Computer Aided Medical Procedures
Prof. Dr. Nassir Navab

Ph.D. Thesis

Random Forests for Medical Applications

Olivier Pauly

Fakultät für Informatik
Technische Universität München

TECHNISCHE UNIVERSITÄT MÜNCHEN

Chair for Computer Aided Medical Procedures & Augmented Reality / I16

Random Forests for Medical Applications

Olivier Pauly

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. J. Schlichter
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. N. Navab
	2.	Prof. Dr. N. Ayache, INRIA, Sophia Antipolis, France

Die Dissertation wurde am 02.01.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 05.07.2012 angenommen.

Abstract

Machine learning is a key component for integrating the knowledge and experience of physicians in medical imaging. With the design of algorithms that are able to generalize from observed evidences, and to make predictions about unseen data, machine learning can be applied in many fields such as computer aided diagnosis, detection and segmentation. In the last decade, random forests became a popular ensemble learning algorithm, as they achieve state-of-the-art performance in numerous computer vision tasks. Consisting in an ensemble of independent decision trees, random forests are very intuitive models that offer a flexible probabilistic framework for solving different learning tasks. Following a divide and conquer strategy, they efficiently create partitions of high-dimensional feature spaces, and model probability distributions in each cell of these partitions. Thereby, they permit to approximate any arbitrary functions or densities for classification, regression or clustering tasks.

In this thesis, we formalize random forests models as ensemble partitioning approaches and propose novel related techniques for classification, regression and clustering. We introduce new task-specific forest models and demonstrate their great potential in different medical applications such as organ localization, segmentation, lesion detection and image categorization. First, multiple organ localization is formulated as a regression problem, in which each voxel votes for the position of all organs of interest. Therefore, we instantiate forest-related techniques to solve efficiently this regression task, and show the benefits of our approach in Magnetic Resonance scans. Further than localization, we tackle the problem of multiple organ segmentation in Computer Tomograms. As strong prior knowledge such as organ arrangement, their size and shape is contained in annotated scans, we propose to integrate such rich information within a novel structured output forest model. Built on a joint classification-regression formulation, our method enforces leaf clusters that are consistent in terms of organ class and spatial location, and learns thereby spatial regularization directly from the data. Through extensive experimentation, we demonstrate the ability of our approach to provide improved class predictions compared to the classical classification strategy. Afterward, we address the problem of detecting Parkinson-related lesions within the midbrain in 3D transcranial ultrasound. To this end, we formulate a detection paradigm that mimicks human experts by using probabilistic modeling of visual and spatial information based on random forests. On a highly challenging database of 3D-TCUS volumes from 22 subjects, our approach show very promising results relatively close to the human inter-rater observability. Finally, to recognize the modality of a medical image, we propose a fast clustering approach based on random ferns to build a dictionary a visual words. Moreover, we introduce in this context a novel clustering approach based on multiple-decisions stumps that we call STARS. Taking advantages of extreme randomization, our both methods achieve very good performance on a real medical database.

Keywords:
Machine Learning, Random Forests, Random Ferns, Medical Image Analysis

Zusammenfassung

Maschinelles Lernen stellt eine wichtige Komponente dar, um das Wissen und die Erfahrung medizinischer Experten in Bildgebungs-Anwendungen wie Computer-gestütze Diagnose, Erkennung und Segmentierung zu integrieren. Im letzten Jahrzehnt sind Random Forests ein populärer Algorithmus für "Ensemble Learning" geworden, da sie in zahlreichen Computer Vision-Problemen state-of-the-art Ergebnisse erbringen. Als Ensemble von voneinander unabhängigen Entscheidungsbäumen sind Random Forests sehr intuitive Modelle, die einen flexiblen, probabilistischen Rahmen anbieten, um verschiedene Lernaufgaben zu lösen.

In dieser Dissertation formalisieren wir Random Forest-Modelle als Methode für "Ensemble Partitionierung", und wir schlagen neue verwandte Techniken zur Klassifikation, Regression und Clustering vor. Wir führen neue Aufgaben-spezifische Forest-Modelle ein, und wir zeigen ihr großes Potenzial in verschiedenen medizinischen Anwendungen wie Lokalisierung von Organen, Segmentierung, Erkennung von Läsionen und Bildkategorisierung. Zuerst wird die Lokalisierung von Organen als Regressions-Problem formuliert, indem jeder Voxel zur Positionsbestimmung jeweils aller Organe beiträgt. Somit verwenden wir Forests-verwandte Techniken um diese Regression Aufgabe zu lösen, und zeigen die Vorteile unserer Methode in Kernspin-Tomogrammen. Über die Lokalisierung hinaus gehen wir die simultane Segmentierung mehrerer Organe in Computer-Tomogrammen an. Da starkes Vorwissen wie die Anordnung von Organen, deren Grösse oder Form in annotierten Datensätzen enthalten ist, schlagen wir vor, diese reiche Information in einem neuen Strukturierten-Output-Forest-Modell zu integrieren. Basierend auf einer gemeinsamen Klassifizierungs- und Regressions-Formulierung erzwingt unsere Methode die Erzeugung von "Leaf-Clusters", die konsistent in Organklasse und räumlicher Position sind. Dadurch wird eine räumliche Regularisierung direkt aus den Daten gelernt. In zahlreichen Experimenten demonstrieren wir die Fähigkeit unserer Methode, verbesserte Klassvorhersagen zu leisten. Danach wenden wir uns dem Problem der automatischen Erkennung von Parkinson-assoziierten Läsionen im Mittelhirn mittels 3D transkraniellen Ultraschall zu. Zu diesem Zweck formulieren wir ein Detektionsparadigma, welches menschliche Experten imitiert, indem es visuelle und räumliche Informationen probabilistisch durch den Einsatz von Random Forests modeliert. Unsere Methode zeigt auf einer anspruchsvollen Datenbank von 3D transkranielle Ultraschall Volumen mit 22 Probanden sehr vielversprechende Ergebnisse, welche sich sehr gut mit den Beobachtungen menschlicher Interrater decken. Um die Modalität eines medizinisches Bild zu erkennen, stellen wir schließlich einen effizienten Random Ferns-basierten Clustering-Algorithmus vor, um ein visuelles Wörterbuch zu lernen. Außerdem führen wir in diesem Zusammenhang eine neue Clustering-Methode ein: die sogenannten STARS, als Ensemble von Multi-Entscheidung-Stumps. Mit dem Vorteil von extremer Randomisierung erreichen unsere beiden Methoden sehr gute Ergebnisse auf einer echten medizinischen Datenbank.

Schlagwörter:
Maschinelles Lernen, Random Forests, Random Ferns, Analyse Medizinischer Bilddaten

Acknowledgments

"I can no other answer make, but, thanks, and thanks."
William Shakespeare

These are (hopefully) the last few sentences I am writing in this thesis. I would like to dedicate them to all the people that helped me throughout this crazy adventure during the four years of my PhD. First of all, I would like to thank my PhD advisor Nassir Navab, not only for his constant trust and support, but also for offering me the great chance of doing research on my beloved topic of machine learning applied to medical imaging, and this, in the wonderful environment of the Computer Aided Medical Procedures group.

In fact, I am deeply thankful to all the guys from our group. I would like to start by mentioning Nicolas Padoy, for his great support and supervision during the first year of my PhD, for all the great discussions we had, and for all the time he spent reading my thesis and giving precious feedback. After his departure, I had the chance to get a great supervision from Diana Mateus, who always brought me to new ideas and challenges. Then, I would like to thank my first office mates Martin Groher and Hauke Heibel, for our collaboration, for the many discussions on medical imaging, for answering all my (philosophical?) interrogations on programming, and of course for our regular (research) meetings in numerous pubs in Munich. Furthermore, I would like to thank Ben Glocker, not only for the great time we had in Cambridge, but also for offering me to crash at his place when I arrived in England, for the exciting research we did together, and for all the nice BBQ afternoons in his garden. Afterward, I would like to thank Stefan Hinterstoißer, not only for being a very precious kicker teammate, but also for the nice ice cream breaks on the terrace at the university and for the numerous sunny days spent in Munich's beer gardens. I would also like to thank Selen Atasoy for all the exciting discussions on so many research topics such as wavelet theory, manifold learning or cognitive science. Further, I had the chance to collaborate with Ahmad Ahmadi, and I would like to thank him for the great time and the great work on 3D transcranial ultrasound. I would also like to mention that without the constant support and help of Martin Horn and Martina Hilla, nothing would have been possible. Finally, I would like to thank all the other CAMP guys: Darko Zikic, Max Baust, Pierre Georgel, Marco Feuerstein, Andreas Keil, Stefan Holzer, Jose Gardiazabal, Loren Schwarz, Slobodan Ilic, Nicolas Brieu, Cedric Cagniart, Richard Brosig, Tobias Lasser, Mehmet Yigitsoy, Steffi Demirci and many others. I would also

like to mention the very good time we had at each MICCAI conference, which was not only an opportunity to discover new exciting works in the field of medical imaging, but also to meet our friends from Imperial College, especially Pete Mountney, Dan Stoyanov and Matina Giannarou.

During these four years, I had the chance to visit two research centers which are Siemens Corporate Research and Microsoft Research Cambridge. I would like to take this opportunity say how much I am thankful to Gözde Ünal and Antonio Criminisi for giving me the chance to work with them, to learn so much and to be involved in such great research projects. Furthermore, I would like to thank Axel Möller-Martinez from the department of Nuclear Medicine at Klinikum Rechts der Isar for his great collaboration. Finally, I would also like to thank the members of my thesis committee Prof. Nicholas Ayache and Prof. Johann Schlichter, for accepting my request to be in my thesis committee, for taking the time to read this thesis, and for taking part to my PhD defense.

I would like to conclude these ackowledgements by saying how much I am deeply thankful to my parents, for their inconditional support through all the years, while I was studying, when I decided to go to Germany, and during my PhD. I would like also to especially thank my grandfather, who always challenged me to explain my work using intuitive explanations. This definitely helped me in improving the way I present my work, design my slides and write this thesis. Finally, I would like to say how grateful I am to have my wonderful Stephanie, that supported me through this adventure with so much love, patience, positive energy, good mood and humour. She is my sunbeam that always knows how to motivate me and cheer me up, or how to make me laugh and distract me by taking me out. Thanks for all.

P.S.: Thanks to all the bands that wrote the wonderful music I've been listening to during thousands of hours of crazy hacking and thesis writing.

CONTENTS

THESIS OUTLINE

Chapter 1: Machine Learning for Medical Applications In this first chapter, we propose a short introduction to machine learning and give a brief overview of its different applications in medical imaging. Starting with the well-known definition of Tom Mitchell, we describe the key components of machine learning and the different tasks of supervised, semi-supervised and unsupervised learning. Then we present a few examples of medical applications such as computer aided diagnosis, organ localization, segmentation, registration, tracking, medical image categorization and retrieval. Afterward, we discuss the challenges of applying machine learning to medical imaging problems, *e.g.* the difficulties of collecting medical data and building a reliable ground truth. Finally, we conclude this chapter by presenting our contributions in the present thesis.

Chapter 2: Random Forests This second chapter constitutes the methodic core of the thesis. Here, we start by defining the decision tree model as a partitioning approach. Aiming at subdividing observations, each tree can be seen as a directed acyclic graph, where each node consists of a splitting function and a posterior model. We detail how to train a tree following a greedy optimization strategy: at each node, several splitting function candidates are generated and the best is chosen according to a task-specific objective function. Finally, a posterior model can be learned from the training data at each leaf of the tree in order to make predictions about new incoming observations. To improve the generalization of single decision trees and overcome their limitations, Random Forests have been introduced as ensemble of independent trees [11]. Two classical strategies for creating independent trees are detailed in this chapter: (1) by using bootstrap aggregating ("bagging") or (2), by injecting randomness in the node optimization. The prediction of a random forest can then be computed by simply averaging the contributions of each individual tree. Afterward, we show that random forests can be instantiated for classification, regression and clustering tasks, by designing appropriate objective functions and posterior models. Additionally, we propose to illustrate this chapter by providing a few intuitions on their behaviour using numerous toy examples.

Chapter 3: Related Ensemble Partitioning Approach: Random Ferns In this third chapter, we present a forest-related approach which is based on the similar principle of ensemble partitioning, namely the Random Ferns, which are often presented as ensemble of constrained trees [71]. We demonstrate in this thesis that in contrast to trees,

they are not hierarchical but can be interpreted as an intersection of decision stumps. Similarly, they can be easily adapted for different tasks such as classification, regression and clustering. Through several toy examples, we investigate the behaviour of Random Ferns, and this, for classification and regression problems.

Chapter 4: Random Forests: Contributions in Medical Applications The fourth chapter reports our contributions in medical imaging using forests-related techniques. We demonstrate their great potential and introduce novel application-specific models by: (1) using an appropriate problem formulation, (2) designing a task-specific objective function and (3), defining an adapted posterior model. First, we present efficient regression approaches based on random ferns and forests to estimate the position and the size of multiple organs of interest in MR Dixon sequences [73]. Compared to state-of-the-art atlas registration, our approaches show better localization accuracy for an increased robustness. Second, we tackle the problem of multiple organ segmentation and propose a new random forest model based on a joint classification-regression formulation. Through exhaustive experimentations, we demonstrate that this joint formulation yields better results than classification by learning spatial smoothness directly from the data. Thereafter, we address the problem of detecting Substantia Nigra echogenicities in 3D transcranial ultrasound, which are related to Parkinson disease. To this end, we formulate a detection paradigm that mimicks human experts by using probabilistic modeling of visual and spatial information based on random forests. To learn this spatial information, we propose a novel parametrization based on two hemisphere-specific coordinate systems that accounts for asymmetric changes of scales and orientation of the midbrain anatomy. On a database of 3D-TCUS volumes from 22 subjects, our approach show very promising results relatively close to the human inter-rater observability. Afterward, we report our work on modality recognition based on the visual content of a medical image [75]. To this end, we use a random ferns clustering approach to build efficiently a dictionary of visual words. On a real database of medical images, we illustrate the advantages of our approach in terms of speed and accuracy. Finally, we propose a novel approach that we call STARS [76], which builds upon an ensemble of multi-decision stumps. We show how to instantiate them for classification and clustering, and analyze their behaviour on a few toy examples. In the same context of modality recognition, STARS are derived for dictionary learning and achieve impressive results that are slightly better as hierarchical K-means or random ferns clustering.

"You can make predictions about everything but the future"
Lao Tzu

CHAPTER

ONE

MACHINE LEARNING IN MEDICAL APPLICATIONS

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"
Tom Mitchell.

1.1 An Illustrated Introduction to Machine Learning

Human being has always been a curious creature, constantly aiming at increasing his knowledge about the world he lives in. In order to build new knowledge on a given topic of interest, the first step is to collect a large amount of observations or realizations. To illustrate our point, let us take the example of an entomologist, who wishes to study a particular insect species, namely the ants. Before starting to build a generic anatomical model, or to construct a categorization, an entomologist needs to go on the "field", walk in the nature and start collecting observations by taking pictures, capturing some specimens or making drawings as shown on fig.1.1. Once a maximum of information from different sources is available, our expert can start analyzing the observations, by extracting some common characteristics or morphological features such as the size, the color, the weight, the shape or the presence of wings. Then, by looking across all observations, performing comparisons based on their different characteristics or features, the entomolgist can identify similar subgroups, categorize his specimens into different casts such as workers, soldiers or queens, build a generic anatomical model for each of these classes, and make predictions for new observed ants.

Towards achieving new knowledge, the typical process of learning consists thus of following steps: (1) collecting a large amount of observations, (2) extracting relevant information, (3) designing a general model that best explains past and future observations. In the case where the complexity of the object of interest is low and the observations are consistent, this process seems scalable for a human being. However this becomes more difficult if the object of interest is the realization of a complex phenomenon, possibly influenced by a lot of factors, and where a large amount of information is available. In this

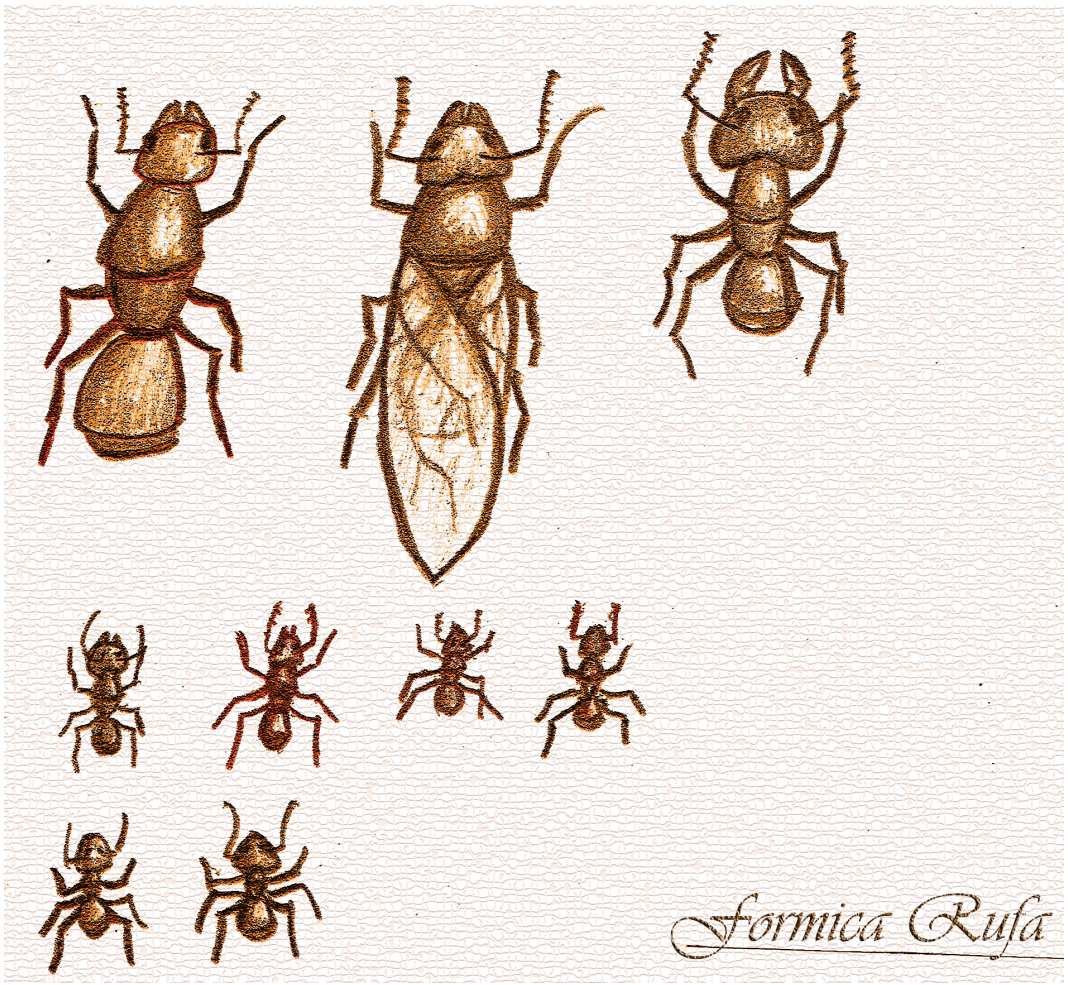

Figure 1.1: Observations: drawing of different ant specimens of the species Formica Rufa

context, a new field emerged a few decades ago to develop computer-based approaches supporting humans in building new knowledge directly from observations: Machine Learning.

Considered as a branch of artificial intelligence, machine learning aims at designing algorithms that are able to learn from past experience in order to make predictions about the "future", *i.e.* new observations. In his tentative definition, Tom Mitchell introduces 3 important components of a learning algorithm. First, the notion of "**experience**" can be understood as the act or process of directly perceiving events or reality according to the Merriam-Webster dictionary. In machine learning, the "**experience**" consists of all the observations of a phenomenon and eventually their associated interpretation. Second, the term "**task**" refers to the goal of the learning algorithm, *e.g.* making decisions, predictions or adapting a behaviour facing new observations. Third, a "**performance**" measure needs to be defined so that a learning system can assess and optimize its own learning ability based on the desired outputs. Taking the example of the categorization of ant specimens, the learning **task** could be defined as the classification of ants into different casts, based on the **experience** consisting in a set of ant observations and their class labels. A

performance measure could be for instance the ratio of prediction errors estimated on new unseen ant observations. So finally, this definition gives a basis explaining the inputs and the outputs of a learning algorithm and that it can measure and improve its perfomance on its own. But one crucial term remains undefined: "learning". "Learning" refers to the process of understanding the observations and inferring a model able to generalize from them. Once the learning phase finished, the system is able to perform predictions about unseen data and to react by making decisions.

Machine learning is applied in many fields where: (1) an immense amount of information from different sources or sensors is available, (2) there is only limited knowledge on the topic, and (3), uncertainty has to be taken into account. Explaining complex phenomena with an appropriate and realistic mathematical model can be a very difficult task. To leverage this problem, machine learning approaches proposes to learn directly from the data. Whether a learning system requires teaching from a human or not, it can be classified into different classes of tasks: supervised, unsupervised and semi-supervised.

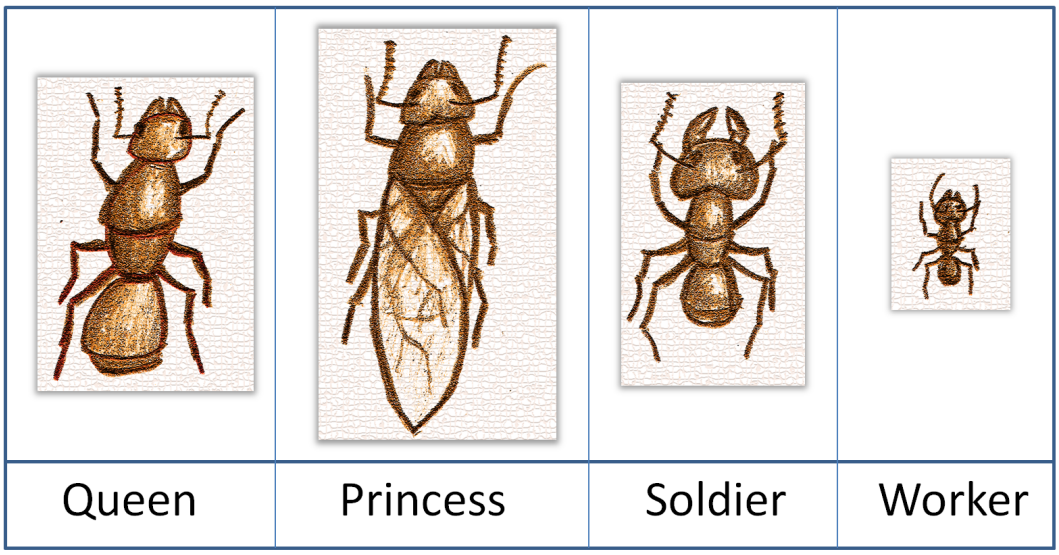

Figure 1.2: Different ant classes: here are depicted 4 different ant casts of the Formica Rufa species, namely the "queen", "princess", "soldier" and "worker"

Supervised Learning: The goal of supervised learning is to design algorithms to teach a system to make decisions or perform predictions based on observations. For instance, let us consider the problem of recognizing the cast of an ant based on its morphological features. Given some specimens with their associated casts as shown on fig.1.2, the goal is to find some discriminant characteristics to be able to classify new observed specimens into one of the following classes: "Queen", "Princess", "Soldier" or "Worker". As we are learning from annotated examples or in other words, a training set, such a task is called *supervised* learning. By looking at the overall size, the presence of wings, the size of the mandibles and the presence of reproduction organs, we could easily design a system which automatically assigns a new observation to one of the previously defined classes. Such a supervised problem is called **classification** and the learning system as shown on fig.1.3

Figure 1.3: Classifier: supervised learner which first learns from annotated observations, and then permits to predict the class label of a new incoming specimen.

is then called **classifier**.

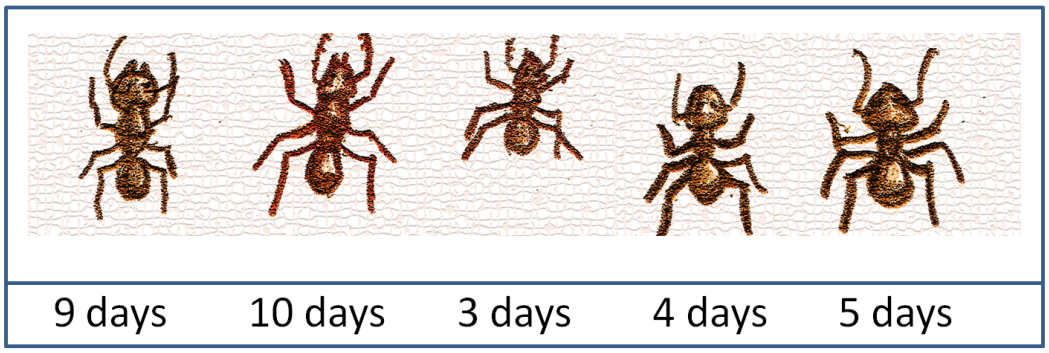

Figure 1.4: Ant aging: here are depicted several ants and their corresponding age

Now let us consider the case depicted in fig.1.4, where we would like to predict the age of the observed specimens. We have exactly the same setup, *i.e.* we are given a training set of ants with their corresponding age, and we want to design a prediction system based on morphological characteristics. The desired output being a continuous variable, such a problem is a supervised problem called **regression**.

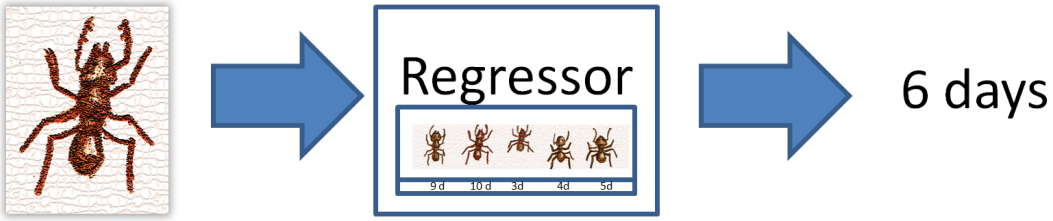

Figure 1.5: Regressor: supervised learner which first learns from annotated observations, and then permits to predict the age of a new incoming specimen.

To summarize, supervised learning consists of two types of tasks, namely classification and regression, in which a (human) *teacher* provides a training set of both observations and corresponding outputs to the learning system. Formally, let us denote by $\mathbf{X} \in \mathcal{X}$ a multi-dimensional observation vector containing for instance all morphological characteristics of an ant, where $\mathcal{X} \subset \mathbb{R}^D$ is called input feature space. To each observation $\mathbf{X} \in \mathcal{X}$, an output $\mathbf{Y} \in \mathcal{Y}$ is associated, where $\mathcal{Y} \subset \mathbb{R}^{D'}$ is the (multi-dimensional) output

space. Depending on the type of output, we can distinguish between the two supervised learning tasks classification and regression: in classification, $\mathbf{Y}$ is a one-dimensional value from a finite set of discrete labels, and in regression, $\mathbf{Y}$ consists of one or more continuous values. Given a training set $\left\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\right\}_{n=1}^{N}$, which embodies what Tom Mitchell calls experience and consists of N past observations and their corresponding outputs, the goal of a learning algorithm is to find a prediction function $\Psi(\mathbf{X}) = \mathbf{Y}$, or in a probabilistic fashion, to model the conditional distribution $P(\mathbf{Y}|\mathbf{X})$ or the joint distribution $P(\mathbf{X}, \mathbf{Y})$. During a training phase, the parameters of the function or distribution model are optimized using the training set according to a predefined performance measure or objective function. Afterward, the trained system can be used to perform predictions for a new unseen observation $\mathbf{X}$, *e.g.* using $\hat{\mathbf{Y}} = \Psi(\mathbf{X})$ or $\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}\in\mathcal{Y}} P(\mathbf{Y}|\mathbf{X})$.

Unsupervised Learning: In unsupervised learning problems, there are no output associated to the observations. The goal may be then to: (1) discover similar groups in the feature space $\mathcal{X}$, known as *clustering* task, or (2) to estimate the distribution of the observations in $\mathcal{X}$ called *density estimation* [9]. So if we consider again our entomology example, now we are given only a few ant specimens, and we aim at discovering similar groups based on a few characteristics. Considering a set of observations $\left\{\mathbf{X}^{(n)}\right\}_{n=1}^{N}$, while the goal of density estimation is to model the distribution $P(\mathbf{X})$, clustering aims at identifying a set of clusters $\mathcal{K} = \{\mathbf{K}_k\}_{k=1}^{K}$ which represent (non-overlapping) subsets of consistent observations within $\mathcal{X}$. Here, the training can be done by optimizing an objective function especially designed in the input feature space $\mathcal{X}$. As illustrated by fig.1.6, the choice of morphological features is crucial for unsupervised tasks as they can yield very different results.

Semi-supervised Learning: In some cases, outputs are known only for a few observations. If we denote by $\left\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\right\}_{n=1}^{N}$ the subset having a corresponding output and $\left\{\mathbf{X}'^{(m)}\right\}_{m=1}^{M}$ the remaining observations, the goal here is to learn the function Ψ or estimate the distributions $P(\mathbf{Y}|\mathbf{X})$ by making use of both data sets. This task can be solved for instance by optimizing a joint objective function consisting of a supervised term using $\left\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\right\}_{n=1}^{N}$, and an unsupervised term based on the remaining $\left\{\mathbf{X}'^{(m)}\right\}_{m=1}^{M}$.

After this brief introduction, we will discuss in the next section the challenges of applying machine learning in the field of medical imaging.

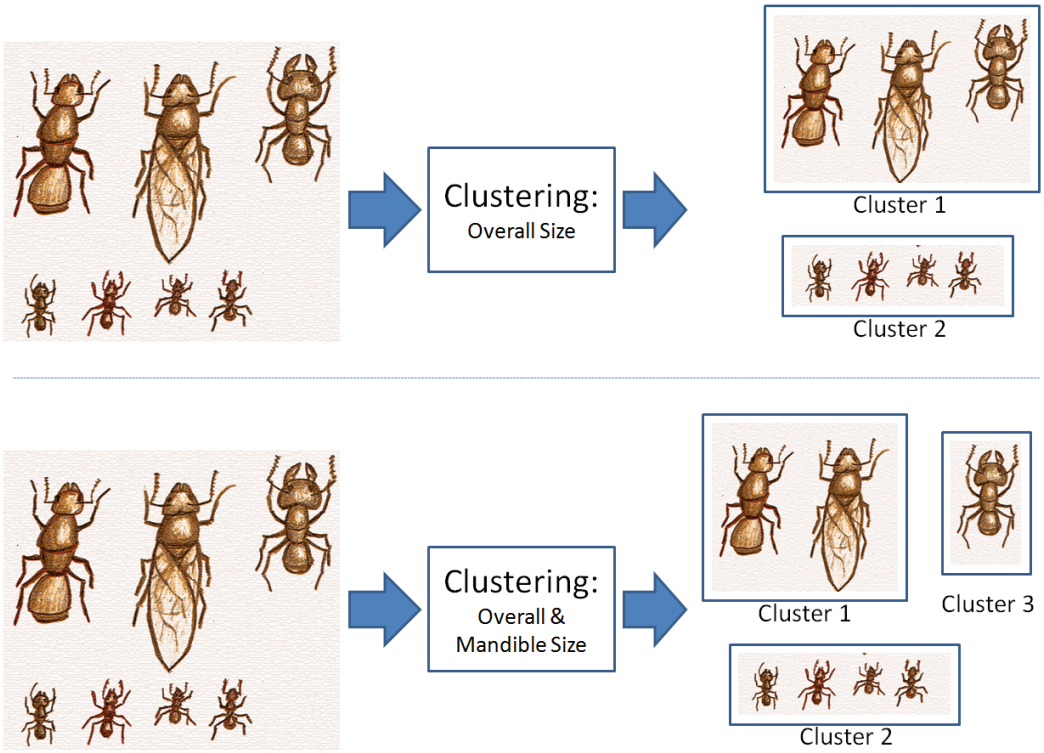

Figure 1.6: Clustering: unsupervised learning that aims at discovering subgroups within the set of ant specimens.

