiti Y.; Oppenheim G. Poggi J.M. Wavelet toolbox user's guide. *The Mathwoks Inc*, 1996. (Cited on pages 76, 77 and 79.)

[34] Hornak J.P. *Basics of MRI*. 1996. (Cited on page 3.)

[35] Milgram M.; Cocquerez J.P. Fermeture des contours par un opérateur local. *Traitement du signal*, 3 (6), 1986. (Cited on pages 11, 89, 101, 102, 104, 105 and 106.)

[36] Kaur A.; Singh K. Wavelets for edge detection in noisy images. *National Conference on Computational Instrumentation NCCI 2010*, pages 184–186. (Cited on pages 11, 89 and 93.)

[37] Savagey J.; Chen K. On multigrids for solving a class of improved total variation based pde models. *Report of researcher*, pages 1–20. (Cited on page 156.)

[38] Bergounioux M.; Piffet L. A second-order model for image denoising. *Set-Valued Var. Anal.*, 18(3-4):277–306, 2010. (Cited on page 69.)

[39] Bergounioux M.; Piffet L. A full second order model for multiscale texture analysis. *[hal-00600430]*, 2011. (Cited on pages 16 and 18.)

[40] Piffet L. A locally anisotropic model for image texture extraction. In *Mathematical image processing*, volume 5 of *Springer Proc. Math.*, pages 141–158. Springer, Heidelberg, 2011. (Cited on pages 9, 27 and 69.)

[41] Piffet L. *Decomposition d'image par modeles variationnels - Debruitage et extraction de texture*. PhD thesis, Universite d'Orleans, Pole Universites Centre Val de Loire, Novembre 23,2010. (Cited on pages 8, 9, 16, 17, 18, 27, 33, 36, 40, 43, 72, 73, 144 and 155.)

[42] Andreu F.; Caselles V.; Diaz J. I.; Mazon J. M. Some qualitative properties of the total variation flow. *J. Funct. Anal.*, 188(2):516–547, 2002. (Cited on page 22.)

[43] Bergounioux M. On poincare-wirtinger inequalities in spaces of functions of bounded variation. *[hal-00515451] version 2*, 10 June 2011. (Cited on pages 17, 18 and 25.)

[44] Luisier F.; Thierry B.; Forster B.; Unser M. Which wavelet bases are the best for image denoising? (Cited on page 83.)

[45] Whitaker R.T.; Pizer S. M. A multi-scale approach to nonuniform diffusion. *CVGIP: Image Understanding*, 57(1):99–110, 1993. (Cited on page 22.)

[46] Chinese names. Multi-band a trous wavelet transform for multisensor image fusion. *Bulletin od advanced technology research*, 4(10):14–18, Oct. 2010. (Cited on page 56.)

[47] Abbasi N.M. The application of fourier analysis in solving the computed tomography (ct) inverse problem. *Report of work*, 2010. (Cited on page 3.)

[48] Wirjadi O. Survey of 3d image segmentation methods. *Models and Algorithms in Image Processing*. (Cited on pages 11, 89 and 98.)

[49] Chan T.F.; Golub; Gene H.; Mulet P. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.*, 20(6):1964–1977, 1999. (Cited on page 22.)

[50] Chambolle A.; Lions P.L. Image recovery via total variation minimization and related problems numerische mathematik. *Journal of Mathematical Imaging and Vision*, 76:167–188, 1997. (Cited on pages 7 and 22.)

[51] Chen Z.; Ning R. Breast volume denoising and noise characterization by 3d wavelet transform. *IEEE Transactions on Pattern Analysis and Machine Intelligent*, II(7):674–693, July 1989. (Cited on pages 9, 10, 47, 59, 75, 76 and 77.)

[52] Angel D. S. Efficient closed contour extraction from range image's edge points. *Proceedings of the 2005 IEEE, International Conference on Robotics and Automation*, 10. (Cited on page 90.)

[53] Mallat S. *A wavelet tour of signal processing*. Academic Press Inc., San Diego, CA, 1998. (Cited on pages 9, 47, 48, 51, 56, 149, 150 and 152.)

[54] Mallat S. A theory for multiresolution signal decomposition: The wavelet representation. *Computerized Medical Imaging and Graphics*, 28:235–246, 2004. (Cited on pages 9, 47, 149, 150 and 152.)

[55] Mallat S.; Zhong S. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7). (Cited on pages 11, 89 and 93.)

[56] Mark J. Shensa. The discrete wavelet transform: Wedding the a trous and mallat algorithms. *IEEE Transaction on Signal Processing*, 40(10), October 1992. (Cited on page 56.)

[57] Vese L.A.; Osher S.J. Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.*, 19(1-3):553–572, 2003. Special issue in honor of the sixtieth birthday of Stanley Osher. (Cited on pages 21 and 24.)

[58] Vese L.A.; Osher S.J. Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.*, 19(1-3):553–572, 2003. Special issue in honor of the sixtieth birthday of Stanley Osher. (Cited on page 115.)

[59] Bredies K.; Kunisch K.; Pock T. Total generalized variation. *SIAM J. Imaging Sci.*, 3(3):492–526, 2010. (Cited on pages 9, 15, 19 and 146.)

[60] Bredies K.; Valkonen T. Inverse problems with second-order total generalized variation contraints. *Supported by the Austrian Science Fund (SFB) F32*, 2009. (Cited on pages 19 and 20.)

[61] Strang G.; Nguyen T. *Wavelets and filter banks*. Wellesley-Cambridge Press, Wellesley, MA, 1996. (Cited on page 56.)

[62] Ganguly D.; Chakraborty S.; Kim T.H. A cognitive study on medical imaging. *International Journal of Bio-Science and Bio-Technology*, 3(3), September 2010. (Cited on page 2.)

[63] Changming S.; Pascal V. Fast linear feature detection using multiple directional non-maximum suppression. *The 18th International Conference on Pattern Recognition (ICPR'06)*, 1. (Cited on page 167.)

[64] Unaldi N.; Asari V.K. Undecimated wavelet transform-based image interpolation. *ISVC 2010*, III:474–483, 2010. (Cited on page 56.)

[65] Monzon L.A.; Beylkin G.; Hereman W. *On almost interpolating and nearly linear phrase compactly supported wavelets (Coiflets)*. PhD thesis, University of Colorado, Boulder CO 80309-0526 and Department of mathematical and Computer sciences, Colorado School of Mines, Golden CO 80401-1877., April 10,1998. (Cited on page 160.)

[66] Meyer Y. Oscillating patterns in image processing and in some nonlinear evolution equations. *The fifteen Dean Jacquelines B. Lewis Memorial Lectures*, March 2001. (Cited on pages 11, 21, 22, 23, 24, 113, 114 and 115.)

[67] Nesterov Y. Gradient methods for minimizing composite. *CORE, Discussion paper.*, 2007. (Cited on pages 9, 27, 36 and 37.)

[68] Daniel T.L.; Akio Yamamoto. Wavelet analysis: Theory and application. *Hewlett-Packard Journal*, pages 44–52, December 1994. (Cited on page 158.)