1.2 Applying Machine Learning to Medical Applications

Machine learning found applications in many fields such as genetics, natual language processing, search engines, computer vision, computational finance or stock market analysis. In the case of medical applications, the interest for learning-based methods seems more recent. Nevertheless, this trend is increasing, and more works containing machine learning as keyword are published each year. Moreover medical imaging conferences such as MICCAI started to dedicate sessions and workshops to learning-based approaches in medical imaging.

This "late" gain of interest could be explained by the fact that applying machine learning to medical imaging is very challenging, and this for several reasons. First, as "objects" of interest are human patients, risk must be minimized and methods that are actually transferred into the clinical routine requires to be well understood, highly reliable and robust. Suffering from their reputation of being black-box machines, learning-based approaches needs to be demystified and thoroughly studied to achieve higher acceptance in the medical context. Second, compared to computer vision, medical images are highly multi-dimensional (3D or more, several channels), they are multi-modal *i.e.* acquired with very different imaging systems, they have very different resolutions and can suffer from low signal to noise ratio. Hence, learning methods needs to be scalable to high-dimensional

problems and robust to noisy or ambiguous information. Third, such approaches require a lot of training data to avoid overfitting and gain enough generalization. Compared to the time needed for taking a picture with a digital camera and making it publicly available on the internet, acquiring medical images is a very long process. Moreover, since subjects are real patients, data needs to be anonymized and has to remain often within the scope of joint projects with hospitals. Consequently, making medical images publicly available for the community is very difficult, and building huge training sets or benchmark databases is very challenging. Fourth, most of the learning-based tasks in medical imaging need to be supervised, *i.e.* data needs to be manually annotated. Building a reliable ground truth or in other words (human) gold standard is often a very tedious task which requires three key resources: (i) a set of medical experts, (ii) a user-friendly and efficient annotation tool, and (iii) a lot of time. Having more than only one expert is crucial to reduce the impact of inter- and intra-observer variability. Indeed, in modalities that need high interpretation skills such as ultrasound, annotation results may vary dramatically depending on the experience of the expert and on the time spent in the annotation task. In fact, due to a "learning" effect, the same observer can provide very different annotations of the same data, depending if it is seen at the beginning or at the end of the labelling process. Hence, to improve the quality of the gold standard, multiple annotations needs to be collected for each data and then merged. As illustrated by fig.1.7, another solution would be to put the human expert and the learning algorithm in the same loop, and to iteratively alternate between machine labelling and human correcting phases until both converge to a ground truth. Nevertheless real breakthroughs can be achieved in medical imaging by proposing approaches that requires only a little or no supervision at all. In the next section, we present a few applications of machine learning in the field of medical imaging such as computer aided diagnosis, detection/segmentation, registration/tracking and image categorization/retrieval.

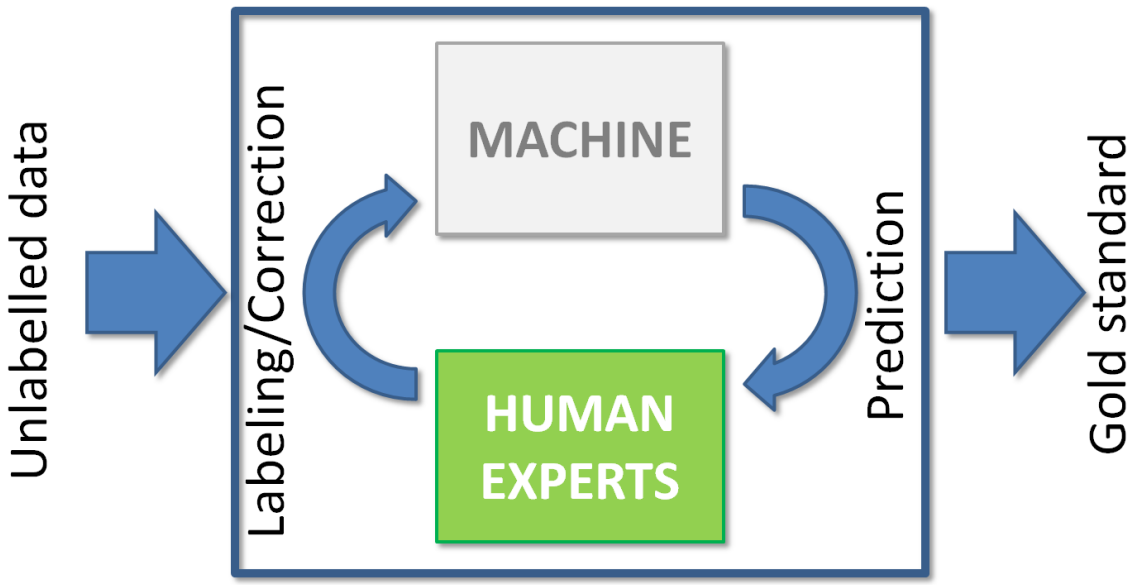

Figure 1.7: Human-Machine Iterative Labeling: iteratively alternate between machine labeling and human correcting phases until both converge to a ground truth

1.3 Learning-based Approaches in Medical Applications

In the medical field, knowledge mainly builds upon the experience or the amount of evidences accumulated by medical experts in the hospitals all over the world. The human body is a very complex machinery, that consists of many components, and which can be influenced by many factors. Hence, modeling its different functions or disfunctions is a very difficult task. With its ability of generalizing from past observations and to perform predictions, machine learning seems to offer the perfect tools to integrate the experience and knowledge of medical experts into medical imaging applications.

The last decade witnessed an increasing interest for learning-based approaches to solve different tasks in the medical field. For instance, computer aided diagnosis aims at supporting experts decisions based on different information sources such as imaging data, symptoms and patient information. In medical image analysis, numerous learning-based approaches permit to automatically detect and segment anatomical structures or diseases in any type of images. Recently, many content-based retrieval techniques have been introduced to provide new access to the information contained in medical databases. This permits to support diagnosis or therapy decisions by retrieving similar cases that have been encountered in the past, or to easily access information from multiple sources for research and teaching. Furthermore, for multi-modal image registration or tracking of medical tools, improvements in robustness and accuracy have been achieved by learning application-specific similarity measures directly from the data.

In the following, we will give a few examples of learning-based approaches in these key medical applications.

1.3.1 Computer Aided Diagnosis

According to the Merriam-Webster dictionary, the definition of diagnosis is *the art or act of identifying a disease from its signs and symptoms*. Nowadays, many imaging systems are available to investigate the different "signs" or symptoms and provide additional information. However, medical images may be very difficult to interpret as they sometimes provide a low signal to noise ratio. Diagnosis becomes then an art or act of **interpretation**, that is subjective and highly depends on the experience level of the observer and the time allowed for the investigation. The role of computer aided diagnosis is to support medical experts bridging the gap between subjective **interpretation** of patient data and objective **identification** of diseases. Trained using past experience of multiple experts, a learning-based diagnosis system permits to improve the objectivity and thereby the reliability of the diagnosis, reducing inter- and intra-observer variability. Obviously, increasing the reliability of diagnosis is crucial for the screening of invasive and lethal diseases such as cancer, where early diagnosis is primordial for reducing mortality rate.

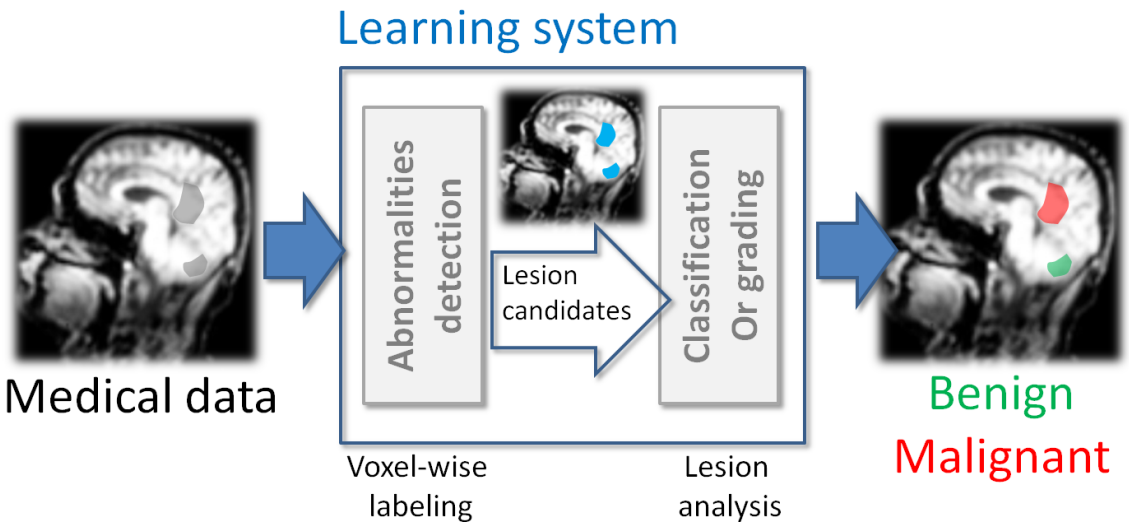

Figure 1.8: Computer Aided Diagnosis: classical learning-based approaches rely on two phases: first the detection or segmentation of possible lesions, and then the classification into benign or malignant

Since medical imaging became digital, machine learning can play a crucial role to support early diagnosis of cancer. The last decade, a lot of learning-based techniques have emerged, most of them based on a two-phases framework as illustrated by fig.1.8: (1) detection and/or segmentation of abnormalities, and (2) classification of the detected abnormality into benign or malignant. Both phases can be casted as a supervised classification task, first providing semantic information by classifying each pixel/voxel as belonging to a suspicious lesion or not, and second permitting to further analyze or quantify the detected/segmented abnormalities and to finally classify them as dangerous or not. Classification being a supervised learning task, it requires a training set consisting of training images where cancer lesions have been manually labelled and identified as benign or malignant. A typical classification scheme consists of three components: (1) feature extraction, (2) feature selection or dimensionality reduction, and (3) classifier. In the first

component, relevant and discriminative information is extracted from the raw images. For instance, to characterize pixel context in medical images, features are computed to encode visual and textural information. Second, feature selection or dimensonality reduction aims at providing a more compact and hopefully discriminative feature representation. Third, a classifier has the role of assigning to each pixel a label "suspicious" or not. Let us give a few examples of learning-based approaches aiming at supporting early diagnosis of lethal diseases such as cancer or atherosclerosis.

As investigated in [83], many works have been focussing on breast cancer which is the most invasive and the first cause of cancer-related death for women. In [87, 16, 69], authors propose automated methods for detecting suspicious masses and micro-calcifications in mammograms. In [108, 96], alternative approaches proposed to use Dynamic contrast enhanced Magnetic Resonance (MR) images for cancer screening. Following a classical workflow, tumors are first delineated in the MR images and then classified as benign or malignant. Similarly, early diagnosis of lung cancer has motivated much research towards automatic detection of glass nodules in computer tomogram (CT) lung images [97, 111, 27]. As these types of nodules have been reported to have a higher probability of becoming malignant, it is crucial to detect them reliably. Proposed approaches aims at first segmenting tumor candidates in the CT images and then classify them as benign or malignant based on some volumetric and textural features. Recently, to improve the quality of prostate cancer diagnosis, learning-based approaches have been explored in [59, 98, 2] using different imaging modalities such as MR, spectrocopy or histopathological images. Again, the goal is first to detect cancerous lesions and second to classify them into different tumor grades. In the field of dermatology, the quality of skin cancer screening highly depends on the experience of the dermatologist. Based on optical images [86] or spectroscopy, learning techniques are also gaining interest for the automatic classification of skin moles into benign or malignant and to distinguish between the different types of skin cancer. Since changes in colors, shape or texture are typical from skin cancer, the mole is first segmented and characterized by extracting color, textural and shape features. Then, a classifier trained on an annotated database of optical images permits to decide whether the mole is dangerous or not.

To prevent heart attacks and monitor the evolution of atherosclerosis in coronary arteries, the most commonly used investigation technique is intravascular ultrasound (IVUS). IVUS permits to acquire high-resolution images of the inner wall of coronary arteries, by using a rotating ultrasound probe which is inserted through a catheter into the femoral artery. Based on RF data or the ultrasound images, many learning methods have been proposed [107, 95, 94] to assess the composition of atherosclerotic lesions within the artery walls. Using visual and textural features, each pixel can be classified into one of the typical classes of tissues such as lipid or fibrous composing such plaque. Finally, based on the assessment of the composition and the morphology of atherosclerotic plaque, the risk of rupture and thereby of heart attack can be evaluated.

1.3.2 Multiple Organ Localization and Segmentation

Medical image analysis aims at first, understanding the semantic content of medical images, and second, extracting and quantifying relevant information for diagnosis or research. Automatic **localization** of anatomical structures and organ **segmentation** are typical problems that can be tackled using learning-based approaches. As illustrated by fig.1.9, while the localization task can be defined as automatically finding the position, the size and optionally the main orientation of an organ, the segmentation task aims at delineating the boundary of an organ by for instance assigning a label to all its voxels.

Localizing automatically multiple anatomical structures permits to **augment** the content of raw medical images by providing additional semantic information. While it could be considered as a preprocessing step for further organ segmentation, diverse clinical applications can benefit from such automatic annotation, such as semantic navigation or content-based retrieval. By registering a new patient scan with an annotated "atlas" scan, the position of all organ of interest can be easily inferred by transferring these annotations to the new patient data. This approach, known as atlas-based registration, is considered as state-of-the-art for multiple organ localization. However, for large field-of-view scans, this task becomes very difficult due to high inter-patient variability.

Inspired by their success in computer vision, machine learning approaches were introduced a few years ago for solving the task of anatomy detection. For instance, aiming at localizing the heart chambers in 3D cardiac CT, a new learning-based method called marginal space learning (MSL) was introduced by Zheng *et al.* in [109]. Using a detection framework, authors propose to break down the complexity of exhaustive search in the full 3D similarity transformation space by using a cascade of three classifiers. Sequentially, the first classifier identifies probable candidates for the 3D position of the organ of interest, and the followings perform a refinment search in position-orientation and finally in the full 3D pose. Recently, regression-based solutions emerged to tackle the problem of organ localization. Exploiting the fact that first, strong prior knowledge is available on human anatomy and second, image acquisition procedures are often standard procedures, it can be expected that voxels, based on their contextual information, can predict the surrounding anatomy. For instance, if a voxel neighborhood shows visual characteristics that are typical of heart tissues, besides the position of the heart, this voxel can also provide a confident estimate of the position of the nearby lungs. In this context, Zhou *et al.* presented in [112] a regression approach for localizing the left ventricle in 2D cardiac ultrasound images. There, a function is learned directly from annotated data to predict the relative position, scale and orientation of the left ventricle.

Going from organ localization towards organ segmentation is far from being straightforward. Indeed, while a classification formulation seems to be a natural choice for assigning an organ label to each pixel, it suffers from a lack of spatial consistency. To tackle this problem with a learning based approach, several options are available: (i) pixel/voxel-wise classification coupled with spatial regularization, (ii) regression providing organ location as initialization followed by a classical segmentation approach, and (iii) a full regression approach. In [110], authors proposed to go for a sequential approach extending their concept of marginal space learning. To delineate organs in CT scans, they add to their cascade of classifiers a last component which is based on a statistical shape model. While

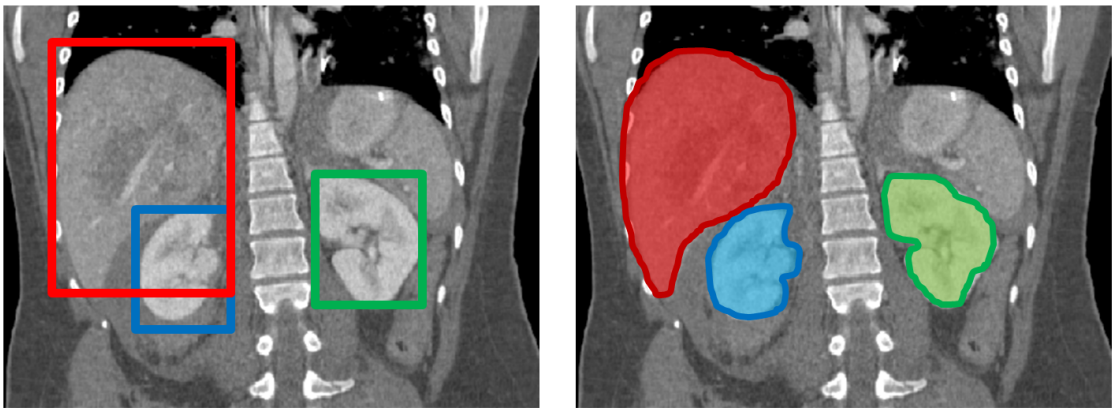

Figure 1.9: Multiple organ localization and segmentation: while localization consists in estimating the position, the size and optionally the orientation of the anatomy of interest, segmentation involves a voxel-wise labeling

they have shown very impressive performance, building such a cascade of classifiers is a computationally intensive learning procedure which requires large training sets.

1.3.3 Image Registration and Tracking

Image registration and tracking are key components in all image analysis or navigation tasks. While image registration can be defined as the task of identifying the geometric transformation that maps the coordinate system of one image to the other, tracking aims at identifying the geometric transformation mapping the model of an object of interest from a frame at timestep t to a frame at timestep $t + 1$. Due to the nature of medical images, both image registration and tracking are very challenging. Indeed, medical images can be multi-modal, *i.e.* they can be acquired with different imaging systems, and they often show low signal to noise ratio.

To improve robustness of registration, learning approaches have been introduced to learn data driven similarity measures. In [53], authors proposed a similarity learning approach for the registration of 3D multi-modal images based on a classification scheme. First, image patch pairs are characterized using local visual features and then they are given to a classifier which decides if these patches are matching or not. Later in [13], the similarity learning is again casted as a binary classification task using an elegant embedding of the input data from two arbitrary feature spaces into the Hamming space. Both approaches showed very promising results for the registration of CT and MR images.

To improve the robustness of tracking, learning approaches can also be used to detect the tool of interest or learn data driven similarity measures. For instance, the tracking of a deformable guide-wire in fluoroscopic sequences is a challenging task due to the low signal to noise ratio of the images and the apparent complex motion of the object of interest. A learning-based tracking approach by detection based on marginal space learning was presented by Barbu *et al.* in [7]. Later, Wang *et al.* proposed in [104] the combination of learning-based detectors and online appearance models.

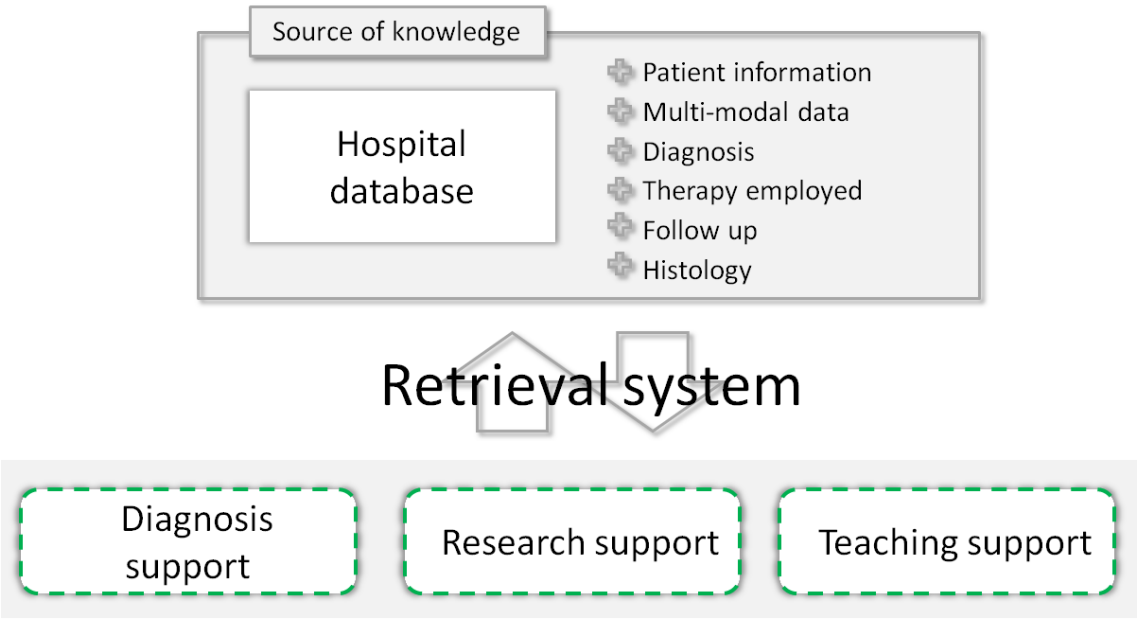

Figure 1.10: Medical content-based retrieval: hospital databases contain a capital of knowledge that can be used for further diagnostics, research or teaching

1.3.4 Medical Image Categorization and Retrieval

During the last decade, major advances in medical imaging have permitted to bring new imaging modalities into the hospitals. Each year, the amount of medical information produced by an hospital never ceases to grow. As illustrated by fig.1.10, this experience accumulated over the years embodies a precious capital of knowledge that can be used for further diagnostics, research or teaching [65]. While a part of this knowledge resides in the medical reports in form of text, rich additional information is contained in the attached digital images produced by different imaging systems. For this reason, alternatives to current text-retrieval methods should be developed, such that research, teaching or even computer aided diagnosis can benefit from the full information contained in the images. Indeed, by retrieving images that show a very similar content as a case of interest, one could also profit from all information attached to these images, such as the diagnosis from different experts, the employed therapies or the different therapy outcomes.

Let us take the example of endomicroscopy. Recently, to assess the risk of gastrointestinal cancer, a new technology has been introduced known as probe-based confocal laser endomicroscopy. With this new imaging system, non-invasive "optical biopsies" can be performed to investigate the nature of the tissues. As endoscopists typically rely on similarity-based reasoning to establish a diagnosis, André *et al.* [3, 4, 5] explored several content-based retrieval approaches to support diagnosis. By retrieving very similar endoscopic videos which were labelled as benign or malignant, they demonstrate state-of-the-art performance for automated diagnosis.

1.4 Our Contributions in this Thesis

To conclude this introduction, the different contributions of this thesis are theoretical as well as application-oriented. Along the methodical chapters 2,3 and the application chapter 4, our theoretical contributions are: (i) to define the random forest framework and related techniques using a partition formalism, (ii) to propose a novel interpretation of random ferns as an intersection of decision stumps instead of the classical definition as constrained tree, and (iii) to introduce a novel ensemble approach we call STARS, that can be seen as an ensemble of multi-decision stumps. By using the partition formalism along this thesis, we can clearly define, compare each approach and analyze their behaviour using numerous toy examples. Thereby, we can hopefully convince the reader that these ensemble techniques are fully transparent models with a well understood behaviour.

As detailed in chapter 4, our contributions in the context of medical applications are: (i) to tackle different medical imaging problems such as organ localization, segmentation, lesion detection and image categorization, and (ii) to design for each application novel and task-specific forest-related approaches. First, we propose to address the problem of multiple organ localization using regression forests and ferns for whole-body MR, building upon the work of Criminisi *et al.* in [21]. We show that forest-related techniques achieve better performance than atlas registration, and this, while being scalable to multiple organs. Moreover, we demonstrate that our novel regression ferns strategy benefits from a very fast training and testing by taking advantage from extreme randomization and the compact fern structure. Beyond organ localization, we introduce a new type of random forest for multiple organ segmentation. Using a joint classification-regression formulation, each voxel is associated to an organ class label as well as its distances to all organ boundaries. By solving this joint objective, the structured output forest learns implicitely spatial regularization directly from the data and provides thereby improved single voxels predictions.

In the context of early diagnosis of Parkinson disease, we contribute to the development of learning-based tools for the support of computer aided diagnosis. Indeed, we introduce a novel paradigm to detect Parkinson-related lesions within the midbrain using 3D transcranial ultrasound. Mimicking human experts, two forest models are designed to capture visual as well as spatial information, the latest being encoded using a novel parametrization that accounts for asymmetric changes of scales and orientation of the midbrain anatomy. On a database of 3D-TCUS volumes from 22 subjects, our approach shows very promising results relatively close to the human inter-rater observability.

In the context of content-based retrieval in medical databases, we address the problem of recognizing the modality of an image based on its visual content only to enable improved image retrieval. To this end, we propose very efficient approaches based on random ferns and STARS clustering to build a dictionary of visual words. Experiments conducted on CT, MR, PET, US and X-ray images taken from the ImageCLEF 2010 database show that our approach is a fast alternative to K-means clustering which provides better performance in terms of accuracy and speed.

Besides our main contributions related to random forests, we also report in the appendix A the novel strategies we proposed for the learning of application-specific similarity

measures in the field of multi-modal registration and tracking. First, we introduced an elegant regression approach based on support vector regression to learn a multi-modal similarity measure. As statistics relating the intensities of two multi-modal images are intuitively constrained by the object of interest and the imaging modalities, we learn a mapping from the subspace described by joint intensity distributions to the target registration error. This yields increased accuracy and robustness compared to classical mutual information on multi-channel MR and SPECT images. In the context of deformable guide-wire tracking, we demonstrate that the robustness of tracking can be also improved by learning a data term directly from fluoroscopic images. Therefore, we adapt our regression framework for tracking, and learn the relationship between features extracted from the original image and the tracking error. To reduce the intrisic dimensionality of this feature space, we first learn a guide-wire motion distribution model. Random samples can then be generated from this distribution, and we can build a training set by computing the visual features and tracking errors corresponding to these deformations. The data term is then learned from this training set using support vector regression. The resulting data term is integrated into a tracking framework based on a second-order MAP-MRF formulation. Experiments conducted on two fluoroscopic sequences show that our approach is a promising alternative for deformable tracking of guide-wires.

CHAPTER

TWO

RANDOM FORESTS

"L'arbre c'est la puissance qui lentement épouse le ciel."
Antoine de Saint-Exupéry

This chapter constitutes the methodic pillar of the thesis, in which we define the decision tree and random forest models. We propose to formalize decision trees as a partitioning approach, as they efficiently subdivide the feature space and model locally the posterior distribution within their leaves. We then discuss how to define task-specific objective functions to optimize their nodes and how to choose appropriate models for the leaf posteriors. Besides all their advantages such as fast learning and prediction, or scalability to large training sets, decision trees have some limitations and tend to suffer from overfitting. To reach an increased generalization, they can be used in an ensemble fashion to constitute a so-called random forest. We will show how to create such ensembles of independent trees by injecting randomness during the training phase. Finally, we discuss how to derive random forests for different learning tasks such as classification, regression and clustering.

2.1 Mathematical Notations

- $\mathcal{X}$: **input** feature space, where $\mathcal{X} \subset \mathbb{R}^D$
- $\mathcal{Y}$: **output** space, where $\mathcal{Y} \subset \mathbb{R}^{D'}$
- $\mathbb{B}$: Boolean set, $\mathbb{B} = \{0,\ 1\}$
- $\mathbf{X}$: feature vector containing one **observation** instance, with $\mathbf{X} \in \mathcal{X}$
- $\mathbf{Y}$: output vector containing one **prediction** instance, with $\mathbf{Y} \in \mathcal{Y}$
- $\mathbf{F}$: decision tree, **directed acyclic graph** of binary decisions, $\mathbf{F} = \{\mathbf{N}, \mathbf{E}\}$
- $\mathbf{N}$: set of node, each node N_l encoding a decision function f_l
- $\mathbf{E}$: set of directed edges, where each edge represents a directed link between two nodes
- f: decision function, defined as $f : \mathcal{X} \to \mathbb{B}$
- Γ: set of decision function candidates
- $\mathbf{v}$: linear projection, defined as $\mathbf{v} : \mathcal{X} \to \mathbb{R}$
- τ: threshold, defined as $\tau \in \mathbb{R}$
- $\mathcal{P}$: partition of the feature space $\mathcal{X}$, ensemble of Z cells $\mathcal{P} = \bigcup_{z=1}^{Z} \mathcal{C}^{(z)}$
- $\mathcal{C}$: cell of a partition
- $\mathcal{F}$: random forest, ensemble of T decision trees $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^{T}$
- $\mathbf{C}$: set of cells from the different partitions of a forest, $\mathbf{C} = \left\{\mathcal{C}_1^{(z_1)}, \cdots, \mathcal{C}_t^{(z_t)}, \cdots, \mathcal{C}_T^{(z_T)}\right\}$
- $\mathcal{K}$: set of K clusters $\mathcal{K} = \{\mathbf{K}_k\}_{k=1}^{K}$
- Δ: objective function

2.2 Decision Trees and Random Forests Models

Aiming at building knowledge from a set of observations, decision trees incarnate a simple tool based on following strategy: **partition** observations by using a set of simple **decisions** in a **hierarchical** fashion. Considering our introductory example of ant specimens, a tree can be easily designed to partition the different observations into 4 casts. As illustrated by fig.2.1, this can be achieved by using only 3 different decisions performed on morphological characteristics which are: (1) does the specimen have reproduction organs, (2) does it possess oversized mandibles, and (3) does it have wings. After applying the first decision on all observations, the specimens corresponding to a negative or positive answer are sent respectively to the left or right branch of the tree. Then, while on the left the decision (2) permits to split the specimens into two groups that can be indentified as "Workers" and "Soldier", on the right, the decision (3) subdivides observations into the casts "Queen" and "Princess".

In a nutshell, decision trees are hierarchical learners consisting of an ensemble of simple (binary) decisions. Back in 1984, Breiman *et al.* formalized in [12] the tree model for classification and regression tasks. Afterward, decision trees became very popular and were widely used in numerous machine learning applications. One reason for their success may be that they benefit from many advantages: they are very intuitive, they are fast and scalable to very large datasets, and they can be formulated in a probabilistic fashion to take uncertainty into account. Over the years, many learning algorithms have been proposed and the most popular is probably the C4.5 of Quinlan [82]. However, learning an optimal decision tree is known to be NP-complete, and it can yield over complex models which are not able to generalize well *e.g.* that suffer from overfitting on the training set.

Inspired by the emergence of ensemble learning, Ho proposed in [41, 42] to construct an ensemble of "weak" decision trees, namely random forests, instead of aiming at optimizing a single complex tree. In these works, authors propose to inject randomization in the learning process in order to create decorrelated trees. By averaging their predictions, authors demonstrate that random forests achieve greater generalization and thereby superior accuracy. In [11], Breiman proposes an alternative approach for injecting randomness in the learning phase. Known as "*bagging*", for bootstrap aggregating, this technique consists of training each independent tree with a random subset of the training set.

Since then, random forests have been successfully used in many applications, mostly formulated as classification tasks. However, they can also be applied to solve regression, clustering, density estimation, semi-supervised learning or manifold learning tasks as demonstrated in [20]. In the present thesis, we will show how to use random forests in different medical applications formulated as classification, regression or clustering tasks.

In the following, we start by defining the decision tree as a directed acyclic graph, and to formalize it as a partitioning approach. Afterward, we detail the node/leaf models and show different example of splitting functions. Finally, we discuss how independent trees can be combined together into a strong learner called random forest.

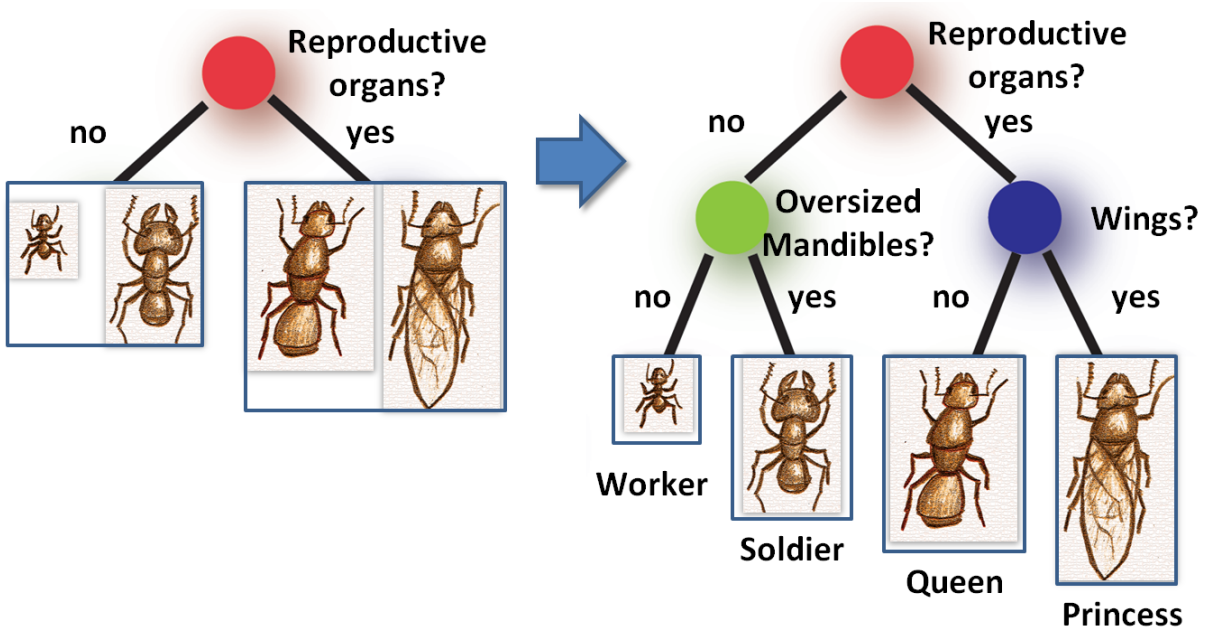

Figure 2.1: Decision tree: its goal is to partition observations by using simple decisions in a hierarchical fashion

2.2.1 Decision Tree

Considering an input feature space $\mathcal{X} \subset \mathbb{R}^D$ and an output space $\mathcal{Y} \subset \mathbb{R}^{D'}$, our goal is to learn a model which is able to perform predictions in $\mathcal{Y}$ given an observation in $\mathcal{X}$. In a probabilistic framework, we can formulate this task as a maximum a posteriori problem:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y} \in \mathcal{Y}} P(\mathbf{Y}|\mathbf{X}) \tag{2.1}$$

Given a training set $\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, we aim at learning the posterior $P(\mathbf{Y}|\mathbf{X})$. Finding a suitable model for this posterior and learning it over the full feature space $\mathcal{X}$ is a very difficult task. To leverage this problem, a decision tree follows a "**divide**" and "**conquer**" strategy: (1) it creates a **partition** over the input feature space using a set of decisions, and (2) it **estimates** $P(\mathbf{Y}|\mathbf{X})$ in each "cell" of this space.

2.2.1.1 Tree Model

Decision trees are based on the following idea: perform predictions using a sequence of simple decisions. In fact, a decision tree model consists of an ensemble of (binary) decisions organized in a hierarchical fashion. First of all, let us analyse briefly what the term "hierarchical" means and what it implies. Referring to the common sense, a hierarchy is an **ordered** structure. Hence, a decision tree $\mathbf{F}$ can be formally defined as a **directed acyclic graph**, composed of a set of **nodes N** and a set of **directed edges E**. Each node encodes a (binary) decision, and is connected by a directed edge to at most one **parent** node from the superior level and at least two **children** nodes from the lower level. "Directed" implies that: (1) the tree can be traversed only using a descending path, *i.e.* the parent-to-children direction, and (2) that nodes from different levels are

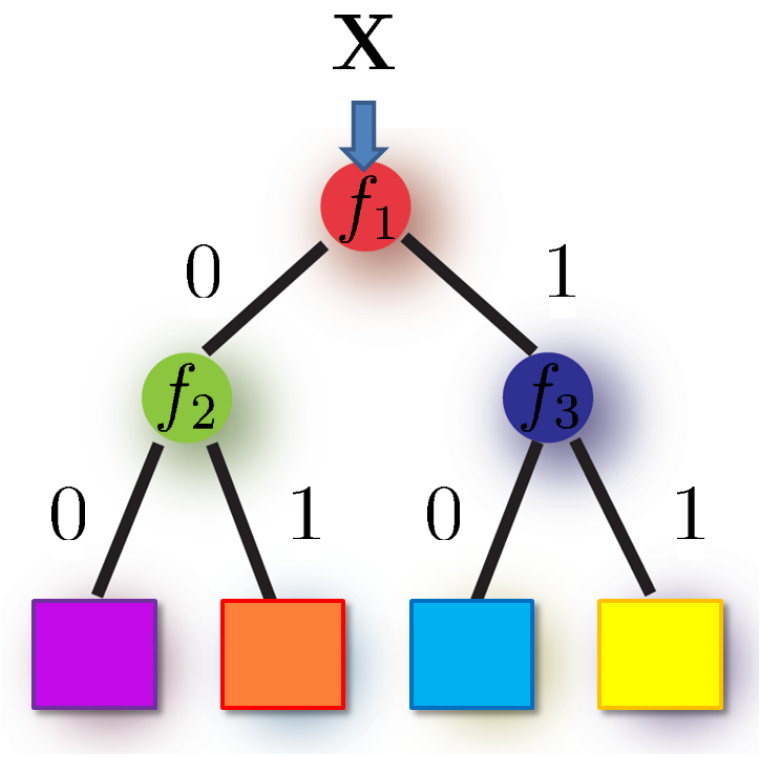

Figure 2.2: Decision tree: a decision tree is a directed acyclic graph, where each node is equipped with a decision function

not intervertible. "Acyclic" means that there is no cycles within a tree model. While the node at the top of a tree is called **root**, nodes at the bottom are called **leaves**. An observation can traverse the tree downward, following a **unique** path which is determined by decisions taken at each traversed node, until it reaches a leaf as illustrated on fig.2.2. During the learning phase, the data that reached a given leaf is used to model the posterior distribution "locally". During the test phase, these posterior distributions permit to make predictions about new unseen observations reaching a given leaf. Note that in the present thesis, we will focus mostly on binary decision trees.

2.2.1.2 "Divide": the Node Model

To perform a binary decision, a node N_l from the set $\mathbf{N}$ of a tree is equipped with a so-called *splitting* function f_l whose role is to split incoming observations denoted by $\mathcal{S}_l$ into two subsets $\mathcal{S}_l^{\text{left}}$ and $\mathcal{S}_l^{\text{right}}$. These two subsets are disjoints, *i.e.* $\mathcal{S}_l = \mathcal{S}_l^{\text{left}} \cup \mathcal{S}_l^{\text{right}}$ and $\mathcal{S}_l^{\text{left}} \cap \mathcal{S}_l^{\text{right}} = \emptyset$, and they are sent respectively to the left and the right child of N_l. The splitting function f_l is defined as follows:

$$\begin{cases} f_l : \mathcal{X} \to \mathbb{B} \\ f_l(\mathbf{X}) = 0, \ \mathbf{X} \text{ is sent to the left} \\ f_l(\mathbf{X}) = 1, \ \mathbf{X} \text{ is sent to the right} \end{cases} \tag{2.2}$$

As reported in [20], there are many possibilities for the class of decision functions.

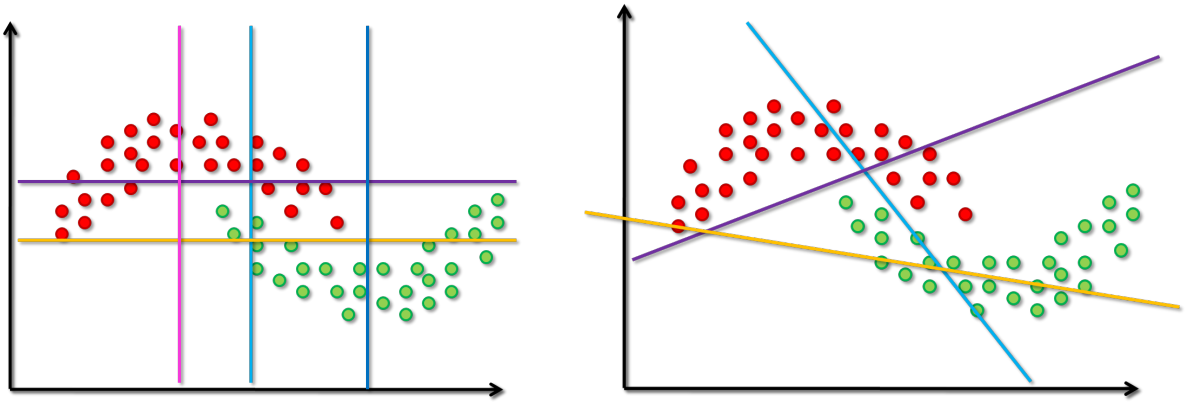

Figure 2.3: Classes of splitting function: the mostly used splitting functions are linear projections followed by a thresholding operation

However, the most common choice is the class of linear projection coupled with a threshold operation:

$$f_l(\mathbf{X}) = (\mathbf{X} \cdot \mathbf{v}_l \geq \tau_l) \tag{2.3}$$

where $\dim(\mathbf{v}_l) = \dim(\mathcal{X})$ and $\tau_l \in \mathbb{R}$. If $\mathbf{v}_l$ has only non-zero entries, then the splitting function is a hyperplane which takes into account all input features as illustrated by fig.2.3 on the right. However, if $\mathbf{v}_l$ is sparse, f_l performs splitting using only a subset of features. In the extreme case where $\mathbf{v}_l$ has only one non-zero component, then splitting is performed only based on one feature *i.e.* along one dimension of $\mathcal{X}$ as shown on fig.2.3 on the left. More complex decision functions such as non-linear can be also used, however the tree philosophy encourages the choice of simple functions which can be efficiently computed and optimized.

Tree learning and node optimization: As shown by the pseudo-code in alg.1, tree learning can be basically defined as an iterative node optimization and splitting. Indeed, at a given node, first a good splitting function has to be chosen and then the training data is split and sent towards the left and the right child. Depending on the chosen class of functions, several parameters need to be determined. In the case of linear projections coupled with simple thresholding, the degree of freedom is $D+1 = \dim(\mathcal{X})+1$. To avoid a complex optimization procedure in a high-dimensional search space, node optimization follows a greedy strategy. If we consider the node N_l, a set of NTry candidates functions $\Gamma_l = \left\{f_l^{(i)}\right\}_{i=1}^{NTry}$ is generated and evaluated given the incoming training points $\mathcal{S}_l$ and a predefined objective function Δ. The best candidate is then the function which maximizes Δ:

$$f_l^* = \mathbf{argmax}_{f_l \in \Gamma_l} \Delta(\mathcal{S}_l, \mathcal{S}_l^{\text{left}}, \mathcal{S}_l^{\text{right}}) \tag{2.4}$$

During the training phase, decision functions at each node are optimized to iteratively split the training until a stopping criteria has been reached.

Algorithm 1: Tree Training: Pseudocode example

\\\\\\\\\\\\\\\\\\\\\\\\\ **Main function** \\\\\\\\\\\\\\\\\\\\\\\\\
Training set: $\mathcal{S} = \left\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\right\},\ n \in \{1, \cdots, N\}$
Tree object: **F**
Parameters: MaxDepth, MinPopPerLeaf, NTry
perform iterative splitting, starting with the root node
depth = 0;
splitNode(**F**.N_0, $\mathcal{S}$, depth, MaxDepth, MinPopPerLeaf, NTry);
Output: trained tree **F**

\\\\\\\\\\\\\\\\\\\\\\\\\ **Iterative splitting** \\\\\\\\\\\\\\\\\\\\\\\\\
function **splitNode**(N_l, $\mathcal{S}_l$, depth, MaxDepth, MinPopPerLeaf, NTry)
Model posterior and initialize node as leaf
N_l.Posterior $\leftarrow$ **estimatePosteriorDistribution**($\mathcal{S}_l$);
N_l.isLeaf = TRUE
If max depth is not reached, try to split
if depth < MaxDepth **then**
  loop over the split candidates
  $\Delta_{\text{best}} = 0$;
  for (**int** $i = 1,\ i \leq \text{NTry},\ i++$) **do**
    $f_l \leftarrow$ **generateSplittingFunctionCandidate**;
    split the data
    $\left(\mathcal{S}_l^{\text{left}}, \mathcal{S}_l^{\text{right}}\right) \leftarrow$ **applySplittingFunction**($\mathcal{S}_l, f_l$);
    If enough points left and right, evaluate the split quality
    if $\left(|\mathcal{S}_l^{\text{left}}| \geq \text{MinPopPerLeaf}\ \&\ |\mathcal{S}_l^{\text{right}}| \geq \text{MinPopPerLeaf}\right)$ **then**
      $\Delta =$ **computeObjectiveFunction**($\mathcal{S}_l, \mathcal{S}_l^{\text{left}}, \mathcal{S}_l^{\text{right}}$);
      if ($\Delta > \Delta_{\text{best}}$) **then**
        N_l.splitFunc = f_l;
        $\Delta_{\text{best}} = \Delta$;
        $\mathcal{S}_{\text{best}}^{\text{left}} = \mathcal{S}_l^{\text{left}}$;
        $\mathcal{S}_{\text{best}}^{\text{right}} = \mathcal{S}_l^{\text{right}}$;
      end if
    end if
  end for
  Check whether we found a good split and iterate splitting if yes
  if $\Delta > 0$ **then**
    N_l.isLeaf = FALSE
    depth = depth + 1;
    splitNode(N_l.leftChild, $\mathcal{S}_{\text{best}}^{\text{left}}$, depth, MaxDepth, MinPopPerLeaf, NTry);
    splitNode(N_l.rightChild, $\mathcal{S}_{\text{best}}^{\text{right}}$, depth, MaxDepth, MinPopPerLeaf, NTry);
  else
    return;
  end if
end if

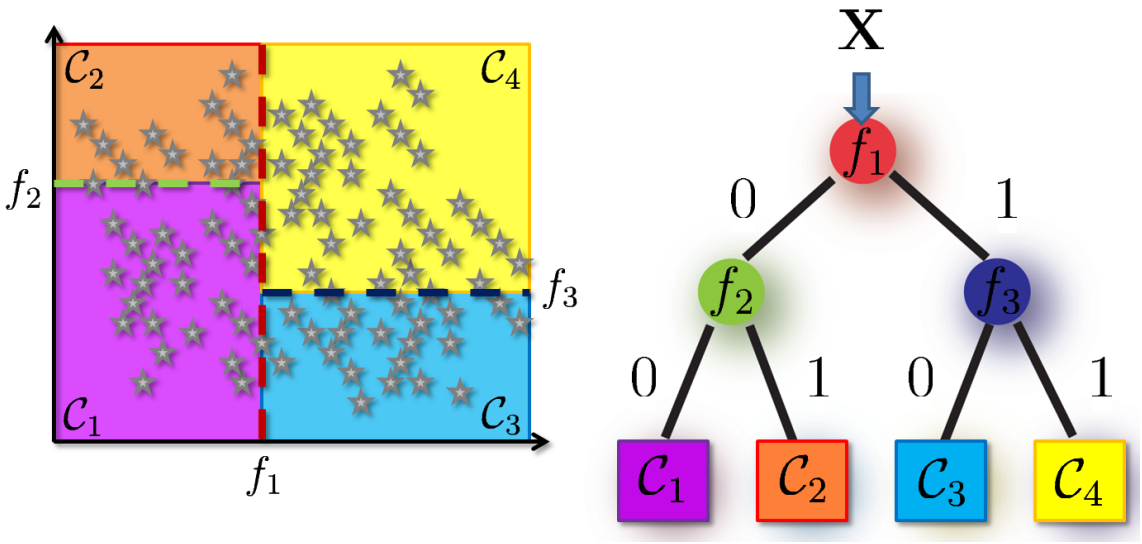

Figure 2.4: Partitioning approach: a decision tree creates a partition of the feature space, and each leaf corresponds to a "cell" of this space

2.2.1.3 "Conquer": the Leaf Model and the Partition Formalism

Once the bottom of the tree has been reached, iterative splitting of the training data stops and the current node becomes a leaf node. Three common stopping criteria can be defined: (1) maximal tree depth, (2) minimum population per leaf, and (3) minimum variation of the objective function Δ. The first criterion considers just the depth of the hierarchy, and once a certain depth has been reached, then the iterative splitting stops. The second criterion is based on the number of training instances arriving in a node, and if the population of training points is below a certain threshold, the splitting stops. The last criterion concerns the objective function which is optimized. If its variation is below a certain threshold, then it is considered that there is no additional information gained after splitting the training instances.

While the role of "internal" nodes is to split and send observations downward the tree, the role of the leaves is to model the posterior distribution given a subset of the training set. As these decisions are taken in the input feature space, all training points arriving in a leaf are consistent in $\mathcal{X}$. Hence, each leaf corresponds to a part or "cell" of the feature space as illustrated by fig.2.4, and the ensemble of leaves of a decision tree builds a partition $\mathcal{P}$ over $\mathcal{X}$. In the remaining of this thesis, we will consider the terms "leaf" and "cell" as synonyms. Let us define this partition as an ensemble of cells $\mathcal{P} = \bigcup_{z=1}^{Z} \mathcal{C}^{(z)}$, where each $\mathcal{C}^{(z)}$ corresponds to a leaf of the decision tree. Note that the $\mathcal{C}^{(z)}$ cover the full feature space $\mathcal{X}$ and have no overlap. Moreover, we emphasize again the fact that trees are directed acyclic graphs, and that interverting nodes would results in a totally different partition of $\mathcal{X}$. Furthermore, per construction, all cells are populated during the training, and posterior distributions can be modelled in each cell as:

$$P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}^{(z)}, \mathcal{P}) \tag{2.5}$$

i.e. by using the subset of the training set that reaches the leaf/cell $\mathcal{C}^{(z)}$. If the partition

$\mathcal{P}$ counts only a few number of cells, then these posteriors might suffer from high uncertainty. On the other side, if their number is very high, each cell will include only a few training points, leading then to overfitting.

Tree prediction: Once a decision tree has been trained, prediction for a new unseen observation $\mathbf{X}$ can be easily performed as detailed in the pseudo-code in alg.2. Depending on the results of the different decision functions, $\mathbf{X}$ is sent downward the tree, following a path which is unique and leads to a leaf $\mathcal{C}^{(z)}$. Hence, at test time, a tree $\mathbf{F}$ can be seen as a *surjective* function taking as input an observation and returning a cell:

$$\begin{cases} \mathbf{F} : \mathcal{X} \rightarrow \{\mathcal{C}^{(1)}, \cdots, \mathcal{C}^{(z)}, \cdots, \mathcal{C}^{(Z)}\} \\ \mathbf{F}(\mathbf{X}) = \mathcal{C}^{(z)} \end{cases} \tag{2.6}$$

The posterior model stored in this leaf permits to perform a prediction by using for instance a maximum a posteriori:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}}\, P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}^{(z)}, \mathcal{P}) \tag{2.7}$$

Algorithm 2: Tree Prediction: Pseudocode example

```
1: Observation: X
2: Tree object: F
3: leafReached = FALSE;
4: currentNode = F.N_0
5: while (leafReached == FALSE) do
6:    \\perform binary decision
7:    val = currentNode.splitFunc(X)
8:    \\depending on the results, go to left or right child
9:    if (val == 1) then
10:      currentNode = currentNode.rightChild
11:   else
12:      currentNode = currentNode.leftChild
13:   end if
14:   \\check whether the observation reached a leaf
15:   if (currentNode.isLeaf == TRUE) then
16:      leafReached == TRUE
17:      Posterior = currentNode.Posterior
18:   end if
19: end while
20: Output: Posterior
```

To conclude this section, decision trees can approximate any arbitrary function if enough training data is available. On one side, decision trees can be considered as non-parametric models since their size depends on the amount of training data. On the other side, a parametric model is learned from the data in each cell. As discussed in

the introduction of this chapter, training an optimal tree is a NP-complete problem, and decision trees are prone to overfitting. Inspired by the trend of ensemble learning, we will show in the next section how to replace a single decision tree by an ensemble of decorrelated trees to achieve greater generalization.

2.2.2 Random Forests

A random forest $\mathcal{F}$ is basically an ensemble of T independent decisions trees $\mathcal{F} = \{\mathbf{F}_1, \cdots, \mathbf{F}_t, \cdots, \mathbf{F}_T\}$. As demonstrated by Breiman in [11], replacing a single tree by an ensemble of decorrelated trees provides very good generalization. During the learning phase, randomness can be injected to achieve independence between trees contructed from the same training set. In the following section, we will explain several tree randomization approaches.

2.2.2.1 Forest training and tree randomization

To build decorrelated or independent trees based on a unique training set, several randomization approaches have been proposed. In [11], Breiman introduced the concept of *bagging* which comes from the combination of the terms "bootstrap" and "aggregating". Given a training set $\mathcal{S} = \{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\}_{n=1}^{N}$, a bootstrap is basically a subset $\mathcal{S}_t$ of the full training set, in which element has been randomly sampled using a uniform distribution, and this, with or without replacement. As illustrated by fig.2.5, each tree $\mathbf{F}_t$ of the ensemble is then trained using a different bootstrap $\mathcal{S}_t$. Finally, predictions from all individual trees are aggregated together using averaging.

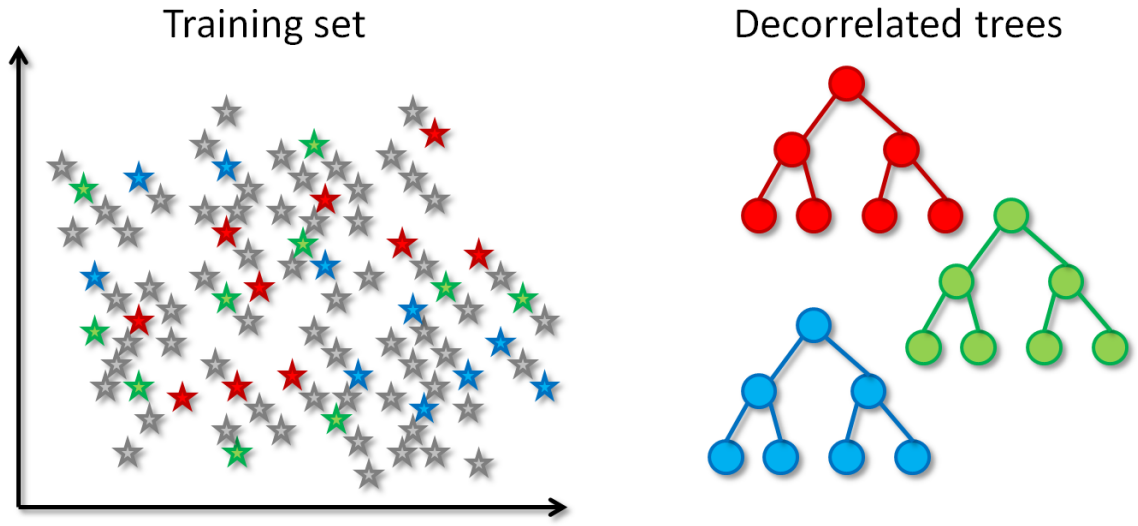

Figure 2.5: Bagging *e.g.* "bootstrap aggregating": each tree is trained on a different random subset of the training set

As reported in [36], randomization can be also injected in the node optimization. Indeed, as this phase relies on a greedy strategy, a set of splitting functions candidates is generated and the best is then chosen according to a predefined objective function. Obviously, randomness can be injected in the generation of function candidates. Let us take the example of linear projections followed by a thresholding operation. First, the projection vector $\mathbf{v}$ can be randomly drawn using any kind of distributions. This encourage the trees to select different type of features and to weight them differently. Furthermore, the choice of threshold τ can be also randomized instead of optimizing it or taking the mean/median of the projected values.

The impact of injecting randomness in the tree training has several advantages: first, increasing the degree of randomness decreases the correlation between the different trees and provide thereby greater generalization, second it enables implicit feature selection if $\mathbf{v}$ is constrained to be sparse and third, it permits to gain independence from the training set, *i.e.* to gain robustness to noisy data.

2.2.2.2 Forest Parameters

Random forests offer a very flexible framework with a lot of freedom for designing task-specific objective functions, different classes of splitting functions or posterior models. Moreover, they possess only a few hyperparameters which influence has been exhaustively studied, as in [20], and is now well understood. The two most important degrees of freedom are: (1) the **number of trees** and (2), the **tree depth**. As illustrated by fig.2.6, increasing the number of trees permits to average out noisy predictions, and thus corresponds in a monotonic decrease of the prediction error. The maximal allowed depth of the tree is a crucial parameter that needs to be optimized as it directly impacts the generalization ability of each tree. Indeed, while a short tree will not be very confident in its prediction because its leaves still contains a lot of heterogenous data, a very deep tree will have very few training data in its leaves to compute reliable statistics. Therefore, the tree start to explain too well the training data, *e.g.* by fitting noisy features, and will suffer from poor generalization. For this reason, the prediction error curve decreases with the tree depth until it reaches a minimum and then increases again. This minimum corresponds to the optimal tree depth, providing a good modeling of the observations and a great generalization.

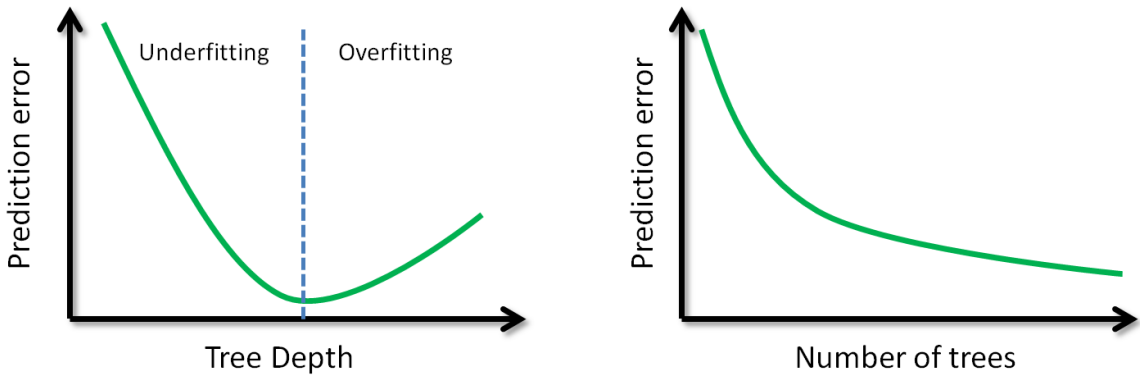

Figure 2.6: Forest parameters: Tree depth and number of trees are the two most important parameters. While increasing the number of trees correspond to a decreasing in the prediction error, tree depth needs to be carefully tuned as it controls the generalization ability of the forests.

2.2.2.3 Forests prediction

Let us consider a random forest of T trees $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^{T}$, each tree $\mathbf{F}_t$ yielding a partition $\mathcal{P}_t$ of the feature space $\mathcal{X}$. As each individual tree can be seen as a surjective function associating an observation $\mathbf{X} \in \mathcal{X}$ to a cell $\mathcal{C}_t^{(z_t)}$ of partition $\mathcal{P}_t$, the whole forest is a

function which associates $\mathbf{X}$ to an ensemble of cells:

$$\mathcal{F}(\mathbf{X}) = \left\{\mathcal{C}_1^{(z_1)}, \cdots, \mathcal{C}_t^{(z_t)}, \cdots, \mathcal{C}_T^{(z_T)}\right\} \tag{2.8}$$

If we consider that each $\mathcal{P}_t$ is equiprobable, the forest prediction can be simply computed by averaging the tree posteriors:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}_t^{(z_t)}, \mathcal{P}_t) \tag{2.9}$$

Averaging is commonly used as it is a good compromise between giving more weight to the most confident tree and reducing noisy contributions [20]. Nevertheless, other aggregation approaches are possible. For instance, the contributions from each individual tree can be ranked according to their confidence, and averaging can be performed by using only a fraction of the most confident predictions. Another alternative would be to perform a weighted averaging of all contributions according to their confidence.

In the following sections, we will detail how to instantiate random forests for classification, regression and clustering tasks.

2.3 Classification Forests

In computer vision or machine learning, random forests have been mainly applied for classification tasks. Besides their advantage of having great generalization ability, being scalable to large datasets, and benefitting of fast training and predictions, they are particularly well adapted to multi-class problems as they are inherently multi-class, and provide probabilistic output. In the following section, we first start with a probabilistic formulation of the multi-class classification task. We then explain how class posterior distributions can be easily modelled in the leaves of each random tree. Afterward, we detail the node optimization procedure and give several examples of objective functions. Thereafter, we will discuss the different approaches for combining tree predictions, and we will show how to handle cases where classes are unbalanced, *i.e.* where classes have different cardinality in the training set. Finally, we will conclude the section with a few classification toy examples.

2.3.1 Problem Statement

In a classification task, we consider an input feature space $\mathcal{X} \subset \mathbb{R}^D$ and an output space $\mathcal{Y} \subset \mathbb{R}$ which is a finite set of K discrete values $\mathcal{Y} = \{y_1, \cdots, y_k, \cdots, y_K\}$ representing the different classes. Our goal is to model the posterior probability distribution $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{X} \in \mathcal{X}$ and $\mathbf{Y} \in \mathcal{Y}$. Hence, given any unseen observation in $\mathcal{X}$, we are able to predict its label using the maximum a posteriori:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y} \in \mathcal{Y}} \, P(\mathbf{Y}|\mathbf{X}) \tag{2.10}$$

Given a training set $\{(\mathbf{X}^{(n)}, Y^{(n)})\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, each tree of a forest $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^{T}$ permits to build a partition $\mathcal{P}_t$ over the input feature space $\mathcal{X}$. Considering the tree model presented in the previous section, two components needs to be instantiated for the classification task: (1) the leaf posterior and (2), the objective function.

2.3.2 Class Posteriors

Let us consider the partition $\mathcal{P}_t = \{\mathcal{C}_t^{(z_t)}\}_{z_t=1}^{Z_t}$ built by the random tree $\mathbf{F}_t$. As illustrated by fig.2.7, class posteriors can be simply approximated in each cell $\mathcal{C}_t^{(z_t)}$ of $\mathcal{P}_t$ as follows:

$$P(y_k|\mathbf{X} \in \mathcal{C}_t^{(z_t)}, \mathcal{P}_t) = \frac{|\left\{\mathbf{X}^{(n)} \in \mathcal{C}_t^{(z_t)}, Y^{(n)} = y_k\right\}|}{|\left\{\mathbf{X}^{(n)} \in \mathcal{C}_t^{(z_t)}\right\}|} \tag{2.11}$$

During the training of the tree, the goal will be to split recursively the training data to reduce the class uncertainty linked to these class posteriors, *i.e.* by creating leaves that are class-consistent. We show in the next section how to define the objective function for node optimization.

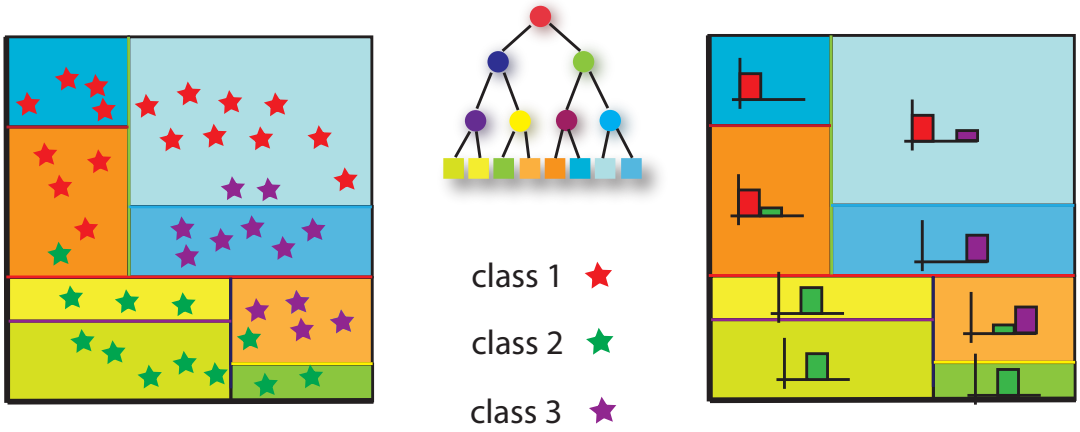

Figure 2.7: Classification forest: each tree $\mathbf{F}_t$ builds a partition $\mathcal{P}_t$ over the feature space and class posteriors can be easily approximated in each cell of $\mathcal{P}_t$

2.3.3 Classification Objective Function

At each node N_l of the tree $\mathbf{F}_t$, a splitting function f_l permits to split the subset $\mathcal{S}_l$ of the training set arriving in this node. As detailed in the previous section, the goal of node optimization is to find the best splitting function according to a predefined objective function. In classification tasks, several objective functions have been proposed that mostly aim at reducing the class uncertainty. In the following, we will define the most popular which is the **Information Gain** and a variant based on the **Gini impurity**.

Information gain measures the difference between the class uncertainty before and after the splitting. A common measure of uncertainty is the so-called Shannon's entropy which is defined for discrete random variables as follows:

$$H(\mathcal{S}_l) = -\sum_{k=1}^{K} P(y_k|\mathcal{S}_l) \log\left(P(y_k|\mathcal{S}_l)\right) \tag{2.12}$$

After splitting $\mathcal{S}_l$ into two subsets $\mathcal{S}_l^{\text{left}}$ and $\mathcal{S}_l^{\text{right}}$ that are respectively sent to the left and right child nodes, the reduction of uncertainty can be measured using the information gain Δ:

$$\Delta = H(\mathcal{S}_l) - w_{\text{left}} H(\mathcal{S}_l^{\text{left}}) - w_{\text{right}} H(\mathcal{S}_l^{\text{right}}) \tag{2.13}$$

where $w_{\text{left}} = |\mathcal{S}_l|/|\mathcal{S}_l^{\text{left}}|$ and $w_{\text{right}} = |\mathcal{S}_l|/|\mathcal{S}_l^{\text{right}}|$. Another variant to Shannon's entropy that can be used within the information gain is the Gini impurity defined as:

$$G(\mathcal{S}_l) = \sum_{k=1}^{K} P(y_k|\mathcal{S}_l)(1 - P(y_k|\mathcal{S}_l)) \tag{2.14}$$

As shown on fig.2.8, both Shannon's entropy and Gini impurity have a similar behaviour and reach their maximum when the class posterior is uniform *i.e.* when $P(y_k|\mathcal{S}_l) = \frac{1}{K}$. Therefore, one can expect similar results by using one of these two functions. There

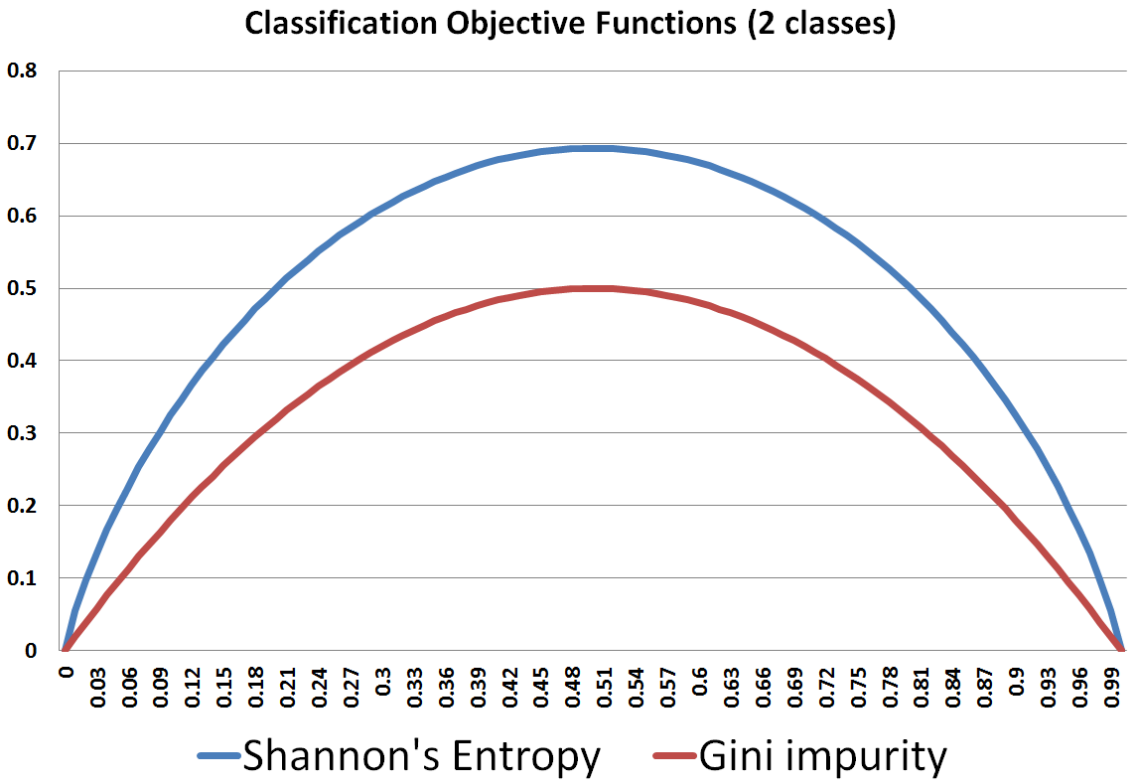

Figure 2.8: Classification objective functions: Information gain and Gini impurity are illustrated in this plot for a binary classification task. The X-axis represents the probability of one class and on the Y-axis the value of both objective functions. Both can be seen as measures of class uncertainty and reach their maximum for 0.5.

are also many other approaches available that propose to minimize the so-called *Out-Of-Bag* (OOB) error. One part of $\mathcal{S}_l$ is used to model the posteriors and the other part to compute the OOB error based on a specific loss function. In this thesis, we focus on node optimization using information gain.

As detailed in the previous section, tree training follows a greedy optimization strategy. At each node, a set of splitting function candidates are generated randomly and the best candidate is chosen as the one maximizing Δ:

$$f_l^* = \mathbf{argmax}_{f_l \in \Gamma_l} \Delta(\mathcal{S}_l, \mathcal{S}_l^{\text{left}}, \mathcal{S}_l^{\text{right}}) \tag{2.15}$$

Intuitively, optimizing these objective functions leads to leaf clusters of data points that are similar in the feature space $\mathcal{X}$ and that belong to the same class.

2.3.4 Forest Prediction

Once the training phase accomplished, predictions can be performed for new incoming observations by sending them through all trees of the forest and combining tree posteriors. As discussed in the previous section, a common approach to compute the forest prediction

$\mathbf{Y}$ for an observation $\mathbf{X}$ is to average the tree posteriors:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} P(\mathbf{Y}|\mathbf{X}, \mathcal{P}_t) \tag{2.16}$$

where $\mathcal{P}_t$ is the partition induced by tree $\mathbf{F}_t$, and then to use the maximum a posteriori:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}\in\mathcal{Y}}\, P(\mathbf{Y}|\mathbf{X}) \tag{2.17}$$

In classification tasks, this approach can be seen as combining trees' smooth labelling outputs. Alternatively, trees' hard labelling outputs can be combined together, and this approach is called **major voting**. In this case, maximum a posteriori is first performed on each tree:

$$\hat{\mathbf{Y}}_t = \mathbf{argmax}_{\mathbf{Y}\in\mathcal{Y}}\, P(\mathbf{Y}|\mathbf{X}, \mathcal{P}_t) \tag{2.18}$$

and the forest finally predicts the label which counts most of the tree "votes" $\left\{\hat{\mathbf{Y}}_t\right\}_t^T$:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}\in\mathcal{Y}} \left(\sum_{t=1}^{T} \left[\hat{\mathbf{Y}}_t = \mathbf{Y}\right]\right) \tag{2.19}$$

where $[\cdot = \cdot]$ is a boolean function outputting a 1 if the proposition is true and a 0 if not. Clearly, the disadvantage of using major voting is that the probabilistic nature of forest outputs *e.g.* class confidence for the labelling decision is lost. Moreover, all votes have the same influence regarding the final prediction, even if individual votes come from tree leaves having very different confidence. While a random forest achieves its great generalization by combining outputs of an ensemble of randomized trees, individual trees show very different performances depending on the part of the feature space the new observation $\mathbf{X}$ is falling. Instead of using a simple averaging of the tree posteriors, one can think of ranking first the tree posteriors based on a measure of uncertainty such as Shannon's entropy, and perform averaging using only the best tree predictions.

2.3.5 Class Balancing Problem

While in classification toy examples, each classes are constructed so that they have similar number of training points, real world applications very often suffer from unbalanced classes. Clearly, if the number of training points for each class is very different, the computation of the posterior using eq.2.11 becomes biased towards the bigger class. To prevent this kind of bias during the training, there are two possible solutions: (1) use a balanced bootstrap of the training set, or (2) use a class normalization when computing the posteriors.

In the first solution, balanced bootstraps of the training set $\left\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\right\}_{n=1}^{N}$ are generated, *i.e.* M observations are sampled from each class with:

$$M < \inf_{\mathbf{Y}\in\mathcal{Y}} |\left\{\mathbf{X}^{(n)},\ \mathbf{Y}^{(n)} = \mathbf{Y}\right\}| \tag{2.20}$$

Each individual tree is then trained using such a bootstrap and posteriors are computed normally using eq.2.11. While this solution seems very simple, it is not applicable in

typical detection cases, where for instance one aims at detecting anomalies that constitutes a very small class compared to the background class. Indeed, one will never be able to cover the rich variability of the background class by learning only from very small subsets. To overcome this problem, class priors can be computed beforehand from the full training set using:

$$P(\mathbf{Y}) = \frac{|\left\{\mathbf{X}^{(n)},\ \mathbf{Y}^{(n)} = \mathbf{Y}\right\}|}{N} \tag{2.21}$$

Hence, the class posterior can be computed in each leaf $\mathcal{C}^{(z)}$ integrating these priors:

$$P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}_t^{(z_t)}, \mathcal{P}_t) = \frac{1}{Q}\frac{1}{P(\mathbf{Y})}\frac{|\left\{\mathbf{X}^{(n)} \in \mathcal{C}_t^{(z_t)}, Y^{(n)} = \mathbf{Y}\right\}|}{|\left\{\mathbf{X}^{(n)} \in \mathcal{C}_t^{(z_t)}\right\}|} \tag{2.22}$$

where Q is a normalization constant. This solution permits to reliably remove the bias introduced by unbalanced classes and will be the approach we use in all our applications.

2.3.6 A Few Toy Examples

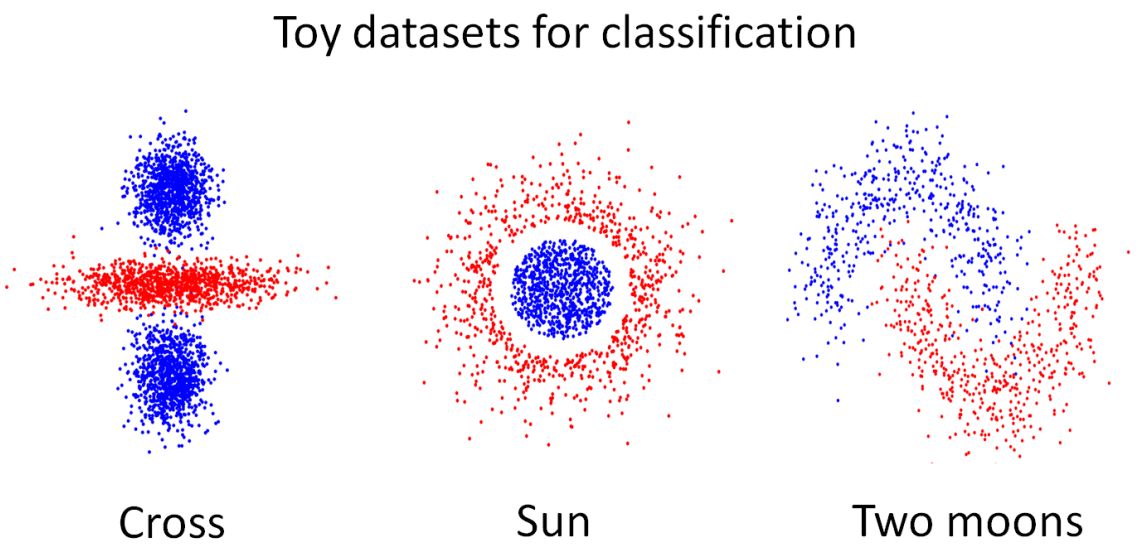

Figure 2.9: Classification toy examples: we propose to study the forests behaviour on these 3 datasets

In this part, we propose to illustrate how forests approximate class posterior distributions, and to show the influence of the main forests parameters, *e.g.* the tree depth and the number of trees. Therefore, we will use 3 toy examples using the "cross", "sun" and "two moons" datasets (see fig.2.9). These three datasets are binary classification problems where classes are represented by blue or red points in a two dimensional feature space. While these classification tasks may seem easy, they reveal a few challenges: classes are not linearly separable, they may consist of separated clusters, or even be concave, and have a few noisy points which overlap on the other class.

Let us start with a single tree, each node splitting function selecting a random dimension and a random threshold chosen within the interval defined by the features of the data points. Hence, these functions correspond to simple axis-aligned splits. We fix the number of function candidates to 10 per node and we vary the depth of the tree between 5 and 15. We propose to plot the resulting posterior over the feature space using a color code varying from deep blue to red according to the posterior values for the blue and the red class.

As shown on fig.2.11, when the tree gains in depth, it provide a posterior which better fits the underlying class distribution. As we consider only a single tree, the changes in posterior values are very sharp witnessing the underlying partition. For a depth of 15, we start noticing some signs of overfitting, as some noisy data points influence the posteriors. Hence, a good compromise has to be found for the tree depth as it has a big influence on the generalization.

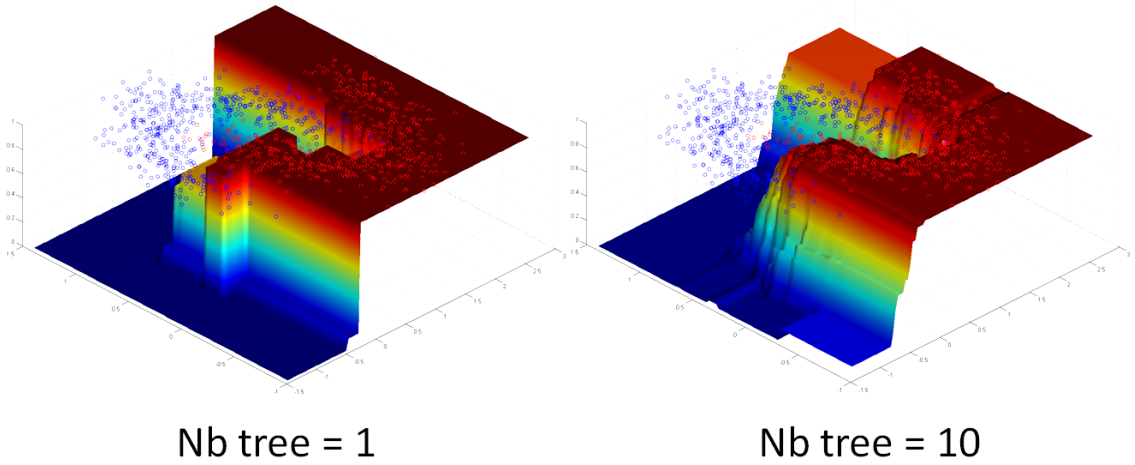

Figure 2.10: Classification posterior of a random forest: Increasing the number of trees provides a smoother posterior and permits to reach a greater generalization.

Now let us set the tree depth equal to 10 and vary the number of trees. As illustrated by fig.2.12 and 2.10, increasing the number of trees permits to get smoother posteriors, yielding smoother boundaries between the classes. Moreover, one can notice that the influence of noisy points decreases, as their contribution in the posterior estimation are averaged out. To conclude, increasing the number of trees permits to achieve greater generalization and smoother posteriors.

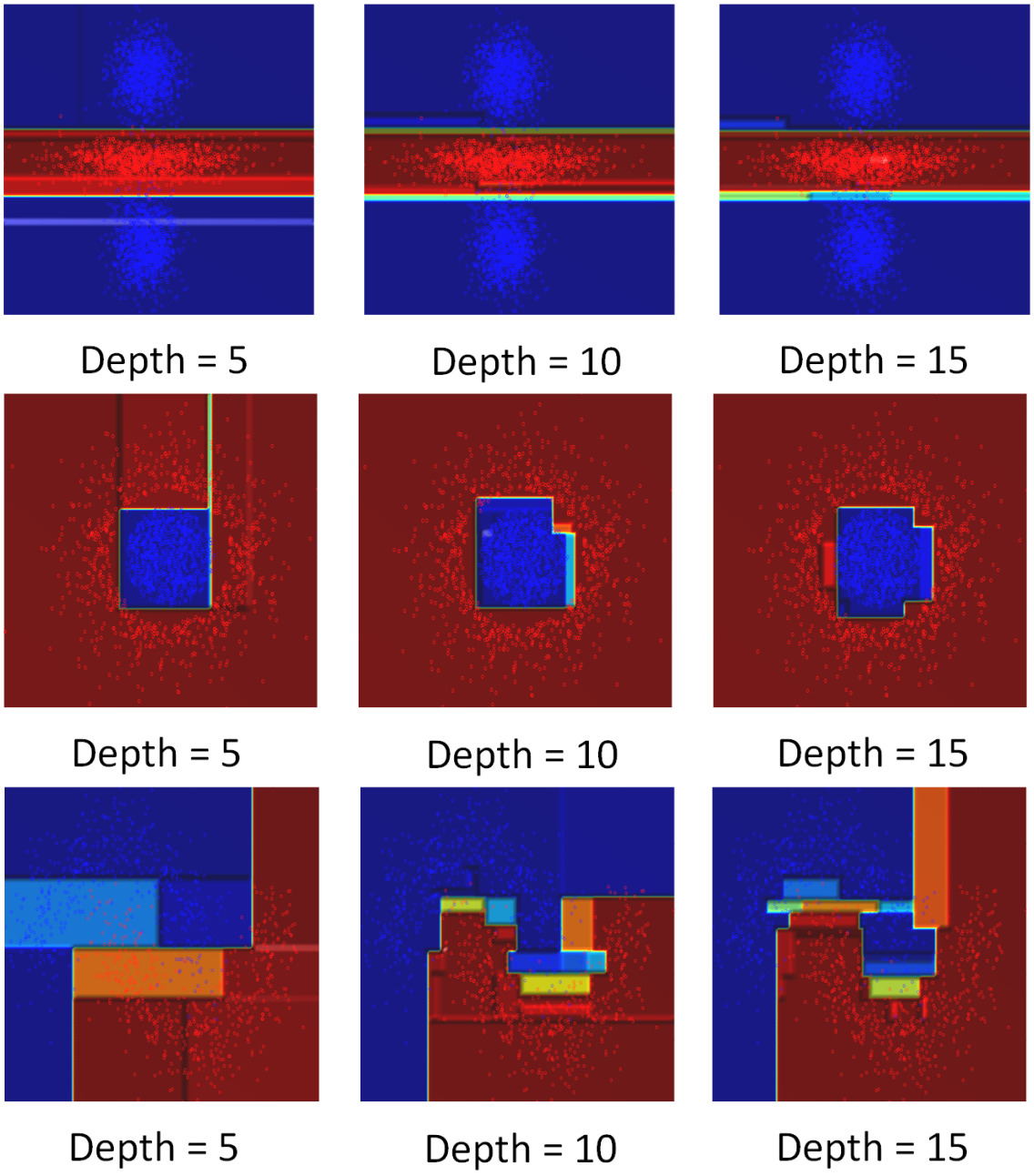

Figure 2.11: Classification posterior of a single random tree: we propose here to study the influence of the depth parameter.

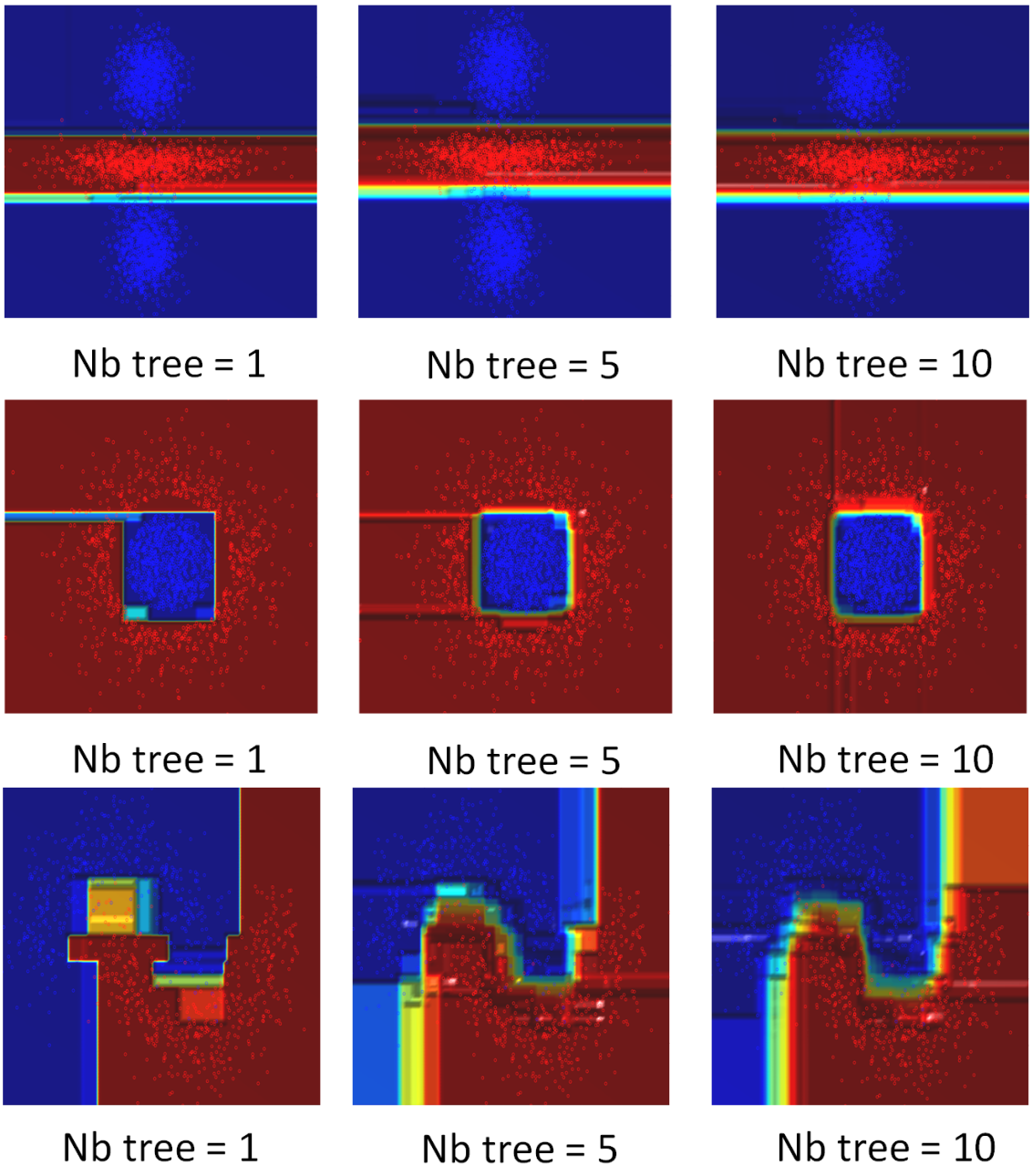

Figure 2.12: Classification posterior of a random forest: here the tree depth is set to 10, and we propose to study the influence of the number of trees.

2.4 Regression Forests

While random forests have been widely used for classification tasks, their ability to solve regression problems has been less studied despite all their advantages. Indeed, regression forests permit to efficiently model complex non-linear functions, and this, while being scalable to large training sets and high dimensional input and output spaces. In fact, they are very similar to classification forests, the only difference residing in the fact that the prediction output is continuous instead of being categorical, and can be multi-dimensional. In this section we start by defining the regression problem in a probabilistic fashion. Afterward, we detail how the posterior can be modeled in each leaf using a simple multivariate Gaussian distribution. Then we explain how to train regression forests and show how to define an objective function for regression. Afterward, we present different prediction approaches, and finally conclude the section with a few regression toy examples.

2.4.1 Problem Statement

We consider an input feature space $\mathcal{X} \subset \mathbb{R}^D$ and an output space $\mathcal{Y}$, which is a multi-dimensional continuous space $\mathcal{Y} \subset \mathbb{R}^{D'}$. To each input feature vector $\mathbf{X}$ is associated an output vector $\mathbf{Y} \in \mathcal{Y}$. Exactly as for classification forests, our goal is to model the posterior probability distribution $P(\mathbf{Y}|\mathbf{X})$. Given a training set $\{(\mathbf{X}^{(n)}, \mathbf{Y}^{(n)})\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, each tree of a forest $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^{T}$ permits to build a partition $\mathcal{P}_t$ over the input feature space $\mathcal{X}$. As for a classification task, two tree components needs to be instantiated for the regression task: (1) the leaf posterior and (2), the objective function.

2.4.2 Regression Posteriors

Let us consider the partition $\mathcal{P}_t = \{\mathcal{C}_t^{(z_t)}\}_{z_t=1}^{Z_t}$ built by the random tree $\mathbf{F}_t$. As illustrated by fig.2.13, posteriors can be modeled in each cell $\mathcal{C}_t^{(z_t)}$ as:

$$P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}_t^{(z_t)}, \mathcal{P}_t) = \mathcal{N}_t^{(z_t)}(\mathbf{Y} \mid \mu_t^{(z_t)}, \mathbf{\Sigma}_t^{(z_t)}) \tag{2.23}$$

$\mathcal{N}_t^{(z_t)}$ is a multivariate Gaussian with mean $\mu_t^{(z_t)}$ and covariance matrix $\mathbf{\Sigma}_t^{(z_t)}$ estimated in the output space $\mathcal{Y}$ from the subset of the training points that fall into the cell $\mathcal{C}_t^{(z_t)}$ of partition $\mathcal{P}_t$. While many other choices are possible to model the posterior in each leaf such as probabilistic linear or Gaussian mixtures [20], we focus in this thesis on single multivariate Gaussian models for their simplicity. A regression tree is finally equivalent to a probabilistic piece-wise constant regressor, and by using trees that are deep enough, one can approximate any arbitrary functions, even with such a simple model, as soon as they are injective. During the training of the tree, the goal is to reduce the prediction uncertainty linked to this multivariate Gaussian model. In the following section, we detail how to define the objective function for node optimization.

2.4.3 Regression Objective Function

As for classification tasks, at each node N_l of the tree $\mathbf{F}_t$, a splitting function f_l permits to split the subset $\mathcal{S}_l$ of the training set arriving in this node. The goal of node optimization

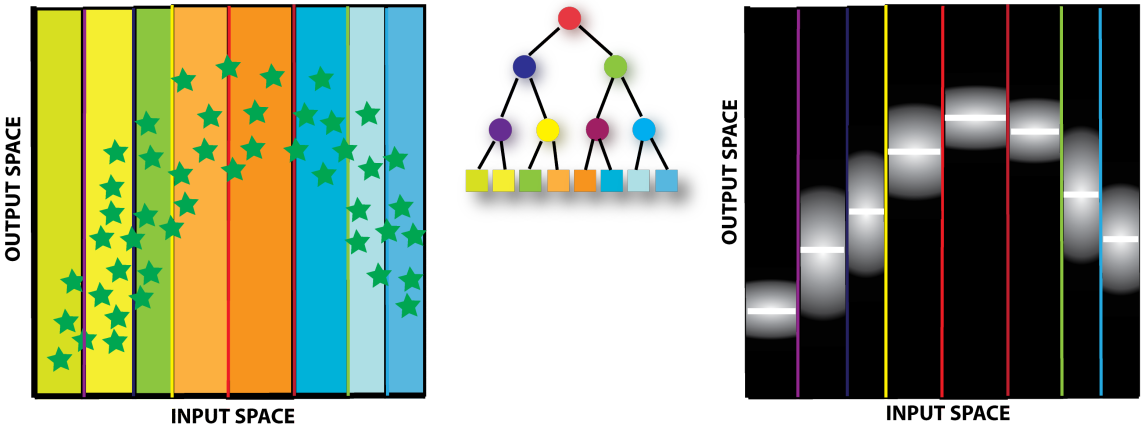

Figure 2.13: Regression forest: each tree $\mathbf{F}_t$ builds a partition $\mathcal{P}_t$ over the feature space and regression posteriors can be easily approximated in each cell of $\mathcal{P}_t$

is to find the best splitting function aiming at reducing the prediction uncertainty. In the present thesis, we will use the **Information Gain**, based on the continuous version of Shannon's entropy:

$$H(\mathcal{S}_l) = \int_{\mathbf{Y}\in\mathcal{Y}} P(\mathbf{Y}|\mathcal{S}_l)\log\left(P(\mathbf{Y}|\mathcal{S}_l)\right)d\mathbf{Y} \tag{2.24}$$

As posteriors are modeled using a multivariate Gaussian, H has following closed form:

$$H(\mathcal{S}_l) = \frac{1}{2}\log\left((2\pi e)^{D'}|\Sigma^{(\mathcal{S}_l)}|\right) \tag{2.25}$$

where $\Sigma^{(\mathcal{S}_l)}$ is the covariance matrix estimated in the output space $\mathcal{Y}$ from the subset of training points $\mathcal{S}_l$. After splitting $\mathcal{S}_l$ into two subsets $\mathcal{S}_l^{\text{left}}$ and $\mathcal{S}_l^{\text{right}}$ that are respectively sent to the left and right child nodes, the reduction of uncertainty can be measured using the information gain Δ:

$$\Delta = H(\mathcal{S}_l) - w_{\text{left}}H(\mathcal{S}_l^{\text{left}}) - w_{\text{right}}H(\mathcal{S}_l^{\text{right}}) \tag{2.26}$$

where $w_{\text{left}} = |\mathcal{S}_l|/|\mathcal{S}_l^{\text{left}}|$ and $w_{\text{right}} = |\mathcal{S}_l|/|\mathcal{S}_l^{\text{right}}|$. Again, during node optimization, several splitting function candidates are generated and the best is then chosen by maximizing Δ:

$$f_l^* = \mathbf{argmax}_{f_l\in\Gamma_l}\,\Delta(\mathcal{S}_l, \mathcal{S}_l^{\text{left}}, \mathcal{S}_l^{\text{right}}) \tag{2.27}$$

Intuitively, optimizing this objective function yields leaf clusters of data points that are consistent in the input feature space $\mathcal{X}$ and in the output space $\mathcal{Y}$.

2.4.4 Forest Prediction

Once the training phase accomplished, predictions can be performed for new incoming observations by sending them through all trees of the forest and combining tree posteriors.

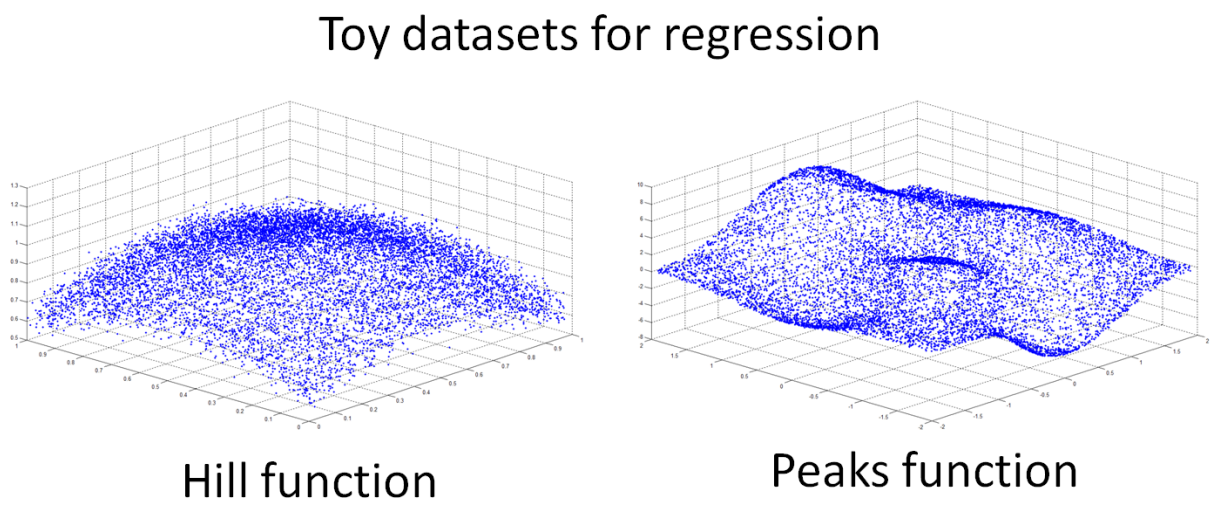

Figure 2.14: Regression toy examples: we propose to study the forests behaviour on these 2 functions

As for classification forests, the posterior distributions from all individual trees can be averaged:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} P(\mathbf{Y}|\mathbf{X}, \mathcal{P}_t) \tag{2.28}$$

where $\mathcal{P}_t$ is the partition induced by tree $\mathbf{F}_t$. Predictions can be then computed using either the maximum a posteriori:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}\in\mathcal{Y}}\, P(\mathbf{Y}|\mathbf{X}) \tag{2.29}$$

or alternatively, the conditional mathematical expectation $\mathrm{E}\,[\mathbf{Y}|\mathbf{X}]$:

$$\bar{\mathbf{Y}} = \int_{\mathbf{Y}} \mathbf{Y} P(\mathbf{Y}|\mathbf{X}) d\mathbf{Y} \tag{2.30}$$

which can be simplified in the case of multivariate Gaussian models to:

$$\bar{\mathbf{Y}} = \frac{1}{T}\sum_{t=1}^{T} \mu_t^{(z_t)} \tag{2.31}$$

where $\mu_t^{(z_t)}$ are the means in the leaves in which observation $\mathbf{X}$ falls in each tree. Furthermore, one can derive the confidence of a leaf prediction from its associated covariance matrix. Indeed, individual trees can show very different confidences depending on the part of the feature space the new observation $\mathbf{X}$ is falling. Hence, tree posteriors can be first ranked according to their confidence, and averaging can be performed using only the best tree predictions.

2.4.5 A Few Toy Examples

In this part, we propose to illustrate how regression forests permit to approximate arbitrary functions, and to show the influence of the main forests parameters, *e.g.* the tree

depth and the number of trees. Therefore, we will use 2 toy examples using two functions parametrized as $z = f(x, y)$ we call "hill" and "peaks" functions (see fig.2.14). These two datasets consist of 10000 points (x, y, z) generated using non-linear functions and additive noise. Here, the x and y dimensions will represent our input feature space and the z dimension the output space. While the first function possess only one maximum, it is very noisy. In contrast, the second function shows more variations but is less noisy.

Let us start with a single tree, where at each node, a dimension is set at random and a threshold is randomly chosen within the interval defined by the features of the data points. This permits to generate functions corresponding to simple axis-aligned splits. The number of function candidates is set to 10 per node and we vary the depth of the tree between 5 and 15. In each leaf, the regression posterior is modelled by a one-dimensional Gaussian distribution, where the mean and the variance are estimated from the training data. We propose to plot the resulting mathematical expectation over the feature space, and to overlay some points of the training set to show how well the predicted function fits the data points. Note that the output or the regression forest in this configuration is an ensemble of piece-wise constant approximations of the input data.

As shown on fig.2.15, when the tree gains in depth, it provide a regression output which better fits the underlying function. Here we consider only a single tree, and the changes in values are very sharp as the tree output is a piece-wise function. In the case of the "hill" function, we can clearly notice problems of overfitting, as noisy data points have a big impact on the regression output. Here again, a good compromise has to be found for the tree depth as it has a big influence on the generalization.

Now let us set the tree depth equal to 10 and vary the number of trees. As illustrated by fig.2.16, increasing the number of trees permits to get smoother regression output. This clearly demonstrates the great potential of regression forest, *i.e.* how an ensemble of piece-wise function approximations can yield a nice and smooth approximation of a non-linear function. Moreover, one can notice that the influence of noisy points decreases, as their contribution in the posterior estimation are averaged out. To conclude, increasing the number of trees also permits to achieve greater generalization.

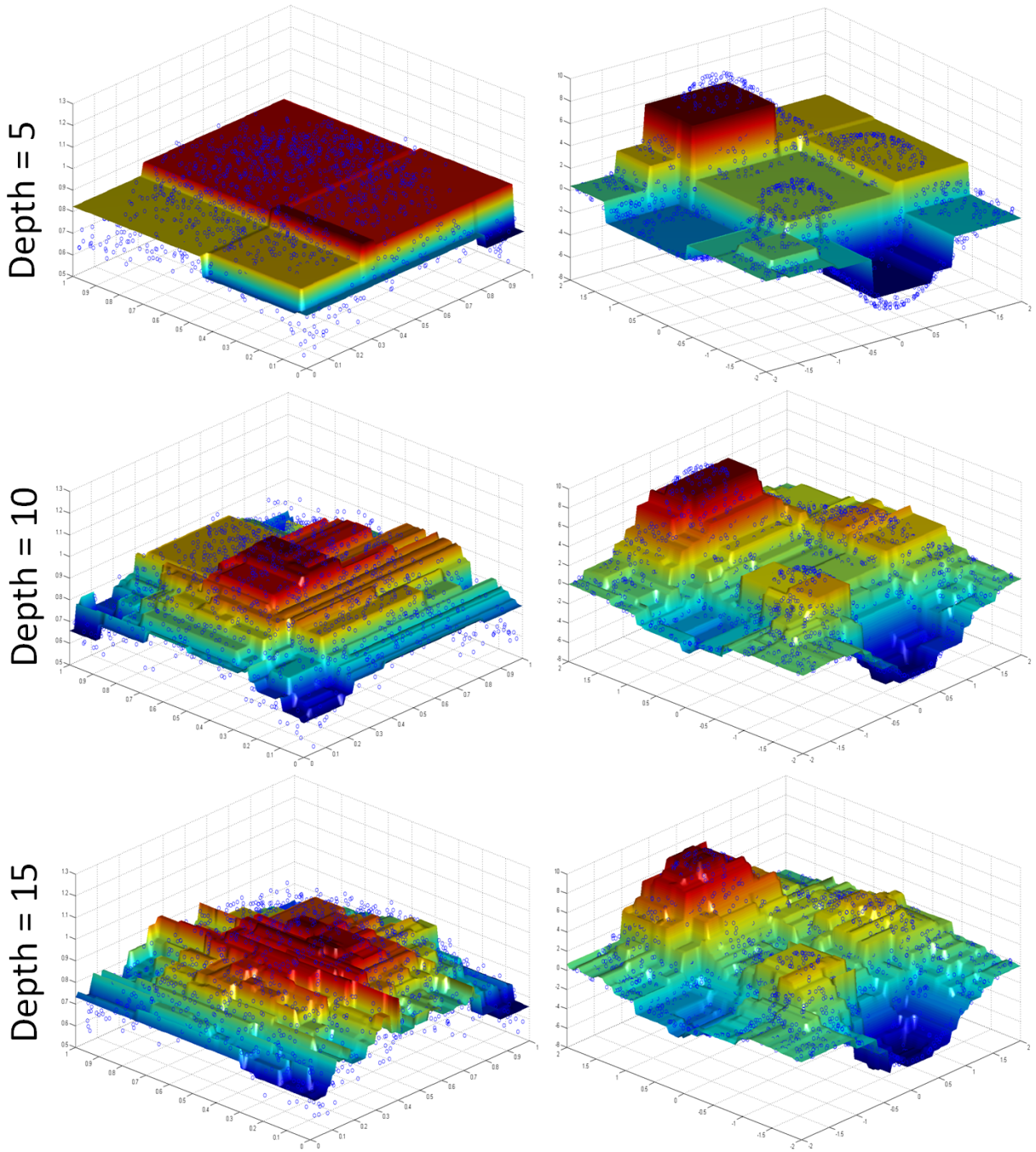

Figure 2.15: Regression output of a single random tree on two toy examples: we propose here to study the influence of the depth parameter.

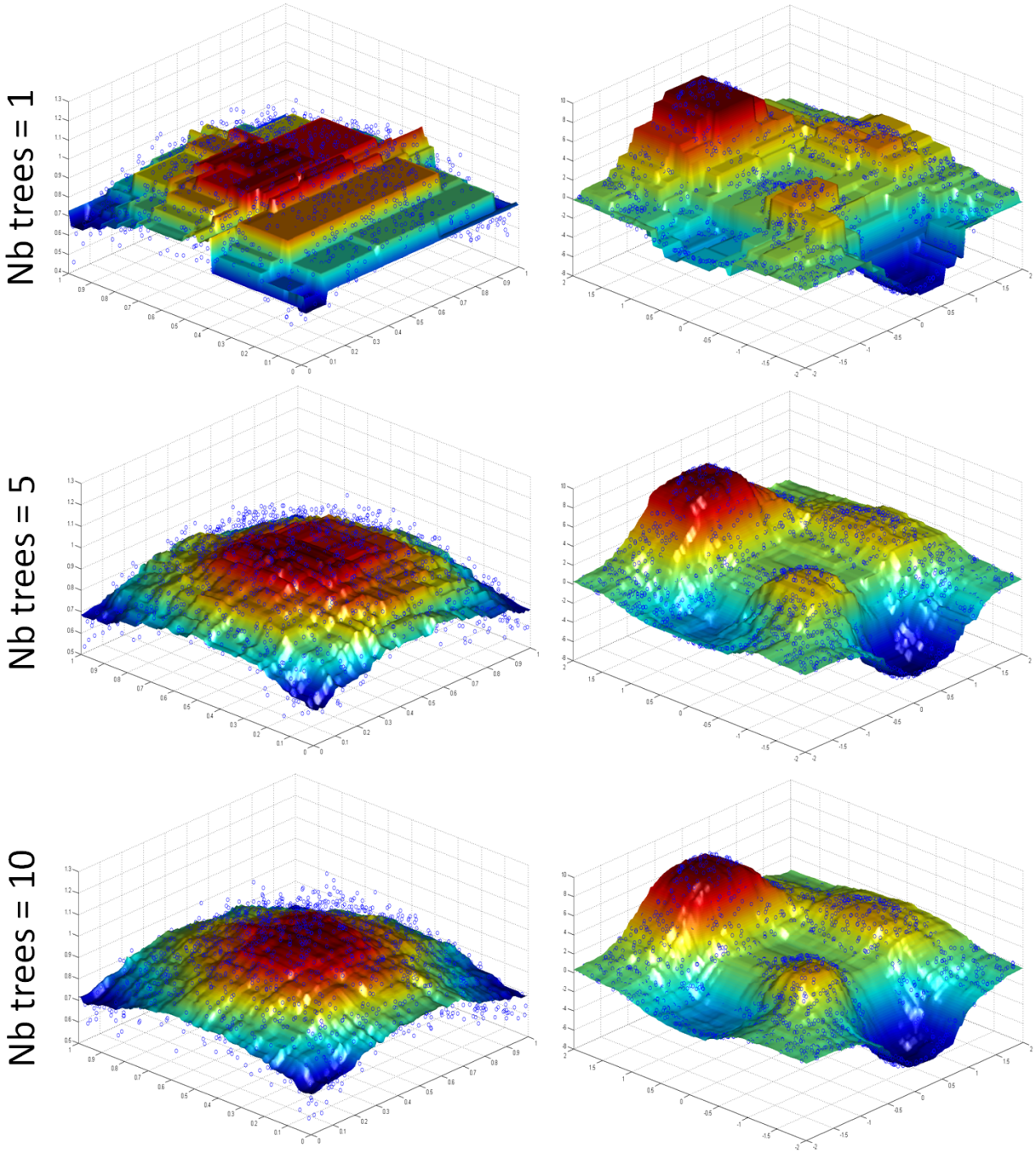

Figure 2.16: Regression output of a random forest on two toy examples: here the tree depth is set to 10 and we propose to study the influence of the number of trees.

2.5 Clustering Forests

Classification and regression are two classical supervised learning tasks as the goal is to model the relationship between an input and an output feature space. As only training points from an input feature space $\mathcal{X}$ are available, clustering and density estimation are unsupervised problem. While in clustering, the goal is to discover "clusters" or in other words groups of points having similar characteristics in $\mathcal{X}$, in density estimation, one aims at modelling the probability distribution $P(\mathbf{X})$ where $\mathbf{X} \in \mathcal{X}$. In the present thesis, we will show how to define, train and use random forests for clustering tasks such as visual dictionary learning for image categorization or retrieval. For the derivation of random forests for density estimation, we invite the reader to refer to [20].

2.5.1 Problem Statement

As clustering is an unsupervised task, we consider an input feature space $\mathcal{X} \subset \mathbb{R}^D$ only. Our goal is to discover a set of K clusters $\mathcal{K} = \{\mathbf{K}_k\}_{k=1}^K$ consisting of observations that are consistent in $\mathcal{X}$. Given a set $\{\mathbf{X}^{(n)}\}_{n=1}^N \in \mathcal{X}$, each tree of a forest $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^T$ permits to build a partition $\mathcal{P}_t$ over the input feature space $\mathcal{X}$. The main idea of clustering forests is very simple: each partition $\mathcal{P}_t$ will be constructed so that each of its cells maximizes the consistency of the points it contains. Each cell of $\mathcal{P}_t$ corresponds then to a cluster.

2.5.2 Cluster Model

Let us consider the partition $\mathcal{P}_t = \{\mathcal{C}_t^{(z_t)}\}_{z_t=1}^{Z_t}$ built by the random tree $\mathbf{F}_t$. As illustrated by fig.2.17, a random tree is able to efficiently find clusters in high-dimensional spaces by simply associating each cell $\mathcal{C}_t^{(z_t)}$ to a cluster according to the partition it induces. Thus, each tree can map a point $\mathbf{X} \in \mathcal{X}$ to a cluster simply by looking at the cell it falls in:

$$\mathbf{F}_t(\mathbf{X}) = \mathcal{C}_t^{(z_t)} \tag{2.32}$$

If the tree is deep, then the partition $\mathcal{P}_t$ counts many cells and thereby many clusters. If the tree is not deep, then it will yield only a few clusters. Of course, the partitioning results from a tree can be further processed by for instance merging neighboring and consistent cells. However, performing additional steps for each tree of the ensemble may increase consequently the clustering complexity.

Now, if we consider the entire forest $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^T$, each individual tree induces its own partition, so a point $\mathbf{X} \in \mathcal{X}$ is finally associated to a vector of cells:

$$\mathcal{F}(\mathbf{X}) = \left\{\mathcal{C}_1^{(z_1)}, \cdots, \mathcal{C}_t^{(z_t)}, \cdots, \mathcal{C}_T^{(z_T)}\right\} \tag{2.33}$$

Instead of having each point belonging to one cluster, each point is associated to a set of clusters coming from different partitioning results of the same feature space. Thus, now following question arises: how can we merge these multiple clustering results into one global clustering? The problem becomes now to find a mapping Λ which associates each set of cells $\mathbf{C} = \left\{\mathcal{C}_1^{(z_1)}, \cdots, \mathcal{C}_t^{(z_t)}, \cdots, \mathcal{C}_T^{(z_T)}\right\}$ to a global cluster:

$$\Lambda(\mathbf{C}) = \mathbf{K}_k \tag{2.34}$$

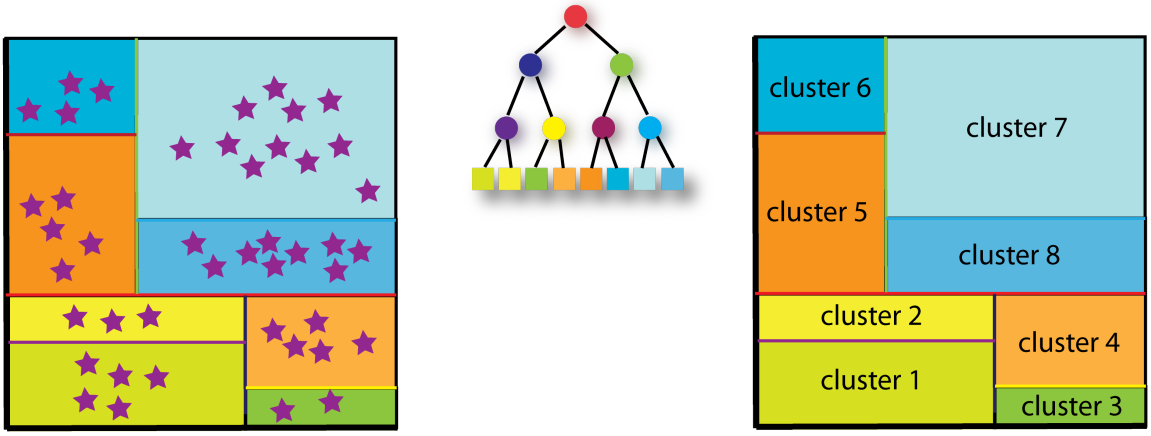

Figure 2.17: Clustering forest: each tree $\mathbf{F}_t$ builds a partition $\mathcal{P}_t$ over the feature space and each cell is associated to a cluster.

Merging multiple clustering is a very general problem, and many approaches have been proposed as reported for instance in [92]. To respect the philosophy of random forests, we focus on two very simple approaches for creating global clustering: (1) perform **intersection** between the different partitions to create a global partition and thereby global clusters, (2) keep the vectors of cells as an **implicit** representation of the global clusters.

Inspired from [79], the first approach proposes to merge all partitions $\{\mathcal{P}_t\}_{t=1}^{T}$ into a global partition $\mathcal{P}_{\text{global}} = \left\{\mathcal{C}_{\text{global}}^{z_g}\right\}_{z_g=1}^{Z_g}$ by computing their intersection. All regions of $\mathcal{X}$ are subdivided so that each cell $\mathcal{C}_{\text{global}}^{z_g}$ corresponds to a unique vector $\mathbf{C} = \left\{\mathcal{C}_1^{(z_1)}, \cdots, \mathcal{C}_t^{(z_t)}, \cdots, \mathcal{C}_T^{(z_T)}\right\}$. Of course, not all vector combinations really represent a global cell. During the training, global cells are identified as those being effectively populated by observations. Finally, these global cells are associated to clusters, and Λ is defined as the mapping associating a vector $\mathbf{C}$ to a global cell $\mathcal{C}_{\text{global}}^{z_g}$ and thereby to a cluster.

Already applied to learn visual dictionary for image categorization [64, 89], the second approach is much simpler. No further operation is required as observations are represented by their cell vector $\mathbf{C}$. The global clustering is kept implicit, and can be seen as a simple concatenation of the multiple clustering results coming from the different trees.

2.5.3 Clustering Objective Function

Let us now briefly detail how to train a clustering tree and how to define an appropriate objective function. Considering a node N_l of the tree $\mathbf{F}_t$, a splitting function f_l needs to be chosen to split the subset $\mathcal{S}_l$ of the training set arriving in this node. To find the best splitting function for this node, we need to define an objective function. While in classification or regression, the objective function was defined on the output space, in the present case, it needs to be constructed in the input space. Indeed, in supervised learning,

decisions are made in the input space, but are chosen so that they also reduce uncertainty in the output space. In contrast to supervised learning, clustering explicitely aims at enforcing a consistence in $\mathcal{X}$, and this, using decisions made in $\mathcal{X}$. An objective function can be then defined to reduce the uncertainty in $\mathcal{X}$, using the **Information Gain** based on the continuous version of Shannon's entropy:

$$H(\mathcal{S}_l) = \int_{\mathbf{X}\in\mathcal{X}} P(\mathbf{X}) \log\left(P(\mathbf{X})\right) d\mathbf{X} \tag{2.35}$$

As the distribution in each node is modeled using a multivariate Gaussian, H has following closed form:

$$H(\mathcal{S}_l) = \frac{1}{2}\log\left((2\pi e)^D |\Sigma^{(\mathcal{S}_l)}|\right) \tag{2.36}$$

where $\Sigma^{(\mathcal{S}_l)}$ is the covariance matrix estimated in the input space $\mathcal{X}$ from the subset of training points $\mathcal{S}_l$. After splitting $\mathcal{S}_l$ into two subsets $\mathcal{S}_l^{\text{left}}$ and $\mathcal{S}_l^{\text{right}}$ that are respectively sent to the left and right child nodes, the reduction of uncertainty can be measured using the information gain Δ:

$$\Delta = H(\mathcal{S}_l) - w_{\text{left}} H(\mathcal{S}_l^{\text{left}}) - w_{\text{right}} H(\mathcal{S}_l^{\text{right}}) \tag{2.37}$$

where $w_{\text{left}} = |\mathcal{S}_l|/|\mathcal{S}_l^{\text{left}}|$ and $w_{\text{right}} = |\mathcal{S}_l|/|\mathcal{S}_l^{\text{right}}|$. During node optimization, several splitting function candidates are generated and the best is then chosen by maximizing Δ:

$$f_l^* = \mathbf{argmax}_{f_l\in\Gamma_l} \Delta(\mathcal{S}_l, \mathcal{S}_l^{\text{left}}, \mathcal{S}_l^{\text{right}}) \tag{2.38}$$

Again, note that optimizing this objective function yields leaf clusters of data points that are consistent in the input feature space $\mathcal{X}$.

2.5.4 Forest Prediction

Once the training phase accomplished, a new incoming observation $\mathbf{X}$ can be pushed through all trees of the forest $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^T$, to compute its corresponding cell vector $\mathbf{C} = \left\{\mathcal{C}_1^{(z_1)}, \cdots, \mathcal{C}_t^{(z_t)}, \cdots, \mathcal{C}_T^{(z_T)}\right\}$. Then, by using one of the two approaches discussed previously, this cell vector $\mathbf{C}$ can be associated to a global cluster by using explicit intersection, or $\mathbf{C}$ can be used as an implicit representation of the global cluster.

2.6 Conclusion

In this chapter, we presented random forests, a fascinating multi-task ensemble learner, which consists of an ensemble of decision trees. In this thesis, we propose a partition formalism to fully understand their philosophy: **divide and conquer**. Indeed, random forests basically aim at constructing piece-wise posterior models by, (1) creating a partition over the full feature space using simple decisions, and (2) model the posterior distribution in each cell of this space. We demonstrated along the different sections that, by defining the right **objective function** and designing an appropriate **posterior model** within the leaf, one can adapt random forests to tackle any kind of learning problem. Indeed, while they have been mainly used for classification, random forests can be formulated to solve many other learning tasks such as regression or clustering.

CHAPTER

THREE

RELATED RANDOM ENSEMBLE PARTITIONING APPROACH: RANDOM FERNS

"So much of life, it seems to me, is determined by pure randomness."
Sidney Poitier

In this chapter, we present a forest-related approach, namely the random ferns. Proposed originally for tracking application [71], random ferns were motivated by the need of fast learning, fast prediction and less memory consumption. Therefore, authors abandonned node optimization to increase the learning speed and designed a constrained tree having only one decision function per level to get a more compact model. Random ferns are a random partitioning approach which are often presented as an ensemble of constrained trees. Indeed, a fern is basically a tree which systematically applies the same decision function for each node of the current level. In the following, we will first introduce random ferns in their original application and explain how they can be interpreted as intersection of decision stumps. We will discuss their similarities and differences with random trees, and show how to instantiate them for classification, regression and clustering tasks.

When they designed the random ferns approach, the main motivation of Özuysal *et al.* was to be able to learn and recognize patch classes, and this, faster than with random trees. As shown on fig.3.1, the basic idea was to perform a sequence of simple tests relying on the comparison of pixel intensities within a patch. The position of the pixels to compare are chosen at random, and the results of the tests are stored as binary numbers. These binary numbers permit then to encode the bin indexes of multinomial distributions that model ferns outputs for the different classes. During the training phase, these multinomial distributions, or in other words class histograms are learned by performing the sequence of tests on training examples. Depending on their results to the tests, the training patches fall in different bins and class histograms are incremented accordingly. Finally, prediction can be made for new incoming patches by performing this sequence of tests, and reading out the class posteriors contained in the different class histograms at the resulting index. While random ferns benefit of a more compact and simple structure than random trees, authors demonstrates in [71] that they show similar performances for patch classification. In the following, we will formalize the random fern model, and show that it can be also interpreted as a partitioning approach. However, at the difference of random trees, random ferns are not real hierarchical models.

3.1 Ferns Model

Considering an input feature space $\mathcal{X} \subset \mathbb{R}^D$ and an output space $\mathcal{Y} \subset \mathbb{R}^{D'}$, we aim at learning the posterior distribution $P(\mathbf{Y}|\mathbf{X})$ where $\mathbf{X} \in \mathcal{X}$ and $\mathbf{Y} \in \mathcal{Y}$. To perform predictions in $\mathcal{Y}$ given an observation in $\mathcal{X}$, we can use maximum a posteriori:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y} \in \mathcal{Y}} P(\mathbf{Y}|\mathbf{X}) \tag{3.1}$$

As for random trees, given a training set $\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, we learn the posterior $P(\mathbf{Y}|\mathbf{X})$ by (1) building a **partition** over the input feature space, and (2) **estimating** $P(\mathbf{Y}|\mathbf{X})$ in each "cell" of this space.

Intuitively, random ferns can be seen as constrained random trees, which have only one decision function or node per level as illustrated by fig.3.2. They build a partition $\mathcal{P}$ over the feature space $\mathcal{X}$, and this, by using the same sequences of decision functions for all training data. This means that data are not explicitely split and sent towards left or right children as in randomized trees, and decision functions are defined over the whole feature space (see fig.3.3). This is the major difference between trees and ferns and implies that a random fern is an ensemble of decision functions $\mathbf{F} = \{\mathbf{N}_l\}_{l=1}^{L}$, and not a real hierarchical model. A more appropriate interpretation would be an **intersection of decision stumps**, where each decision stump is represented by a node N_l. Each node N_l is equipped with a splitting function f_l defined as:

$$\begin{cases} f_l : \mathcal{X} \to \mathbb{B} \\ f_l(\mathbf{X}) = (\mathbf{X} \cdot \mathbf{v}_l \geq \tau_l) \end{cases} \tag{3.2}$$

where $\dim(\mathbf{v}_l) = \dim(\mathcal{X})$ and $\tau_l \in \mathbb{R}$. The role of f_l here is to split the *full* feature space into 2 halves we denote $\mathcal{H}_l^{(0)}$ and $\mathcal{H}_l^{(1)}$ when $f_l(\mathbf{X}) = 0$ or 1 respectively.

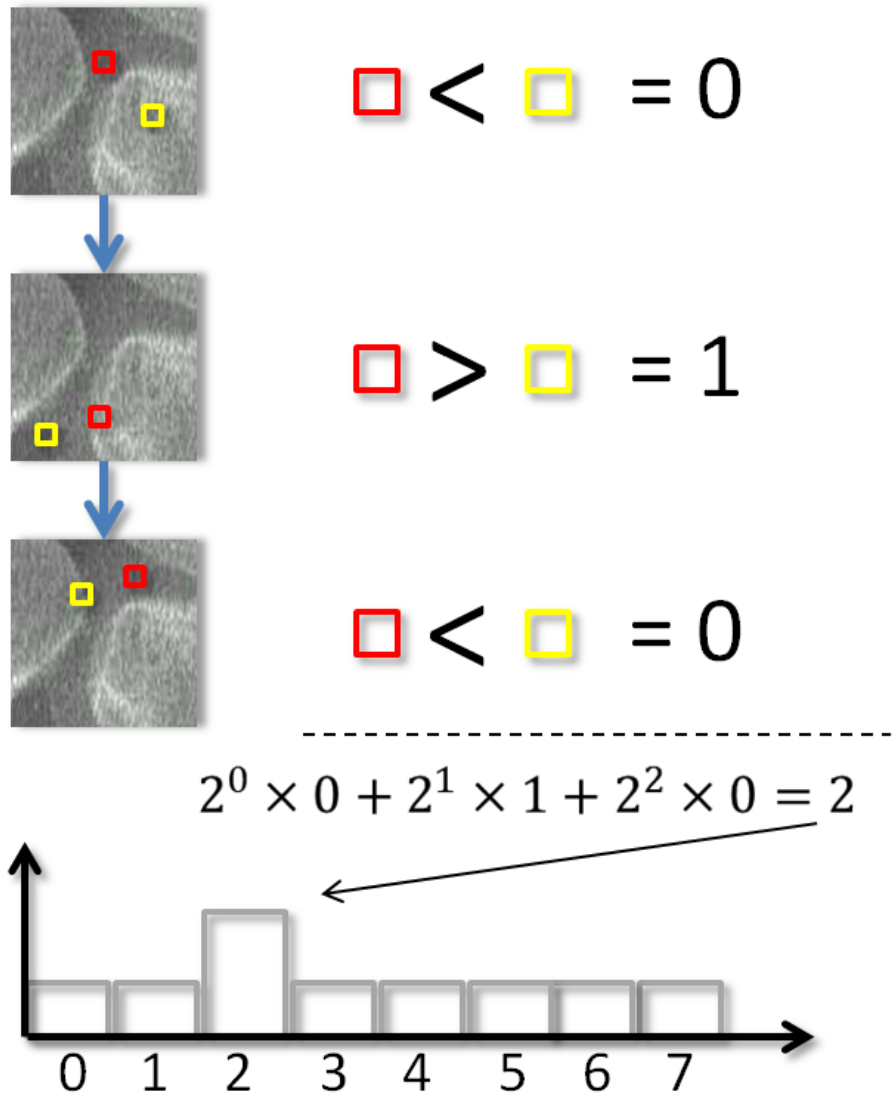

Figure 3.1: Random ferns: Introduced for patch classification, random ferns rely on a sequence of simple tests comparing the intensity of pixels at random positions. Results of these tests are stored as binary numbers that encode bin indexes, or in other words, cells of the feature space.

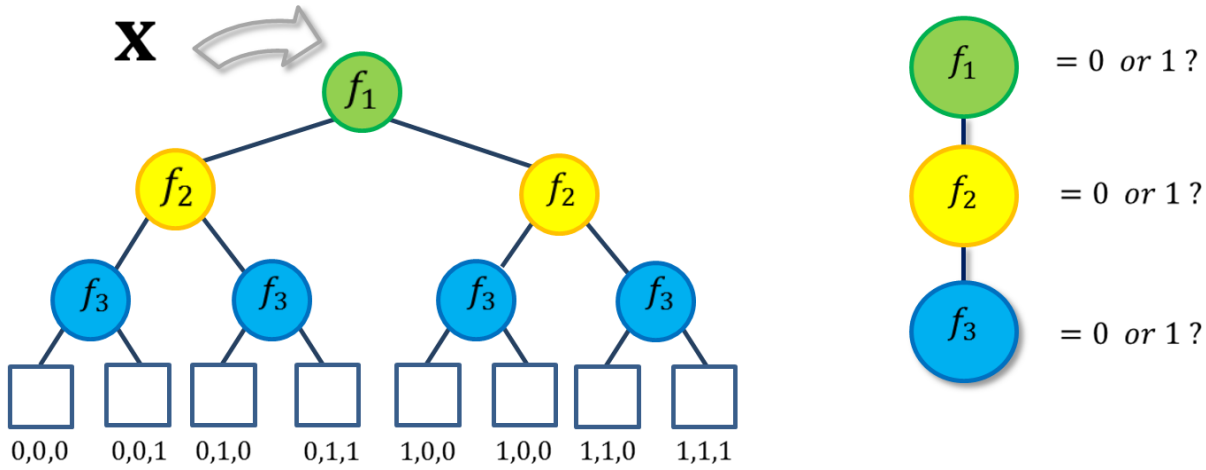

Figure 3.2: Random ferns are often interpreted as constrained random trees: they have only one decision function or node per level.

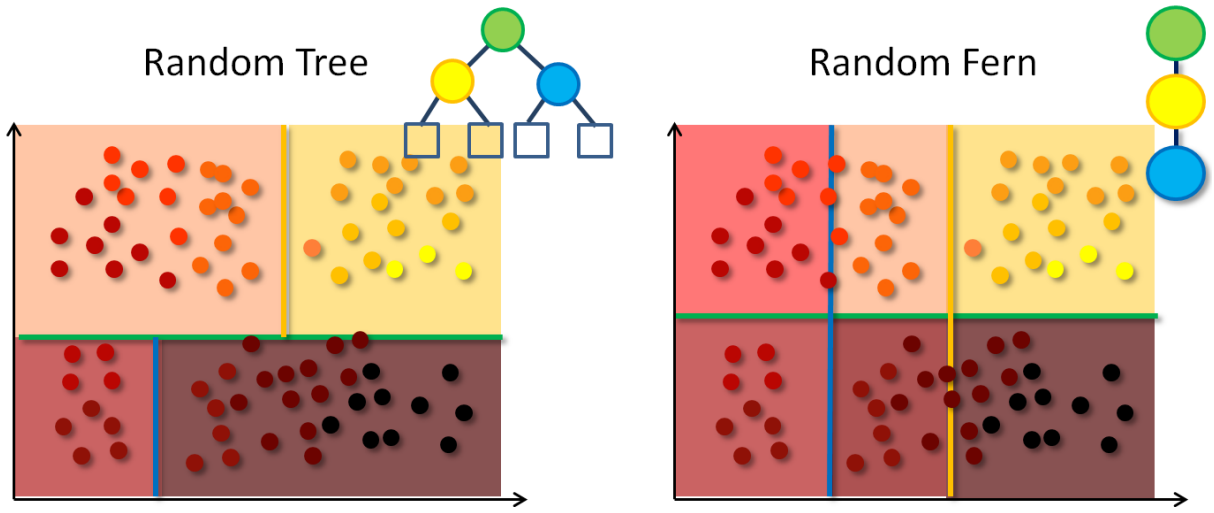

Figure 3.3: Partitions induced by random trees and ferns: As they have only one node per level, ferns have decision functions defined over the whole feature space.

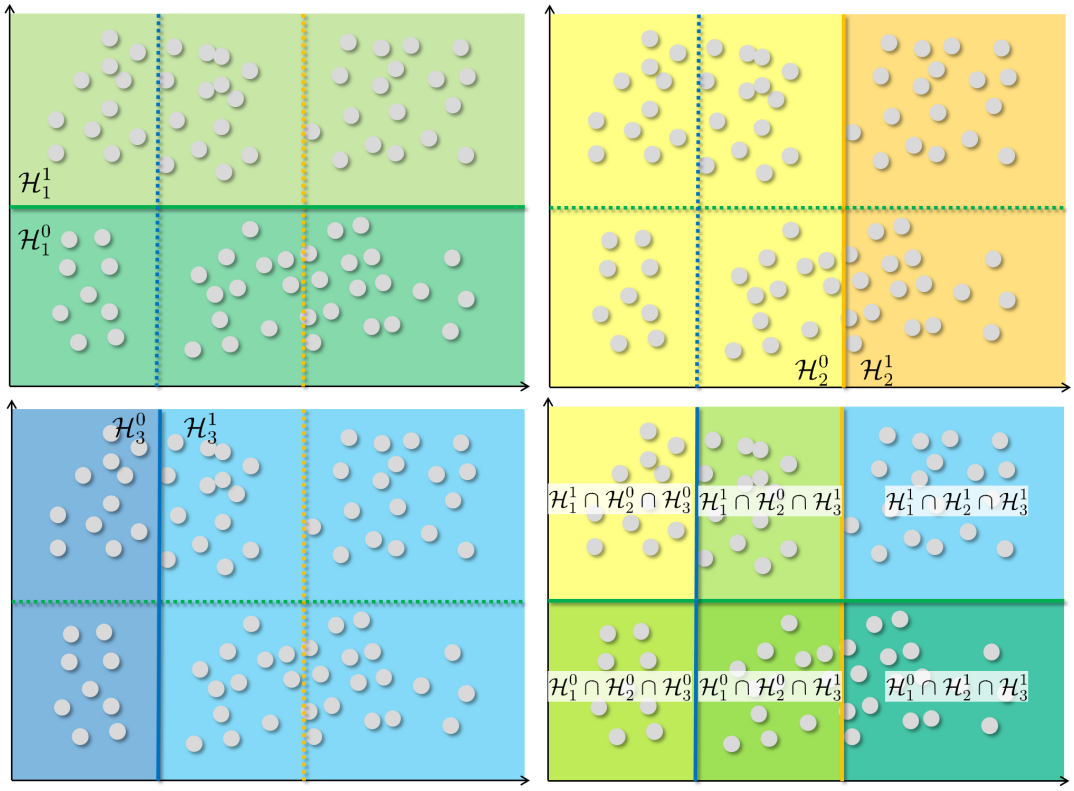

Figure 3.4: Random ferns as intersection of decicsion stumps: Each decision function splits the whole feature space in two half spaces. Cells of the partition induced by a random fern result from the intersection of these half spaces.

Let us now consider a point $\mathbf{X} \in \mathcal{X}$ which is sent through the fern $\mathbf{F}$. Gathering the outputs of the decision functions at each node, $\mathbf{X}$ is associated to a set of half-spaces $\left\{\mathcal{H}_1^{(b_1)}, \cdots, \mathcal{H}_l^{(b_l)}, \cdots, \mathcal{H}_L^{(b_L)}\right\}$, where the superscript b_l denotes the binary output of function f_l. As $\mathbf{X}$ belongs to all these half-spaces, one can define the cell $\mathcal{C}^{(z)}$ containing $\mathbf{X}$ as:

$$\mathcal{C}^{(z)} = \bigcap_{l=1}^{L} \mathcal{H}_l^{(b_l)} \tag{3.3}$$

As illustrated in fig.3.4, we can thereby create a partition $\mathcal{P} = \{\mathcal{C}^{(z)}\}_{z=1}^{Z}$ over $\mathcal{X}$. In the ferns implementation, the computation of this intersection is implicitly solved by using a binary encoding. Indeed, the outputs of all decision functions are combined to determine the index z of the cell $\mathcal{C}^{(z)}$ in which $\mathbf{X}$ falls as follows:

$$z = 2^0 \cdot b_1 + \cdots + 2^{l-1} \cdot b_l + \cdots + 2^{L-1} \cdot b_L \tag{3.4}$$

Clearly, the order in which the decision functions are evaluated does not change the underlying partition. Indeed, nodes can be interverted, yielding only a change in the binary encoding, but the resulting partition would stay the same. Computations on nodes could be even parallelized. However, random ferns suffer from two important limitations: (1) there is no guarantee that all cells of the partition $\mathcal{P}$ will be populated during the training phase, and (2) not all binary combinations induce a possible cell. The first limitation implies that, in contrast to random trees, a random fern may create empty cells during its training phase. This can happen for instance if the fern is very deep, or if the training set is not big enough or not representative of the feature space. Then, as no incoming training data reaches these empty cells, no posterior models can be learned. Consequently, if new incoming observations fall in this non-populated cell, no reliable prediction can be performed, except if a prior distribution is available. The second limitation is less problematic as it only means that the cell encoding could be more compact. Let us now detail briefly the training procedure of random ferns.

3.1.1 Random Ferns Training

As shown in alg.3, the training procedure of random ferns is very simple. In contrast to random trees, there is no node optimization, and the training data is not explicitely split. Hence, the observations just need to be pushed through all nodes and then all binary outputs are stored. Afterward, the corresponding cell indexes are computed from the binary outputs of each point. Finally, posteriors can be computed in each cell $\mathcal{C}^{(z)}$ from its associated training points:

$$P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}^{(z)}, \mathcal{P}) \tag{3.5}$$

3.1.2 Random Ferns Prediction

Once a random fern has been trained, a prediction for a new unseen observation $\mathbf{X}$ can be very efficiently performed as detailed in the pseudo-code in alg.4. $\mathbf{X}$ is basically pushed through all nodes, and binary outputs are gathered to compute the index of the cell $\mathcal{C}^{(z)}$

Algorithm 3: Random Ferns Training: Pseudocode example

Training set: $\mathcal{S} = \left\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\right\},\ n \in \{1, \cdots, \mathrm{N}\}$
Random fern object: **F**
Parameters: NbNodes
initialize matrix containing binary vectors
$\mathbf{B}$ = new Matrix (N,NbNodes),
loop over the nodes
for (**int** $i = 1,\ i \leq$ NbNodes, $i++$) **do**
    $f \leftarrow$ **generateRandomSplittingFunction**;
    F.splitFunc$\{i\} \leftarrow f$
    compute binary outputs and store them
    $\mathbf{B}(:, i) \leftarrow$ **computeSplittingFunctionOutputs**$(\mathcal{S}, f)$
end for
*compute cell indexes from binary vectors and store them in vector **Z** of length N*
$\mathcal{Z} \leftarrow$ **computeCellIndexes**$(\mathbf{B})$
loop over the cells and estimate posterior
$Z = 2^{\mathrm{NbNodes}}$
for (**int** $z = 1,\ z \leq$ Z, $z++$) **do**
    retrieve training points falling in current cell
    $\mathcal{S}_z \leftarrow$ **retrieveDataInCell**$(\mathcal{S}, \mathcal{Z}, z)$
    Learn posterior from these training points
    F.Posterior$\{z\} \leftarrow$ **estimatePosteriorDistribution**$(\mathcal{S}_z)$;
end for
Output: trained random fern **F**

Algorithm 4: Random Ferns Prediction: Pseudocode example

Observation: **X**
Random fern object: **F**
initialize binary vector
$\mathbf{B}$ = new Vector (NbNodes),
loop over the nodes
for (**int** $i = 1,\ i \leq$ NbNodes, $i++$) **do**
    compute binary outputs and store them
    $\mathbf{B}(i) \leftarrow$ **computeSplittingFunctionOutputs**$(\mathbf{X}, \mathbf{F}.\mathrm{splitFunc}\{i\})$
end for
compute cell indexes from binary vectors
$z \leftarrow$ **computeCellIndexes**$(\mathbf{B})$
retrieve cell posterior
Posterior = **F**.Posterior$\{z\}$;
Output: **Posterior**

it falls in. Hence, at test time, a fern $\mathbf{F}$ can be seen as a function taking as input an observation and returning a cell:

$$\begin{cases} \mathbf{F} : \mathcal{X} \rightarrow \{\mathcal{C}^{(1)}, \cdots, \mathcal{C}^{(z)}, \cdots, \mathcal{C}^{(Z)}\} \\ \mathbf{F}(\mathbf{X}) = \mathcal{C}^{(z)} \end{cases} \tag{3.6}$$

The posterior model stored in $\mathcal{C}^{(z)}$ permits to perform a prediction by using for instance a maximum a posteriori:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}} \, P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}^{(z)}, \mathcal{P}) \tag{3.7}$$

To conclude, random ferns benefit of very fast learning and prediction and this, while having a compact structure. However, they can encounter some problems if no prior is available to "fill" empty cells. In the following part, we will briefly show that as random forests, random ferns can be used in an ensemble fashion to constitute strong learner.

3.1.3 Random Ferns Ensemble

As random forests, an ensemble $\mathcal{F}$ of T independant random ferns $\mathcal{F} = \{\mathbf{F}_1, \cdots, \mathbf{F}_t, \cdots, \mathbf{F}_T\}$ can constitute a strong learner. Since random ferns are not subject to any optimization procedure, they are strongly decorrelated and thus, don't need any further randomization step as bagging for instance. Each random fern $\mathbf{F}_t$ yields a random partition $\mathcal{P}_t$ of the feature space $\mathcal{X}$. During the prediction phase, the ferns ensemble can be considered as a function which associates an unseen observation $\mathbf{X}$ to an ensemble of cells:

$$\mathcal{F}(\mathbf{X}) = \left\{\mathcal{C}_1^{(z_1)}, \cdots, \mathcal{C}_t^{(z_t)}, \cdots, \mathcal{C}_T^{(z_T)}\right\} \tag{3.8}$$

Considering that each $\mathcal{P}_t$ is equiprobable, the ensemble prediction can be simply computed by averaging the tree posteriors:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{T} \sum_{t=1}^{T} P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}_t^{(z_t)}, \mathcal{P}_t) \tag{3.9}$$

As random ferns are constructed without any optimization, averaging seems to be the most robust prediction approach. However, exception handling has to be performed for predictions coming from "empty" cells. Either, all cells can be initialized using a prior distribution, or prediction from empty cells have to be discarded.

3.1.4 Random Ferns Parameters

Ensembles of random ferns offer a lot of freedom for the choice of different classes of splitting functions or posterior models. They possess only a few hyperparameters, the most important being: (1) the **number of ferns** and (2), the **fern depth**. Similarly as forests, increasing the number of ferns permits to average out noisy predictions, and thus corresponds in a monotonic decrease of the prediction error. The maximal depth of the fern is a crucial parameter that needs to be optimized as it directly impacts generalization. However, there is a major difference due to the "optimization-free" nature of

the random ferns. On one side, a fern needs to be much deeper than a tree to achieve a good partitioning of the data, and on the other side, they are less prone to overfitting as they do not explicitely fit the underlying data structure. Nevertheless, if a fern becomes to deep, then the risk of creating empty cells increases, and predictions may become less reliable. For this reason, the prediction error curve also decreases with the fern depth until it reaches a minimum and then increases again. This minimum corresponds to the optimal ferns depth, providing a good partitioning of the observations and a great generalization. In the following, we will shortly discuss how to derive random ferns ensemble for classification, regression and clustering tasks.

3.2 Random Ferns for Classification, Regression, Clustering

Similarly to random forests, random ferns permits to tackle several supervised and unsupervised tasks such as classification, regression and clustering. As random ferns usually do not have any optimization procedures, only posteriors in the cells are task specific. As the posteriors are defined is very similar to random forests, we give only a short reminder for the sake of completeness. We propose also to study the influence of the ferns parameters on a few toy examples.

3.2.1 Classification Ferns

We consider the input feature space $\mathcal{X} \subset \mathbb{R}^D$ and the output space $\mathcal{Y} \subset \mathbb{R}$ which is a finite set of K discrete values $\mathcal{Y} = \{y_1, \cdots, y_k, \cdots, y_K\}$. Our goal is to model the posterior probability distribution $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{X} \in \mathcal{X}$ and $\mathbf{Y} \in \mathcal{Y}$. Given a training set $\{(\mathbf{X}^{(n)}, Y^{(n)})\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, each fern of an ensemble $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^{T}$ permits to build a partition $\mathcal{P}_t$ over the input feature space $\mathcal{X}$. Considering the partition $\mathcal{P}_t = \{\mathcal{C}_t^{(z_t)}\}_{z_t=1}^{Z_t}$ built by the random fern $\mathbf{F}_t$, class posteriors can be estimated in each cell $\mathcal{C}_t^{(z_t)}$ of $\mathcal{P}_t$ as follows:

$$P(y_k|\mathbf{X} \in \mathcal{C}_t^{(z_t)}, \mathcal{P}_t) = \frac{|\left\{\mathbf{X}^{(n)} \in \mathcal{C}_t^{(z_t)}, \mathbf{Y}^{(n)} = y_k\right\}|}{|\left\{\mathbf{X}^{(n)} \in \mathcal{C}_t^{(z_t)}\right\}|} \tag{3.10}$$

Influence of ferns parameters: In this part, we propose to show the influence of the main ferns parameters, *e.g.* the fern depth and the number of ferns. Therefore, we will use the same 3 toy examples as for classification forests using the "cross", "sun" and "two moons" datasets (see fig.3.5). Remember that these three binary classification problems reveal a few challenges: non-linearly separable, multi-clusters classes, and noisy data points.

Let us start with a single fern, each node function selecting a random dimension and a random threshold, corresponding thus to axis-aligned splits. Remember that ferns are optimization-free so we do not need to set a number of function candidates. We vary only the depth of the fern between 5 and 15. We propose to plot the resulting posterior

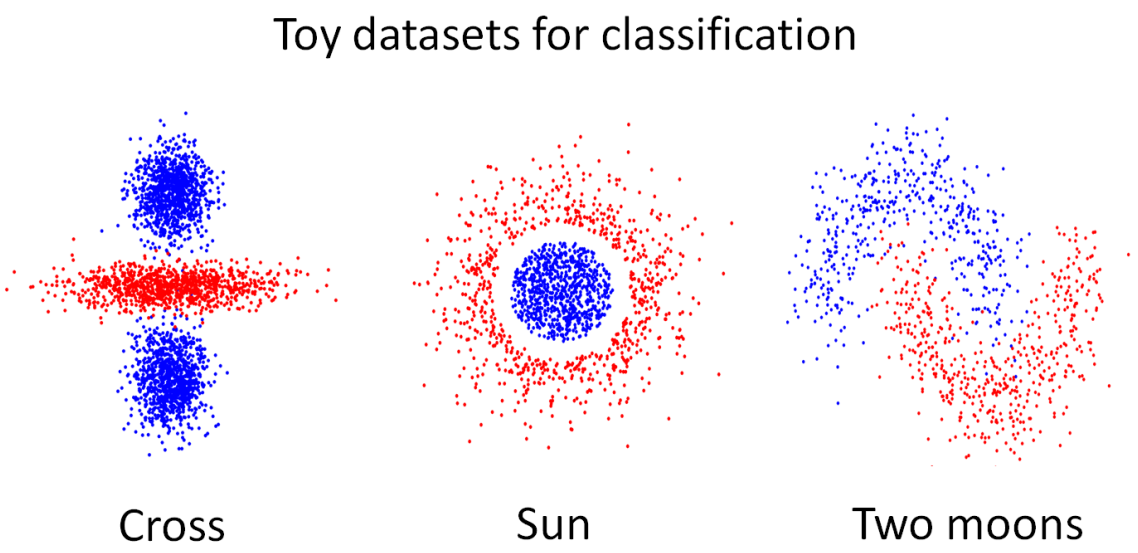

Figure 3.5: Classification toy examples: we propose to study the ferns behaviour on these 3 datasets

over the feature space using a color code varying from deep blue to red according to the posterior values for the blue and the red class.

As shown on fig.3.8, when the fern gains in depth, it builds a more complex partition yielding a posterior which better matches the underlying class distribution. Again, as there are no optimization, the partition construction is not making use of the data, which explains the high variability of a single fern. Thus, a single fern has to be in general deeper than a tree to approximate well the class posterior distribution. On the other side, even with a depth of 15, no signs of overfitting are visible, as noisy data points have no influence on the building of the partition. Nevertheless, with an increasing depth the risk of creating empty cells gets higher. Thus a good compromise has to be found for the ferns depth, as it moreover has a big influence on the generalization.

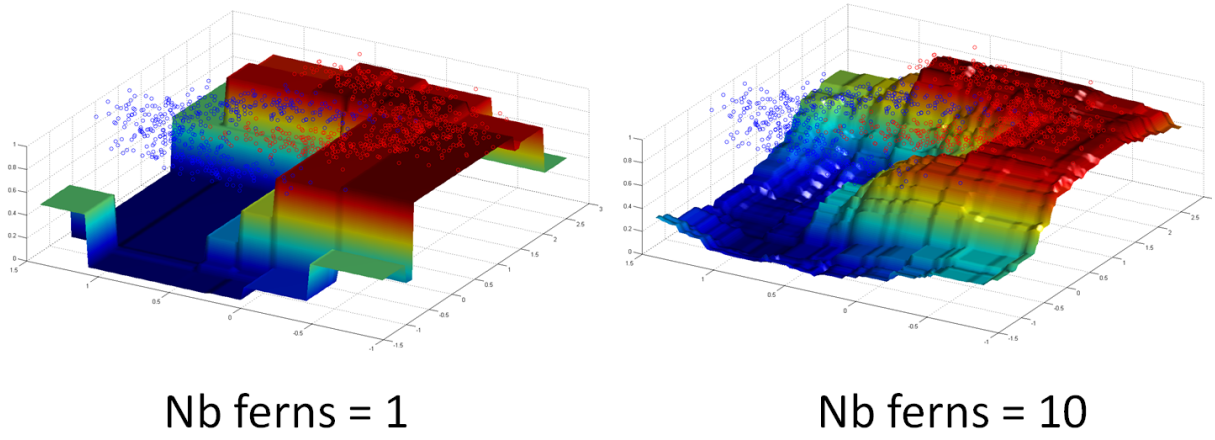

Figure 3.6: Classification posterior of a random ferns ensemble: Increasing the number of ferns provides a smoother posterior and permits to reach a greater generalization.

Now let us set the depth equal to 10 and vary the number of ferns. As illustrated by

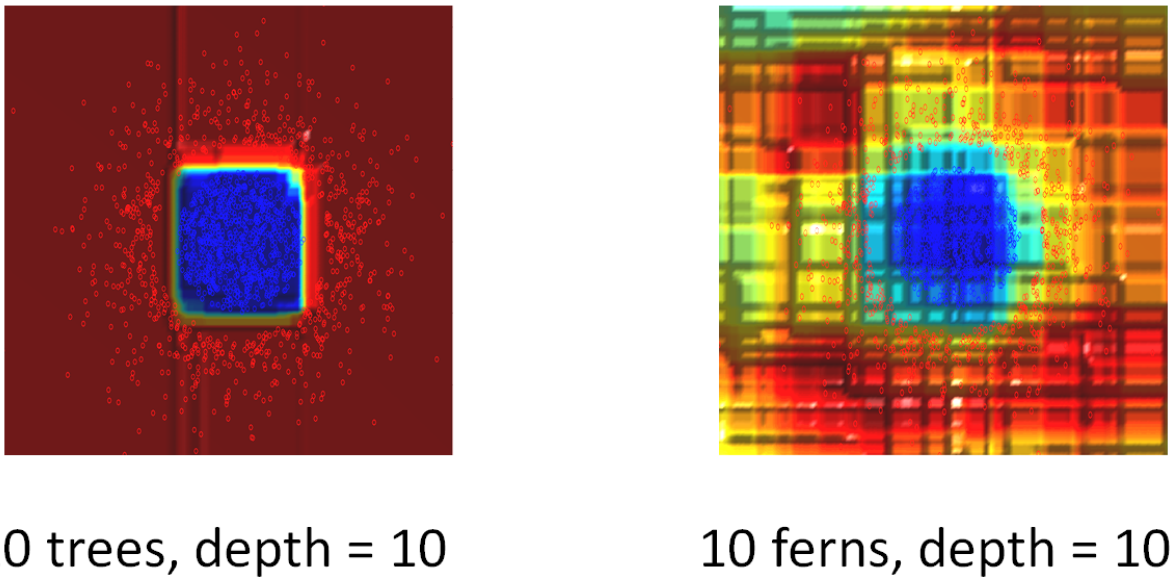

Figure 3.7: Comparison classification forest and ferns ensemble: for a depth of 10, trees have more sharper posteriors and ferns get smoother class boundaries.

fig.3.9 and 3.6, since ferns are not optimized, increasing the number of ferns is crucial as it permits to better fit the data, achieve better generalization and get smoother posteriors, *i.e.* smoother boundaries between the classes. While comparing the predictions of random ferns to forests (see fig.3.7), one can see that a forest provides sharper posteriors and spherical clusters can not be well fitted using axis-aligned splits. Due to their random nature, ferns show already smoother class boundaries even when using axis-aligned splits.

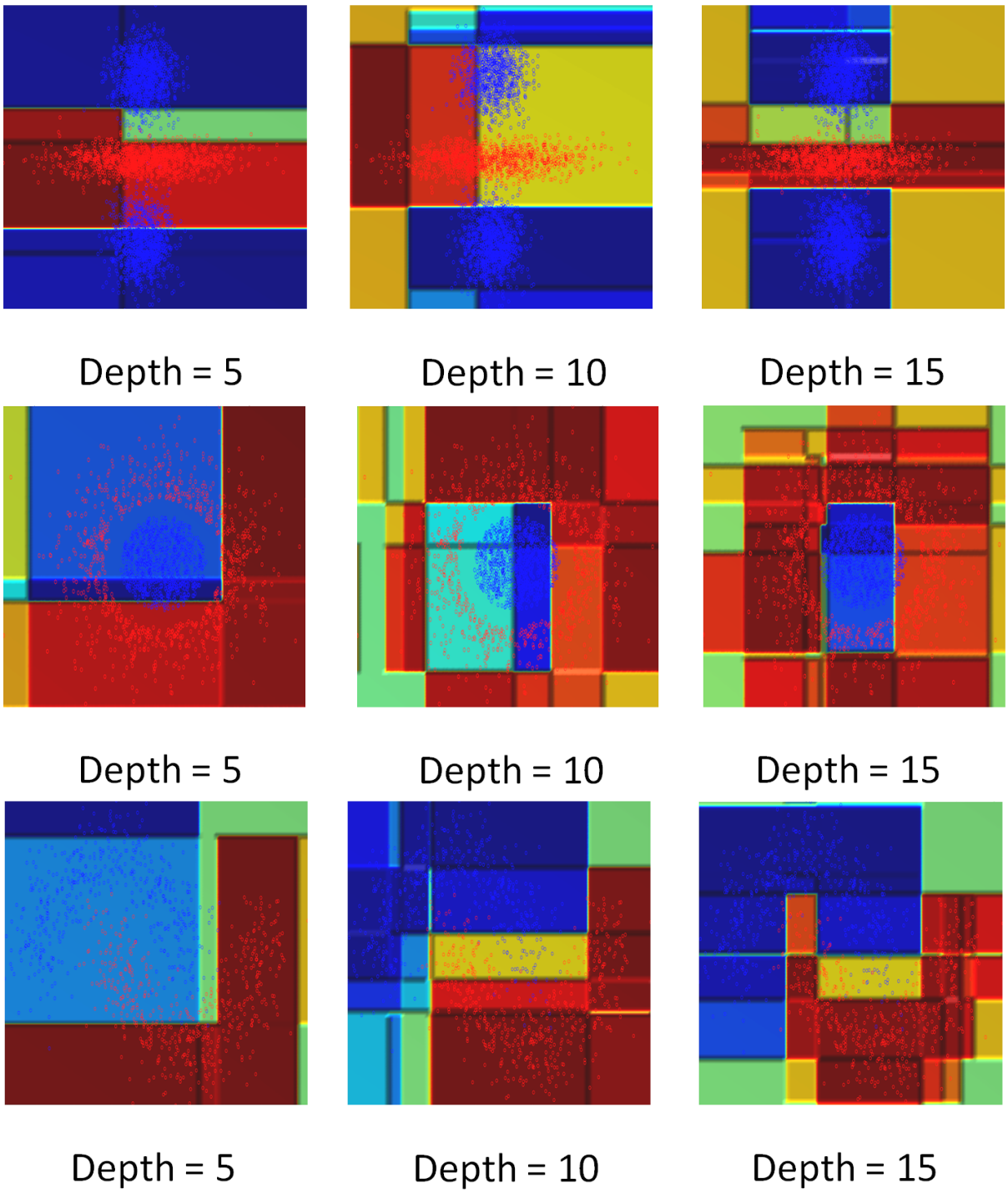

Figure 3.8: Classification posterior of a single random fern: we propose here to study the influence of the depth parameter.

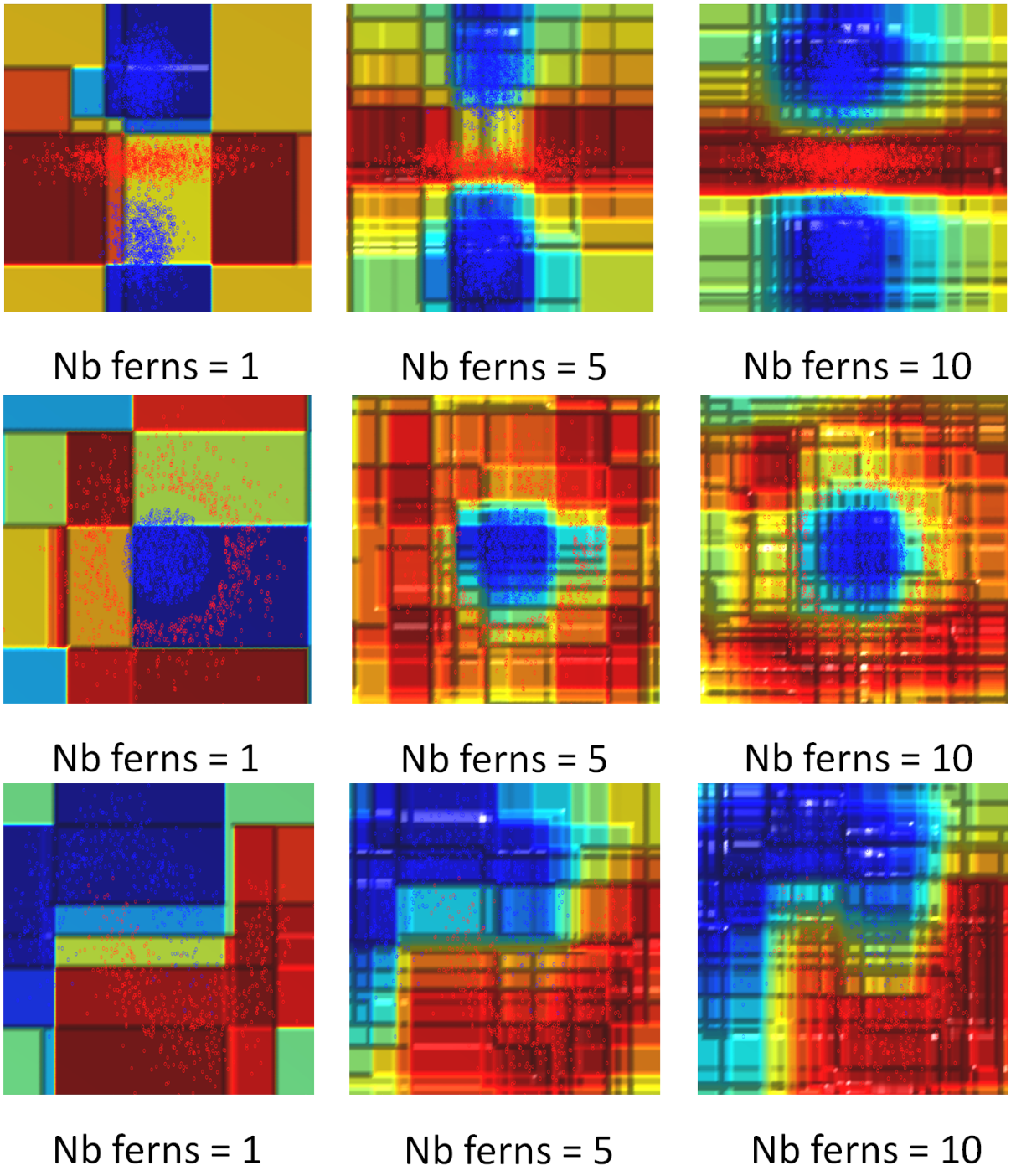

Figure 3.9: Classification posterior of a random ferns ensemble: here the depth is set to 10, and we propose to study the influence of the number of ferns.

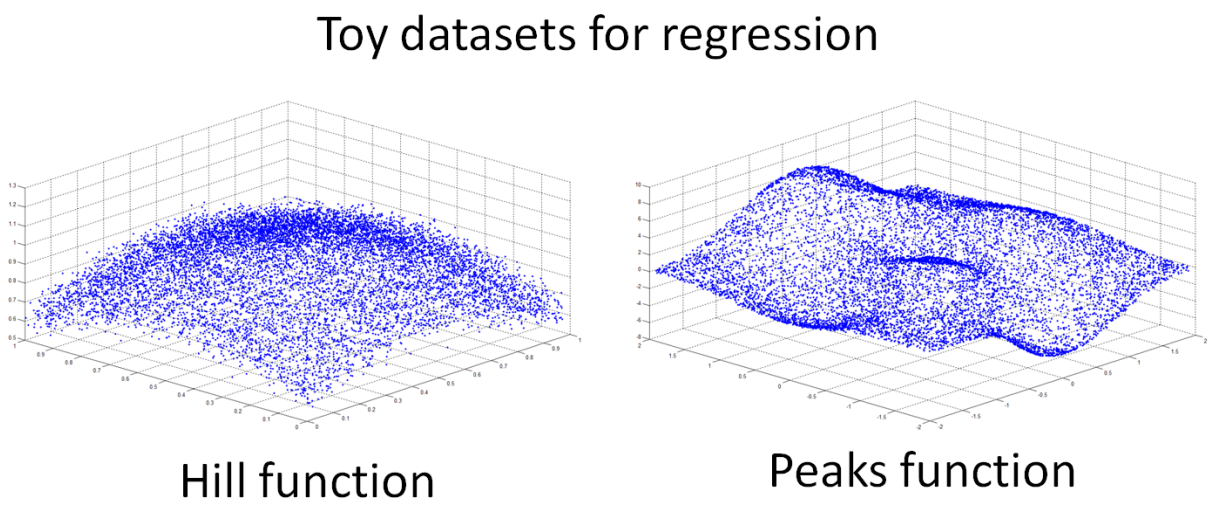

Figure 3.10: Regression toy examples: we propose to study the ferns ensemble behaviour on these 2 functions

3.2.2 Regression Ferns

Here we consider the input feature space $\mathcal{X} \subset \mathbb{R}^D$ and the output space $\mathcal{Y} \subset \mathbb{R}^{D'}$. Similarly we aim at modeling the posterior probability distribution $P(\mathbf{Y}|\mathbf{X})$. Given a training set $\{(\mathbf{X}^{(n)}, Y^{(n)})\}_{n=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, each fern of an ensemble $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^{T}$ builds a partition $\mathcal{P}_t$ over $\mathcal{X}$. If we consider the partition $\mathcal{P}_t = \{\mathcal{C}_t^{(z_t)}\}_{z_t=1}^{Z_t}$ built by the random tree $\mathbf{F}_t$, posteriors can be modeled in each cell $\mathcal{C}_t^{(z_t)}$ as:

$$P(\mathbf{Y}|\mathbf{X} \in \mathcal{C}_t^{(z_t)}, \mathcal{P}_t) = \mathcal{N}_t^{(z_t)}(\mathbf{Y} \mid \mu_t^{(z_t)}, \mathbf{\Sigma}_t^{(z_t)}) \tag{3.11}$$

$\mathcal{N}_t^{(z_t)}$ is a multivariate Gaussian with mean $\mu_t^{(z_t)}$ and covariance matrix $\mathbf{\Sigma}_t^{(z_t)}$ estimated in the output space $\mathcal{Y}$ from the subset of the training points that fall into the cell $\mathcal{C}_t^{(z_t)}$ of partition $\mathcal{P}_t$.

Influence of ferns parameters: In this part, we propose to show how ferns ensemble perform in regression tasks, in order to approximate arbitrary functions. We will demonstrate the influence of the main ferns parameters, *e.g.* the fern depth and the number of ferns. Therefore, we will use 2 toy examples using the two "hill" and "peaks" functions (see fig.3.10). These two datasets consist of 10000 points (x, y, z) generated using non-linear functions and additive noise. Remember that our input feature space is here represented by x, y and the output space by z.

Let us start with a single fern, where at each node a dimension and a threshold are chosen at random generating thereby axis-aligned splits. Again, here there is no optimization step. We vary the depth of the tree between 5 and 15. In each leaf, the regression posterior is modelled by a one-dimensional Gaussian distribution, where the mean and the variance are estimated from the training data. We propose to plot the resulting mathematical expectation over the feature space, and to overlay some points of the training set to show how well the predicted function fits the data points. Note that

the output or the random ferns in this configuration is an ensemble of piece-wise constant approximations of the input data.

As shown on fig.3.12, when the fern gains in depth, it provide a regression output which better fits the underlying function. However, as it is optimization-free, the size of the cells may not always be adapted to the variation in z. This explains why the extrema of the peaks function seems cut out when the fern is not deep enough. In the case of the "hill" function, we can clearly notice problems of empty cells where the predicted surface shows holes. This happens when the fern gets to deep and no prior information is available. Similarly as for trees, a good compromise has to be found for the fern depth as it has a big influence on the generalization.

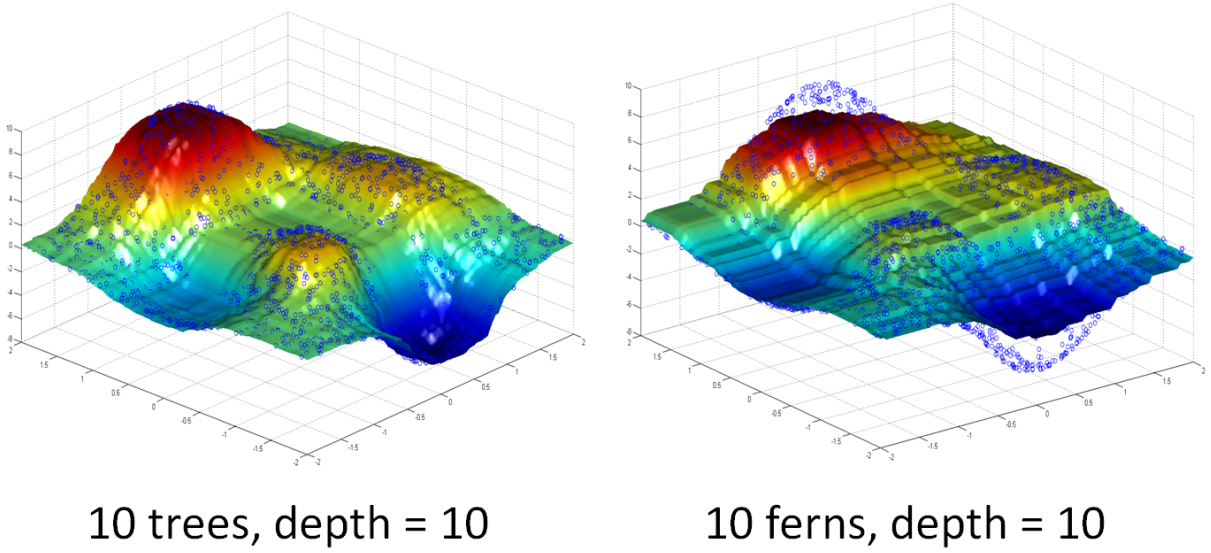

Figure 3.11: Comparison regression forest and ferns ensemble: Clearly, for a depth of 10, trees achieve better prediction than ferns, as ferns needs to be more deep to give a better approximation.

Now let us set the fern depth equal to 10 and vary the number of ferns. As illustrated by fig.3.13, increasing the number of ferns permits to get smoother regression output. Comparing the predictions of random ferns to regression forests (see fig.3.11), one can see that a forest provides a better fit for a comparable depth and number of trees. Indeed, random ferns need to be more deep to increase their prediction accuracy. Again, their partitions are not created according to the training set, and consequently they don't create smaller cells to better fit fast variations of the output. However, considering the fact that they are "optimization-free", predictions results are impressive.

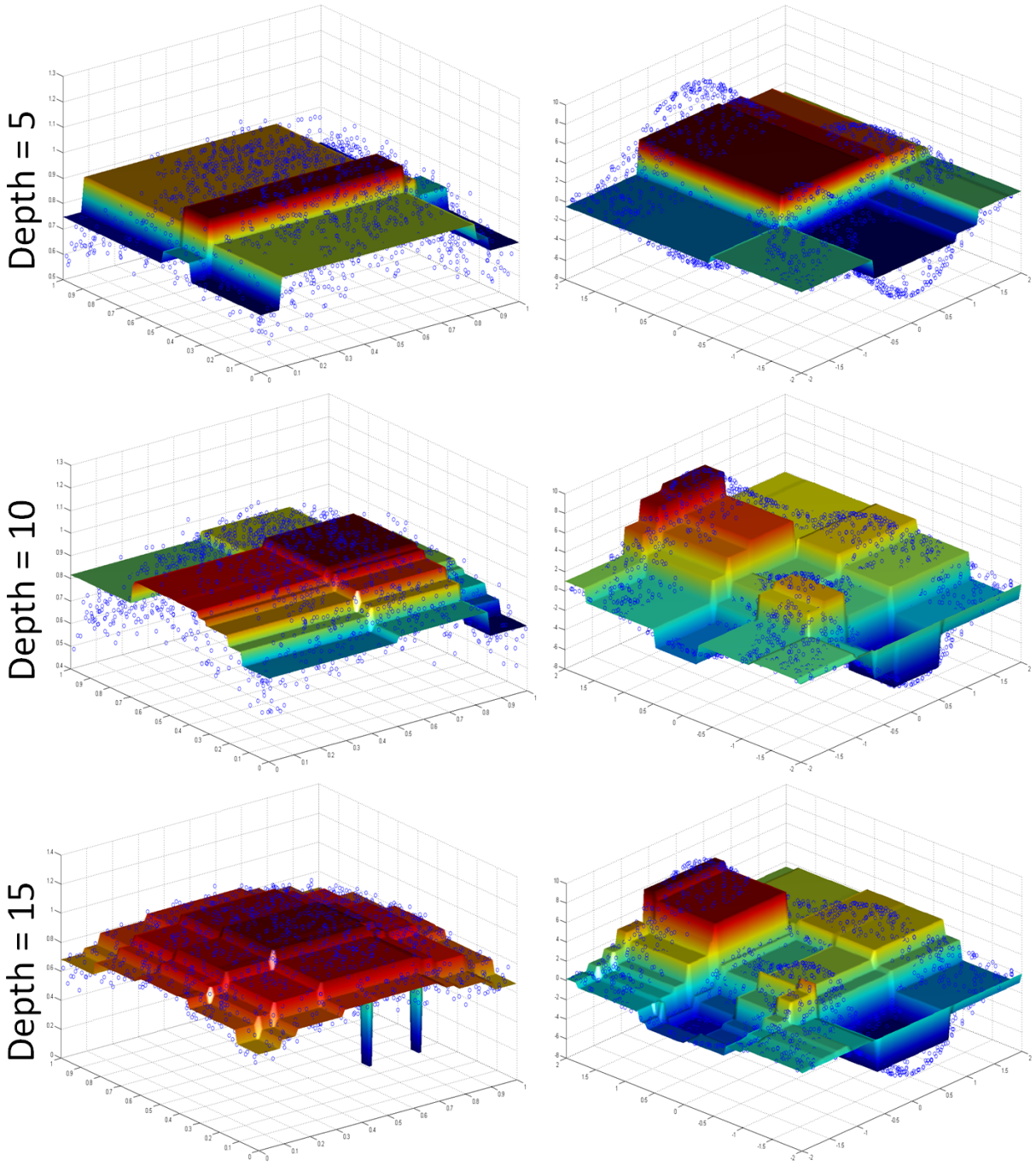

Figure 3.12: Regression output of a single random fern: we propose here to study the influence of the depth parameter.

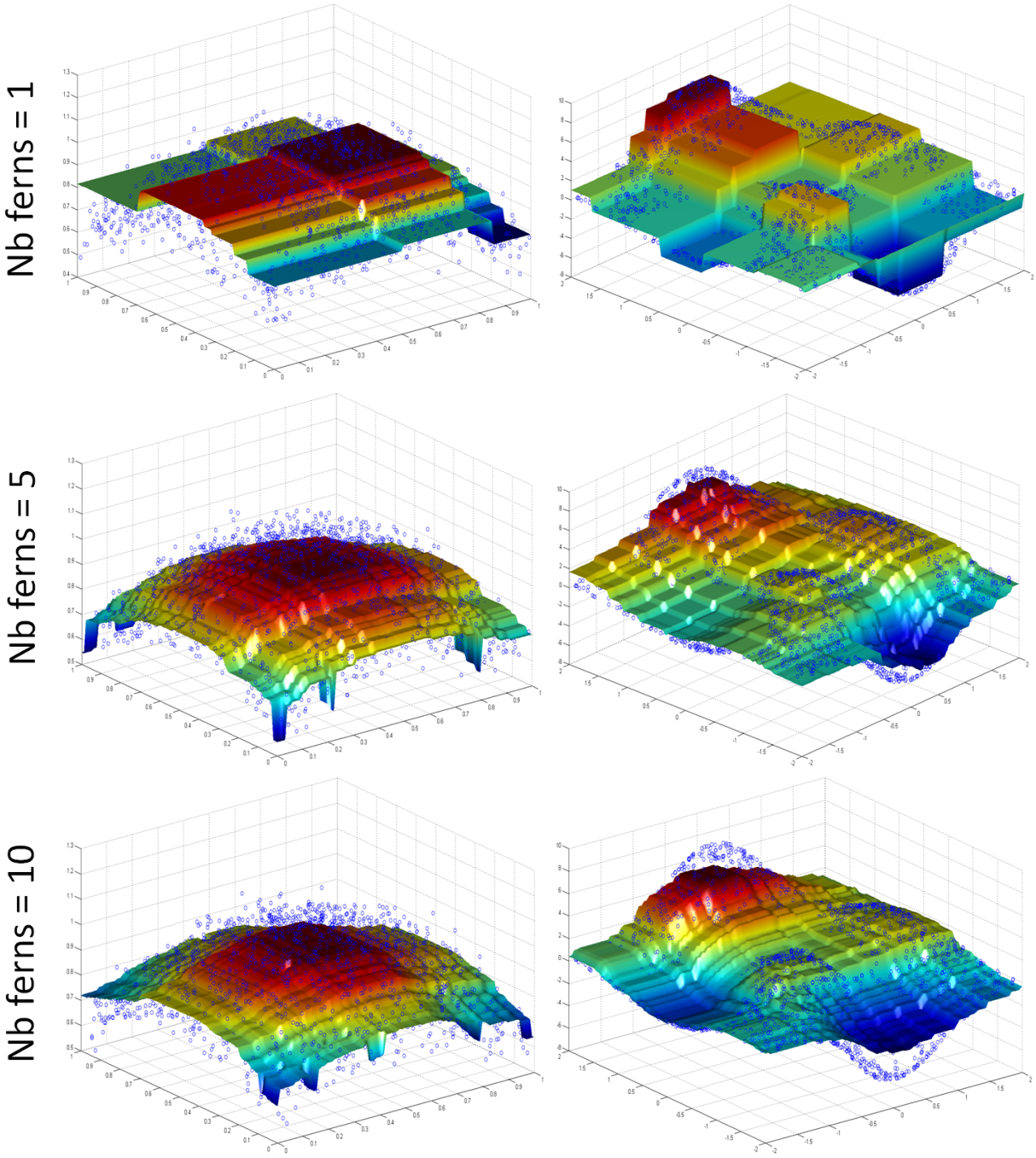

Figure 3.13: Regression output of a random ferns ensemble: here the depth is set to 10 and we propose to study the influence of the number of ferns.

3.2.3 Clustering Ferns

In the case of clustering, we only consider an input feature space $\mathcal{X} \subset \mathbb{R}^D$. As for random forests, each cell of the partitions induced by the random ferns are associated to a cluster. Hence, each random fern is used to map a point $\mathbf{X} \in \mathcal{X}$ to a cluster, and this happens by simply looking at the cell it falls in:

$$\mathbf{F}_t(\mathbf{X}) = \mathcal{C}_t^{(z_t)} \tag{3.12}$$

Thus, each observation is associated to a set of clusters coming from different partitioning results of the same feature space. These multiple clustering results can be merged into one global clustering using the 2 approaches presented in the previous chapter which are: (1) perform **intersection** between the different partitions to create a global partition and thereby global clusters, (2) keep the vectors of cells as an **implicit** representation of the global clusters.

3.3 Conclusion

In this section, we presented an efficient variant of the random forests, which can be derived for many learning tasks such as classification, regression and clustering. Since their introduction, they have been always seen as ensemble of constrained trees. However, to fully understand the nature of random ferns and their properties, the best interpretation is to consider each fern as an **intersection of decision stumps**. This permits to better identify their advantages and pitfalls. Indeed, they benefit of a very compact structure and are usually "optimization-free" learner. However, since they are constructed as an intersection of decision stumps, some cells may stay unpopulated during the learning phase, in contrast to random trees. If no prior information is available to "fill" these empty cells, then problems may be encountered if observations fall in these "black holes". Moreover, due to their highly randomized nature, they need to be much deeper than trees to reach same performance, and this, even more if the feature space is high-dimensional or contains uninformative features. On the other hand, they are more robust to noisy features as their optimization-free learning provide dependence from the training set.

CHAPTER

FOUR

RANDOM FORESTS: CONTRIBUTIONS IN MEDICAL APPLICATIONS

"Medicine makes people ill, mathematics makes them sad [...]"
Martin Luther

In this chapter, we report our forests-related contributions in different medical imaging applications. First, we present an efficient regression approach based on random ferns and forests to estimate the position and the size of multiple organs of interest in whole-body multi-channel MR scans. Further, we propose to tackle the problem of multiple organ segmentation using a novel joint classification-regression random forest model. Through exhaustive experimentations, we demonstrate that this joint formulation yields better results than classification by learning spatial smoothness directly from the data. In the context of early diagnosis of Parkinson's disease, we introduce a novel paradigm to detect Parkinson-related lesions within the midbrain using 3D transcranial ultrasound. Two forest models are designed to capture visual as well as spatial information, the latest being encoded using a novel parametrization that accounts for asymmetric changes of scales and orientation of the midbrain anatomy. Afterward, we report our work on modality recognition based on the visual content of a medical image. To this end, we use random ferns clustering to build efficiently a dictionary of visual words, and demonstrate on a real database of medical images the advantages of our approach in terms of speed and accuracy. Finally, we introduce a new ensemble approach called STARS: Several Thresholds on a Random Subspace. Motivated by the fact that using multiple decisions at each node instead of relying on binary decisions may be beneficial in the case of complex non-linear clusters, STARS can be seen as ensemble of multiple-decision stumps. Applied to the task of modality recognition, they provide better results than hierarchical clustering and random ferns.

4.1 Multiple Organ Detection and Localization in multi-channel Magnetic Resonance scans

Automatic localization of multiple anatomical structures in medical images provides important semantic information with potential benefits to diverse clinical applications. In the current section, we describe an efficient approach for estimating location and size of multiple anatomical structures in MR scans which has been published in [73]. Our contribution is three-fold: (1) we apply supervised regression techniques to the problem of anatomy detection and localization in whole-body MR, (2) we adapt random ferns to produce multi-dimensional regression outputs and compare them with random regression forests, and (3) introduce the use of 3D LBP descriptors in multi-channel MR Dixon sequences. The localization accuracy achieved with both fern- and forest-based approaches is evaluated by direct comparison with state of the art atlas-based registration, on ground-truth data from 33 patients. Our results demonstrate improved anatomy localization accuracy with higher efficiency and robustness.

4.1.1 Introduction

Following the success of combined PET/CT, the possibility of combining PET with MRI has gained increased interest, as significant advantages are expected compared to PET/CT for many imaging tasks in neurology, oncology and cardiology [47]. However, before its introduction in the clinical practice, a technical challenge impacting the quality of PET/MR imaging needs to be solved: the attenuation correction of 511 keV photons according to the radiodensity of the tissues. While in PET/CT [51], radiodensity information provided by CT at X-ray energies can be converted into attenuation information, MR does not provide any information on the tissue density. Therefore, methods have been investigated to generate an attenuation correction map directly from MR. For brain imaging, atlas-based solutions using registration were evaluated in [52, 43]. For whole-body imaging, different approaches based on the classification of tissues into 4 classes (background, lungs, fat, and soft tissue) have been investigated, for instance in [62]. While previous methods showed promising results for attenuation correction of whole body imaging with PET/MR, they propose only a coarse tissue classification, not accounting for organ-specific attenuation and for the attenuation introduced by bones. To further improve the quality of whole-body PET data reconstruction, we aim at generating organ-specific attenuation information directly from MR. Therefore, the position of the organs which impact the attenuation of photons need to be known. In this section, we present a novel regression approach for simultaneously localizing multiple organs in multi-channel whole-body MR. In fact, we propose a strategy based on random ferns for efficient regression and compare them to random regression forests. Experiments on 33 patient scans demonstrate better performance than atlas-based techniques in terms of accuracy, speed, and robustness.

4.1.2 Related Work

Classical object detection algorithms are based on sliding windows and classifiers whose role is to predict whether a voxel belongs to the object of interest or not. In [100], Viola and Jones introduced a fast detection approach based on a cascade of classifiers trained using Adaboost. Built as a succession of classifiers taking sequentially more and more features into account, this approach achieved impressive performance for real-time face detection.

In medical applications, there has been an increasing interest in regression-based solutions for organ localization. Since the human body consists of a specific arrangement of organs and tissues, it can be expected that voxels, based on their contextual information, can predict the surrounding anatomy. For instance, if the neighborhood of a voxel shows an appearance which is typical of heart tissue, besides the position of the heart, this voxel can provide an estimate of position of the nearby lungs. In [112], Zhou *et al.* introduced an approach based on boosting ridge regression to detect and localize the left ventricle (LV) in cardiac ultrasound 2D images. There, the learned function predicts the relative position, scale and orientation of the LV based on Haar-like features computed on 2D images. Impressive results are demonstrated on echocardiogram sequences. To detect and localize the heart chambers in 3D cardiac CT, Zheng *et al.* proposed in [109] an approach called marginal space learning (MSL). To break down the complexity of learning directly in the full 3D similarity transformation space, the authors demonstrate that training a classifier on projections of the original space effectively reduces the search space. Using this idea, they build a cascade of classifiers based on probabilistic boosting tree (PBT) to predict first the position, then the position-orientation and finally the full 3D pose. In [110], the authors push this idea further to non-rigid marginal space learning using statistical shape models. Although these approaches have shown very good performance on CT scans, building such a cascade of classifiers is a computationally intensive learning procedure which requires large training sets.

In this work, we avoid intensive training by building a single regressor predicting simultaneously the position of multiple organs. In [21], Criminisi *et al.* proposed a regression approach based on random forests for the localization of organs in 3D CT scans. The authors showed that their method achieves better performance than atlas registration, and this, while benefiting of fast training and testing. While in [21], the authors could rely on absolute radiodensity values provided by CT, here, we deal with MR images which provide only relative values and suffer from field inhomogeneities. To tackle this challenging problem, we adapt the regression forest framework by introducing 3D LBP descriptors. Additionally, we implement a random ferns regression approach and compare it with forests. Both regression techniques are evaluated and compared to an atlas-based registration approach.

4.1.3 Proposed Method

This section describes details of our organ detection and localization approach. First, we cast this problem as a regression task. Second, we introduce our feature representation based on water and fat channels computed from MR Dixon sequences. Third, we present

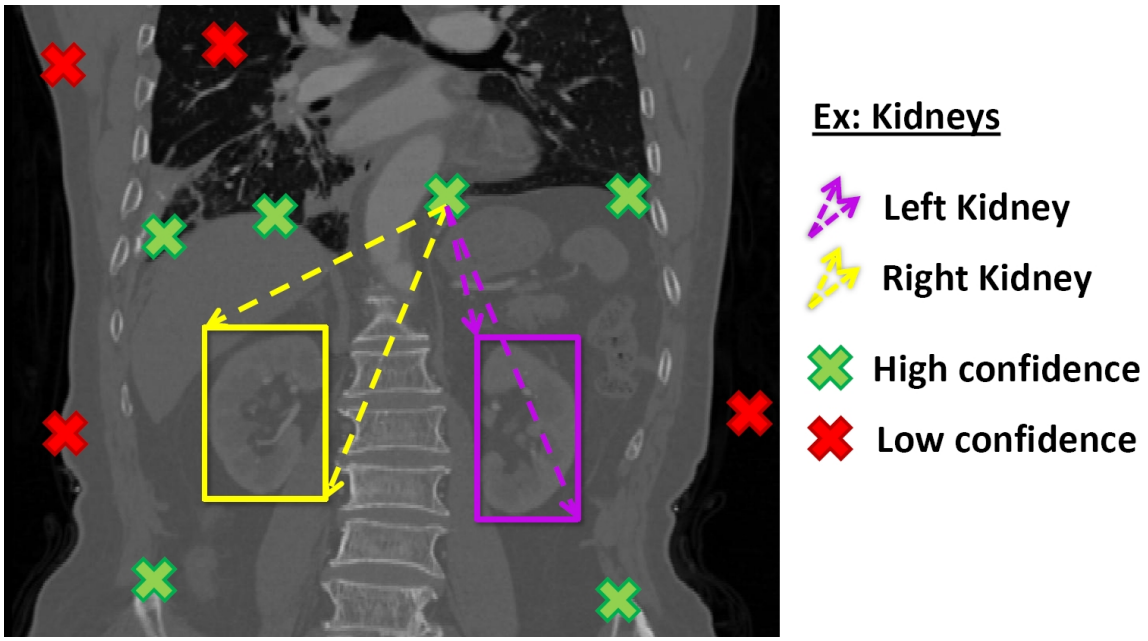

Figure 4.1: Organ Localization Approach: Learn a probabilistic mapping from voxels to organ bounding boxes

our regression strategy using ferns and forests. Finally, we show how to combine voxel predictions to localize all organs of interest in one shot.

4.1.3.1 Problem Statement

In the context of MR Dixon sequences, we are given two MR channels, i.e. the water and fat channels represented by the two intensity functions $\mathbf{I}^{(\text{water})} : \Omega \rightarrow \mathbb{R}$ and $\mathbf{I}^{(\text{fat})} : \Omega \rightarrow \mathbb{R}$, where $\Omega \subset \mathbb{R}^3$ is the image domain. Considering a set of K organs of interest, their location within a patient scan can be represented by a set of bounding boxes $\mathcal{O} = \{\mathbf{O}_1, \cdots, \mathbf{O}_k, \cdots, \mathbf{O}_K\}$, where each 3D bounding box $\mathbf{O}_k$ contains one organ and is parametrized as a vector $\mathbf{O}_k = [x_k^0,\ y_k^0,\ z_k^0,\ x_k^1,\ y_k^1,\ z_k^1]$. Now given the water and fat channels from an unseen patient, the goal of multiple organ localization is to estimate simultaneously the parameters of the different bounding boxes containing the organs of interest.

In our framework, we propose a probabilistic regression strategy in which each voxel $\mathbf{x} \in \Omega$ votes for the ***relative offsets*** to all organs bounding boxes. We denote by $\mathbf{Y} = [\mathbf{Y}_1, \cdots, \mathbf{Y}_k, \cdots, \mathbf{Y}_K]$ the vector containing ***all*** relative offsets between voxel location $\mathbf{x} = [x,\ y,\ z]$ and the different bounding boxes, where each component $\mathbf{Y}_k$ is defined as:

$$\mathbf{Y}_k = \left[x_k^0 - x,\ y_k^0 - y,\ z_k^0 - z,\ x_k^1 - x,\ y_k^1 - y,\ z_k^1 - z\right] \tag{4.1}$$

On fig.4.2, these relative displacements between one voxel $\mathbf{x}$ and the heart or liver bounding box are represented by the arrows. Here, we consider the following organs: head, left lung, right lung, heart and liver.

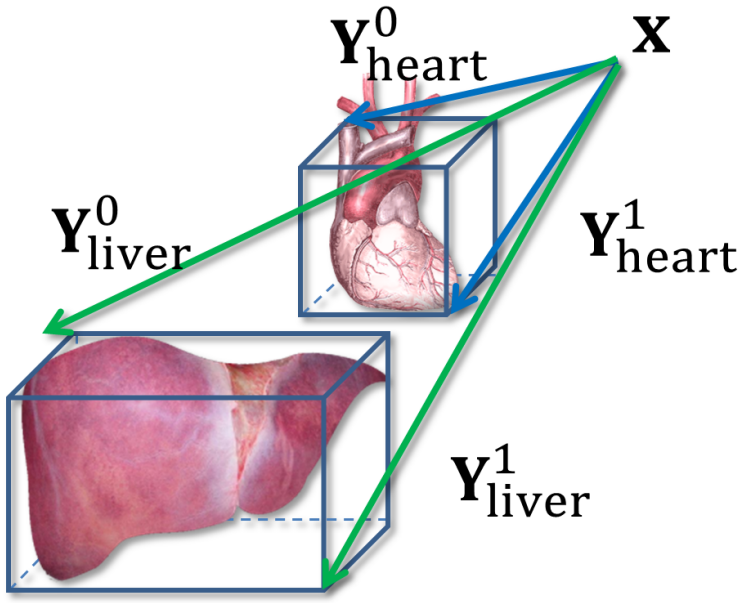

Figure 4.2: Voxel predictions: Relative displacements from a voxel to the bounding boxes of all organ of interest

In a probabilistic fashion, we aim at modeling the probability distribution $P(\mathbf{Y}|\ \mathbf{x}, \mathbf{I}^{(\text{water})}, \mathbf{I}^{(\text{fat})})$. The contribution of each voxel to the position of all organ bounding boxes can be then estimated using either the maximum a posteriori:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}}\, P(\mathbf{Y}|\ \mathbf{x}, \mathbf{I}^{(\text{water})}, \mathbf{I}^{(\text{fat})}) \tag{4.2}$$

or the mathematical expectation:

$$\hat{\mathbf{Y}} = \int_{\mathbf{Y}} \mathbf{Y}.P(\mathbf{Y}|\ \mathbf{x}, \mathbf{I}^{(\text{water})}, \mathbf{I}^{(\text{fat})})d\mathbf{Y} \tag{4.3}$$

While individual votes will produce very noisy predictions, their probabilistically weighted combination will produce an accurate output (see fig.4.1). Now, in such high-dimensional spaces, modeling the posterior distribution $P(\mathbf{Y}|\ \mathbf{x}, \mathbf{I}^{(\text{water})}, \mathbf{I}^{(\text{fat})})$ directly is very challenging. Therefore we propose to use a random ferns regression approach. Following a "divide" and "conquer" strategy, they provide efficient piecewise approximations of any distribution in high-dimensional spaces by: (1) partitioning the space using simples decisions, and (2) estimating the posterior in each "cell" of this space. Before we explain in more details our random ferns regression approach, we describe in the next part the new features we introduce to characterize the visual context of a voxel $\mathbf{x}$ using both the fat and water channels from the MR Dixon sequence.

4.1.3.2 Feature Representation

As described in [58], MR Dixon imaging techniques are based on the one shot acquisition of a so-called "in phase" scan where water and fat signals are in-phase and an "opposite

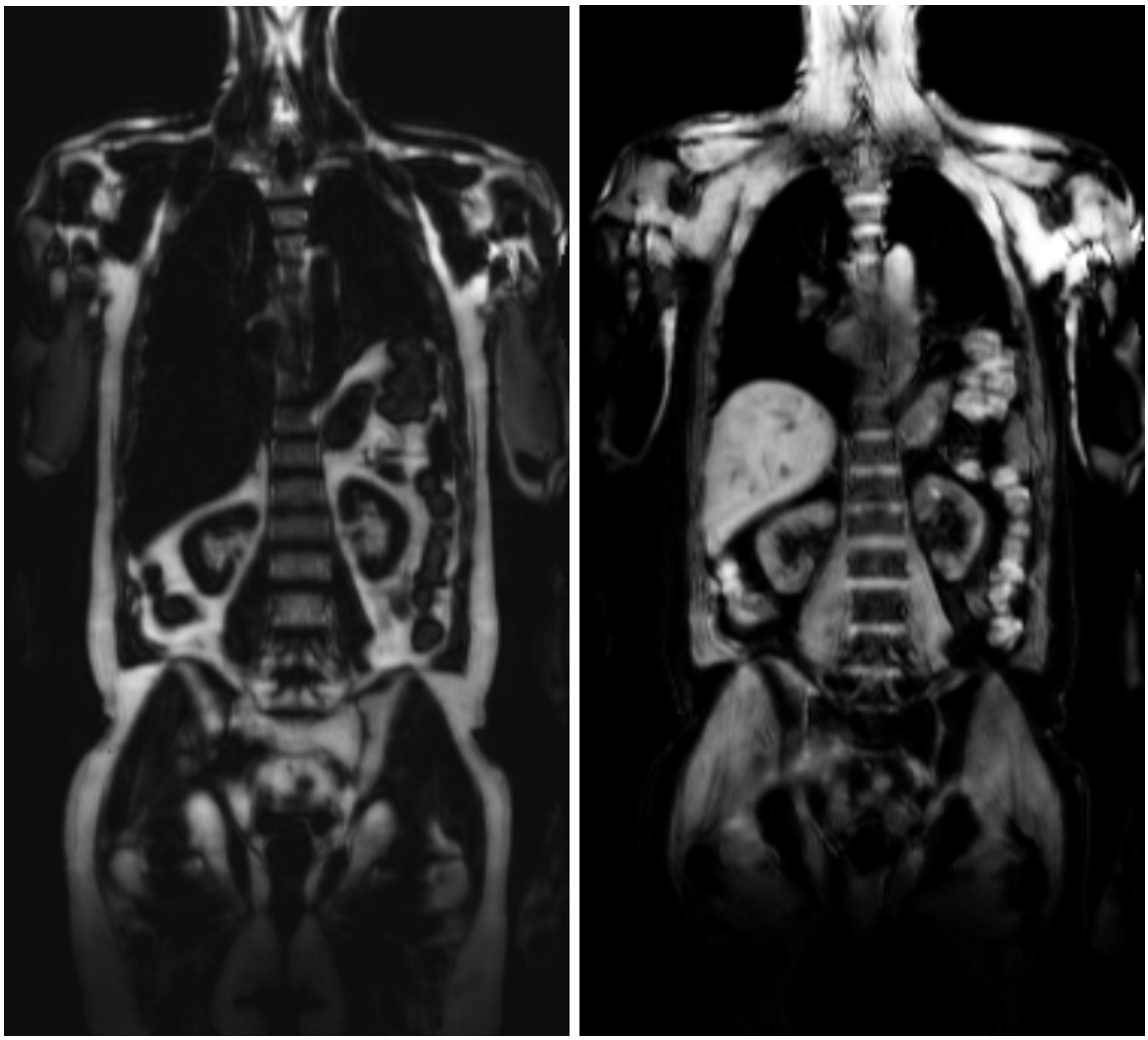

Figure 4.3: MR Dixon sequence: such a protocol permits to generate two MR channels, namely the "fat" and "water" weighted scans.

phase" scan where water and fat signals are 180° out-of-phase. Using these two scans from the same patient, water and fat signals can be separated to construct a water $\mathbf{I}^{(\text{water})}$ and a fat $\mathbf{I}^{(\text{fat})}$ channel as shown on fig.4.3. Since these 2 channels are perfectly registered, we propose to take advantage from their complementary nature and design a feature representation based on both water and fat information. While in CT intensity information is directly related to the underlying tissue distribution, MR intensity information is not absolute and suffers from variability between different images. For this reason, we will not rely on intensities as in [21], but on textural information by employing Local Binary Patterns (LBP) [68]: we propose to extract textural context variations at different scales (see fig. 4.4).

Let us consider a 3D region $\mathcal{R}^s_{\mathbf{x}}$ at scale s centered on voxel location $\mathbf{x}$ and a set $\{\mathcal{N}^{s,q}_{\mathbf{x}}\}_{q=1}^{Q}$ of Q 3D asymmetric cuboidal regions having different sizes, orientations and offsets in the neighborhood of $\mathbf{x}$. Using this, we can extract two binary feature vectors

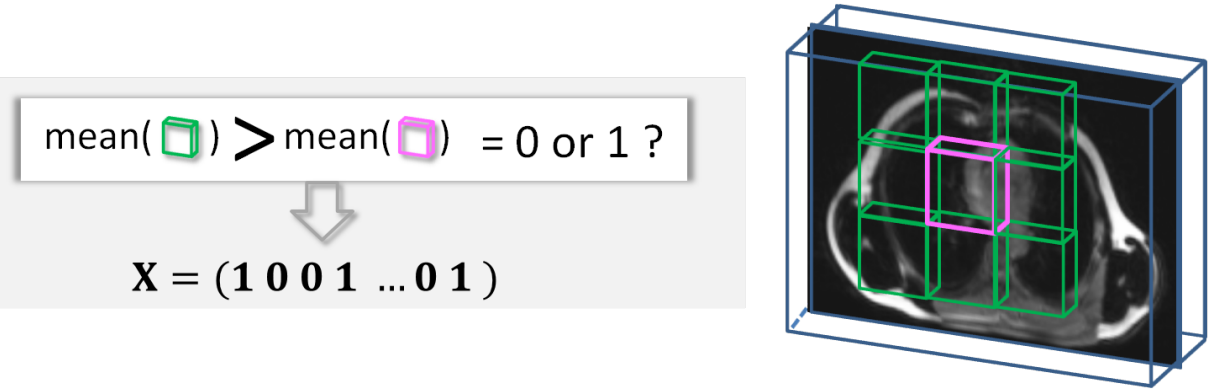

Figure 4.4: 3D LBP multi-scale features: Mean intensities of neighboring regions are compared and encoded into a binary feature vector.

$\mathbf{X}_s^{(\text{water})}$ and $\mathbf{X}_s^{(\text{fat})}$ from the two channels where each entry is the result of the following binary test comparing average intensities within regions $\mathcal{N}_{\mathbf{x}}^{s,q}$ and $\mathcal{R}_{\mathbf{x}}^{s}$:

$$\mathbf{X}_s^{(i)}[q] = \frac{1}{|\mathcal{N}_{\mathbf{x}}^{s,q}|} \sum_{\mathbf{x}' \in \mathcal{N}_{\mathbf{x}}^{s,q}} \mathbf{I}^{(i)}(\mathbf{x}') < \frac{1}{|\mathcal{R}_{\mathbf{x}}^{s}|} \sum_{\mathbf{x}' \in \mathcal{R}_{\mathbf{x}}^{s}} \mathbf{I}^{(i)}(\mathbf{x}'), \tag{4.4}$$

and this, $\forall q \in \{1, \cdots, Q\}$ and $i \in \{\text{water}, \text{fat}\}$. Repeating this operation at several scales results in two feature vectors $\mathbf{X}^{(\text{water})}$ and $\mathbf{X}^{(\text{fat})}$ describing the multi-scale textural context for both channels in the neighborhood of voxel location $\mathbf{x}$. Since $\mathbf{X}^{(\text{water})}$ and $\mathbf{X}^{(\text{fat})}$ are binary vectors, they can be further encoded to reduce their dimensionality. Finally, they are concatenated in one feature vector: $\mathbf{X} = \left[\mathbf{X}^{(\text{water})}, \mathbf{X}^{(\text{fat})}\right]$.

4.1.3.3 Ensemble Regression Approaches

This section explains how we use ferns and forests to efficiently approximate the posterior distribution $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{X}$ represents the visual context of voxel $\mathbf{x}$ given the two channels $\mathbf{I}^{(\text{water})}$ and $\mathbf{I}^{(\text{fat})}$. While regression forests have been used for detecting organs in CT [21], there exists little work on ferns-based regression. In [26], Dollar *et al.* use a ferns-based regressor in a cascade fashion for pose detection of objects in 2D images. In contrast, we use a single ensemble regressor which permits to capture information on the position of all organs of interest.

Piece-wise Regression: We assume a training set $\left\{(\mathbf{X}^{(n)}, \mathbf{Y}^{(n)})\right\}_{n=1}^{N}$ computed over a set of M patient MR volumes. To efficiently approximate the distribution $P(\mathbf{Y}|\mathbf{X})$, we propose to use random ferns to first subdivide the input feature space by building a partition $\mathcal{P}$ over it. After subdividing the feature space, we obtain cells containing data points which are easier to model even with simple mathematical models such linear or constant functions. As illustrated by the low-dimensional toy example on Fig. 4.5, the combination of these models over the whole partition results then in a complex non-linear model. Formally, $\mathcal{P}$ is defined as an ensemble of Z cells $\mathcal{P} = \left\{\mathcal{C}^{(z)}\right\}_{z=1}^{Z}$. With $\mathcal{P}$ given,

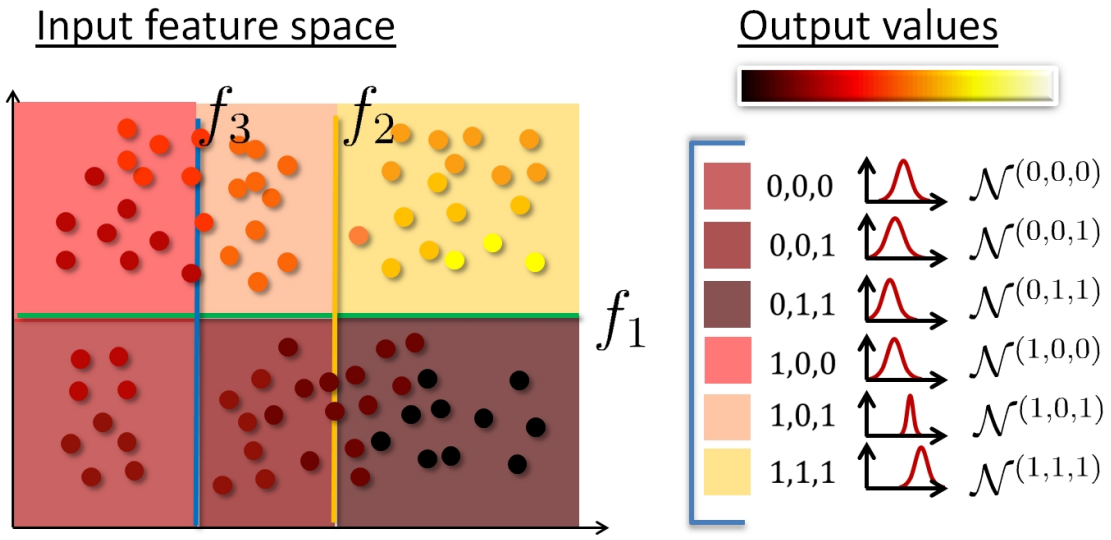

Figure 4.5: Random Ferns Regression: The data samples are associated to a color value in the output space. The lines represent the splitting functions that create a partition over the input feature space. In each cell, simple models are fitted to the points. Their combination over the full space results in a complex non-linear predictor

we propose to model the posterior in each cell $\mathcal{C}^{(z)}$ as follows:

$$p(\mathbf{Y}|\mathbf{X} \in \mathcal{C}^{(z)}, \mathcal{P}) = \mathcal{N}^{(z)}(\mathbf{Y}|\mu^{(z)}, \Sigma^{(z)}) \tag{4.5}$$

where $\mathcal{N}^{(z)}$ is a multivariate Gaussian distribution whose parameters are estimated during the training phase. In fact, this choice permits to model the full distribution as a piecewise Gaussian distribution. In contrast to fitting a Gaussian mixture model, partitioning is here performed in the input feature space and not in the output space. Based on this, we can model the probability distribution of $\mathbf{Y}$ over the full feature space according to partition $\mathcal{P}$. Clearly, the quality of the posterior approximation depends on the partition $\mathcal{P}$. If its number of cells Z is low, then the posterior approximation will be very rough. On the other hand, if Z is high, each cell will include few training points. In this case, the partition $\mathcal{P}$ tends to overfit the training data and suffers from poor generalization. To achieve better generalization, we construct multiple independent partitions $\{\mathcal{P}_t\}_{t=1}^{T}$ using an ensemble of random ferns. The posterior estimates from the different partitions of the ensemble are then combined using averaging.

Training/Testing: During the training of a fern, the whole training data is used at each node. This is in contrast to trees where only a subset is considered at each node. If we consider again the training set $\left\{(\mathbf{X}^{(n)}, \mathbf{Y}^{(n)})\right\}_{n=1}^{N}$ computed over a set of different patient scans, all feature vectors are pushed through the ferns ensemble and fall into the cells of the different partitions. Finally, the parameters of each Gaussian can be estimated for each cell $\mathcal{C}_t^{(z_t)}$ using the subset $\left\{\mathbf{Y}^{(n)}|\mathbf{X}^{(n)} \in \mathcal{C}_t^{(z_t)}\right\}_{n=1}^{N}$ of training data that fell into $\mathcal{C}_t^{(z_t)}$. In this work, we do not use optimization in the construction of our ferns regressor, *i.e.* the

splitting functions are chosen randomly. While this permits to have a very fast training procedure, it provides independence from the training set. This can be an advantage for instance in the case of noisy data. Once the training has been performed, all node functions and thresholds are frozen. During the test phase, an unseen data point $\mathbf{X}$ is pushed through the whole ensemble until it reaches a cell in each partition. Then, each cell contributes to the final prediction using its stored Gaussian model as seen in section 4.1.3.1. Next, we describe how to combine the predictions to localize all organs of interest.

4.1.3.4 Anatomy localization

Let us consider the water $\mathbf{I}^{(\text{water})}$ and fat $\mathbf{I}^{(\text{fat})}$ channels of an unseen patient. From both channels, a set of feature vectors $\left\{\mathbf{X}^{(n)}\right\}_{n=1}^{N}$ is extracted from voxel locations $\left\{\mathbf{x}^{(n)}\right\}_{n=1}^{N}$. By pushing this set of feature vectors through the regression ensemble, predictions $\left\{\hat{\mathbf{Y}}^{(n)}\right\}_{n=1}^{N}$ are computed as described in section 4.1.3.1. They correspond to the relative displacements $\hat{\mathbf{Y}}^{(n)} = \left[\hat{\mathbf{Y}}_1^{(n)}, \cdots, \hat{\mathbf{Y}}_k^{(n)}, \cdots, \hat{\mathbf{Y}}_K^{(n)}\right]$ between each location $\mathbf{x}^{(n)} = \left[x^{(n)}, y^{(n)}, z^{(n)}\right]$ and all organ bounding boxes $\mathcal{O} = \{\mathbf{O}_1, \cdots, \mathbf{O}_k, \cdots, \mathbf{O}_K\}$. The bounding box of organ $\mathbf{O}_k$ can be finally estimated as follows:

$$\mathbf{O}_k = \sum_{n=1}^{N} w_n \left(\hat{\mathbf{Y}}_k^{(n)} + \left[\mathbf{x}^{(n)}, \mathbf{x}^{(n)}\right]\right) \tag{4.6}$$

where each w_n weights the contribution of voxels according to the confidence of their predictions. Note that $\sum_{n=1}^{N} w_n = 1$. In this work, we discard contributions having low confidence and perform averaging on the remaining predictions.

4.1.4 Experiments and Results

In this section, we compare our approach based on regression ferns with regression forests and the current state-of-the-art multi-atlas registration.

Data: Our dataset currently consists of scans from 33 patients who underwent a 3-Tesla whole-body MR Dixon sequence. All patients have cancer (mostly neck, lung, liver cancer) and show a high variability in their anatomy partially due to their disease. For the detection and localization of organs, we use the water and fat channels. In each scan, we manually delineated the bounding boxes for following organs: head, left lung, right lung, liver and heart. The size of the volumes are $192 \times 124 \times 443$ and the pixel spacing is $2.6 \times 2.6 \times 2.6$ mm.

Regression approach: 100 runs of cross-validation experiments have been conducted where each experiment consists of a training phase on 20 patients chosen randomly and a test phase on the 13 remaining patients. For both forests and ferns, all parameters (number of trees/ferns and tree depth/number of nodes) have been tuned by performing grid-search within the same range for both techniques. Note that node optimization has been performed for random forests based on information gain. For prediction, each fourth

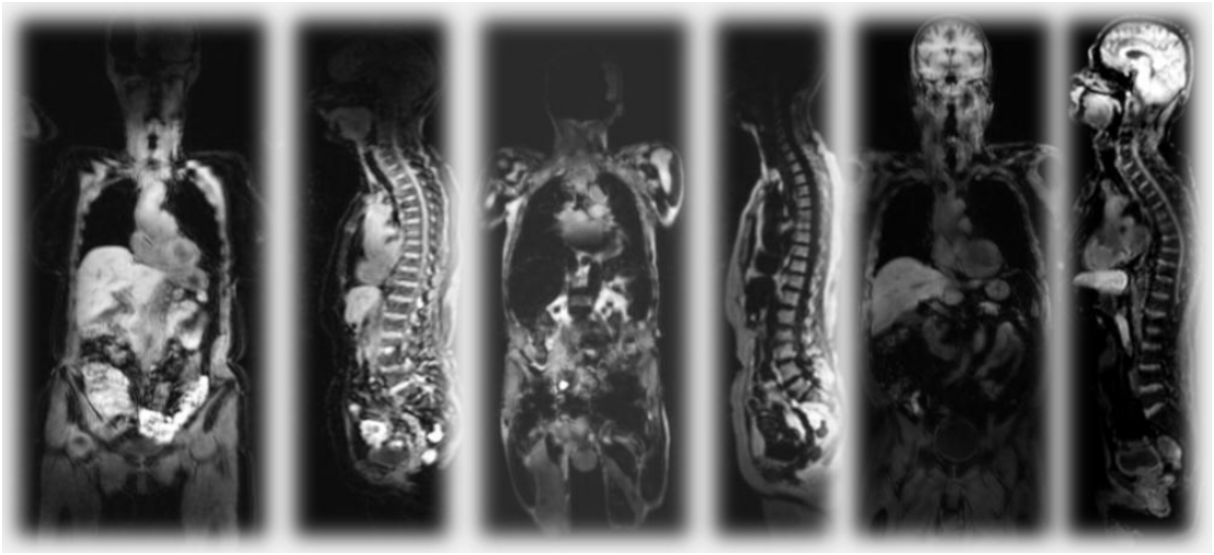

Figure 4.6: Real patient data: MR Dixon sequences from 33 cancer patients have been used for our cross-validation experiments.

MEAN LOCALIZATION ERRORS (mm)					
Organs	**Random Ferns**	**Random Forests**	**Atlas min**	**Atlas max**	**Atlas mean**
Head	9.82 ± 8.07	10.02 ± 8.15	18.00 ± 14.45	70.25 ± 34.23	35.10 ± 13.17
Left Lung	14.95 ± 11.35	14.78 ± 11.72	14.94 ± 11.54	60.78 ± 29.47	30.41 ± 11.39
Right Lung	16.12 ± 11.73	16.20 ± 12.14	15.02 ± 13.69	63.95 ± 30.13	29.85 ± 12.62
Liver	18.69 ± 13.77	18.99 ± 13.88	18.13 ± 16.26	70.59 ± 32.88	31.74 ± 13.49
Heart	15.17 ± 11.70	15.28 ± 11.89	13.31 ± 11.03	60.38 ± 28.90	29.82 ± 12.23
Overall	$\mathbf{14.95 \pm 11.33}$	$\mathbf{15.06 \pm 11.55}$	$\mathbf{15.88 \pm 13.40}$	$\mathbf{65.19 \pm 31.12}$	$\mathbf{31.38 \pm 12.58}$

Table 4.1: Organ localization results: Compared to atlas-based method, our approaches based on random ferns and forests achieve better accuracy and lower uncertainty.

pixel is used and described using 3D LBPs computed over 26 cuboidal regions chosen at 3 different scales.

Multi-atlas registration: 100 runs of cross-validation experiments have been performed. Each experiment is defined as follows: a set of 20 patients are chosen randomly as multi-atlas database and 1 patient is randomly chosen as test case. All 20 patients from the database are registered to the test patient using affine registration. Then, using the ground truth position of the bounding boxes of the test patient (which is not available in reality), we evaluate the theoretical lower and upper bounds of the error by using the patients in the database who provide the lowest and highest localization error. The mean error is computed over the whole database.

Results: Results reported on Tab.4.1 shows that we achieve an accuracy which is better than the "best case" atlas accuracy, while providing an increased robustness. Taking

a look at the localization error per organ, one can notice that the lowest error for our approach is achieved for the localization of the head, which is due to the fact that the head is surrounded by a lot of air which makes it easier to localize. While the heart shows second lowest error, lungs and liver were more difficult to localize. This is mainly due to the high inter-patient variability of the shape of these organs and to breathing-related deformations. The best results were obtained with 14 ferns/6 nodes for random ferns, and 6 trees/depth of 8 for regression forests. On a laptop with MATLAB 64 Core Duo 2.4 GHz, the training/testing time on 20/13 patients is 0.7/0.5 s for random ferns. Random Forests need 25/1 s. Concerning atlas registration, each single affine registration needs 12.5 s. Now if we analyze the results obtained by random ferns and regression forests, both approaches reach comparable localization accuracy. At first glance, one could expect forests to provide better localization performance as they benefit of node optimization in contrast to ferns. This can be explained by the limited size of the feature pool. Indeed, the 3D LBP like features compose a compact and relevant set of features to represent the visual context of voxels. For this reason, ferns achieve very good localization accuracy while being much faster to train and evaluate. To conclude, our approach provides a fast and robust solution for organ detection and localization and thus fulfills our requirements towards organ-specific attenuation map.

4.1.5 Conclusion

Our contribution is a supervised regression approach based on random ferns and random forests to detect and localize in one shot multiple organs in whole-body multi-channel MR images. Experiments conducted on a dataset of 33 patients show that our approach achieves an accuracy which is better than atlas-based methods, while providing higher robustness (lower uncertainty) and faster training/prediction times. Furthermore, this approach can be also useful to integrate semantic information *i.e.* incorporating organ labels in further applications such as registration, image navigation or image retrieval. In future work, the online performance of the proposed approach could be investigated to enable a fast updating of our organ localization system. Then we would like to move towards the construction of organ-specific attenuation correction maps.

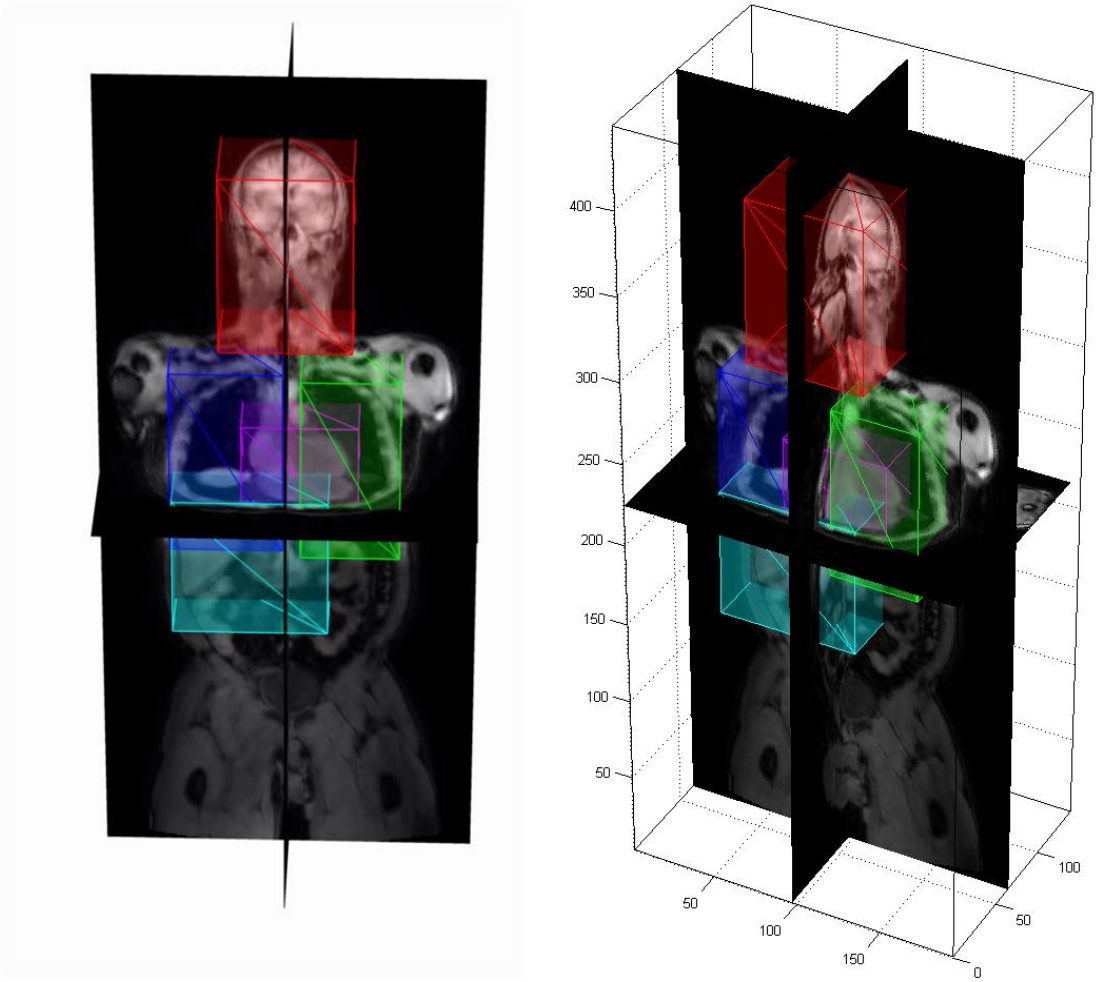

Figure 4.7: Organ localization results: 3D visualization of the localization outputs

4.2 Multiple Organ Segmentation in CT scans

In this section, we report our latest work on the segmentation of multiple organs in CT scans, which has been published in [37]. We introduce a new type of random forests that is built on a joint classification and regression formulation: the forest model aims at solving jointly (1) a classification task in which each voxel is associated to an organ class label and (2), a regression task in which each voxel is mapped to its distances to all organ boundaries. This enables the selection of more discriminative features leading to leaf clusters that are consistent in terms of **class** and **spatial location**. This implicitely integrates spatial regularization directly within our forest model. Experiments performed on real CT datasets demonstrate the benefits of our joint formulation for multiple organ segmentation.

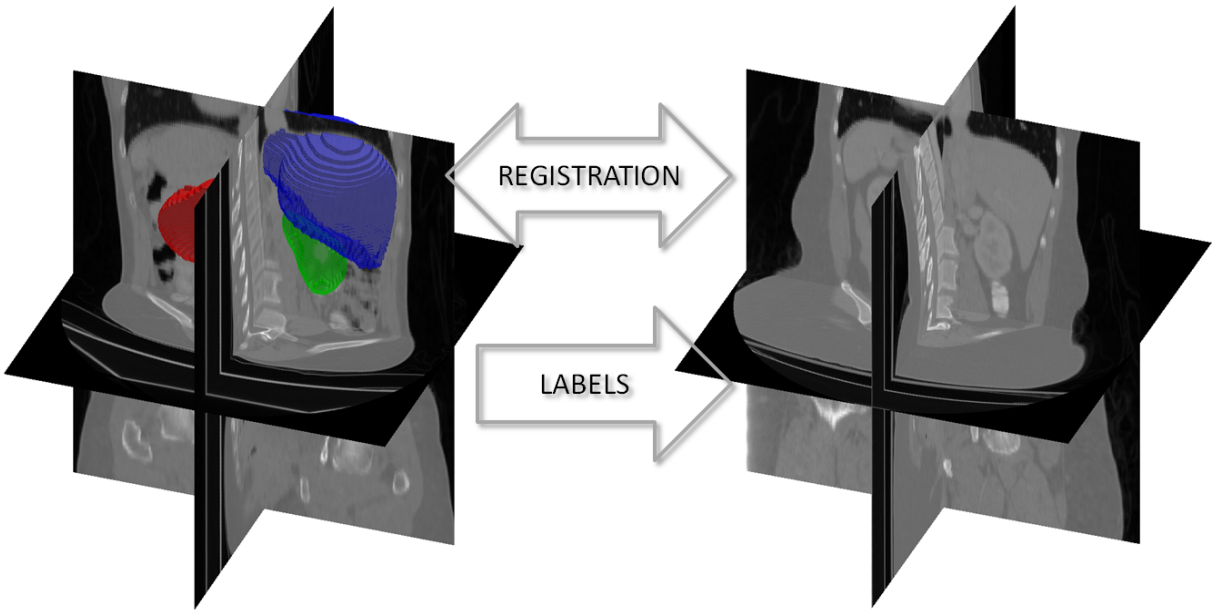

Figure 4.8: Atlas registration: State-of-the-art for multiple organ segmentation

4.2.1 Introduction

Organ segmentation can be defined as the task of assigning each voxel of a CT scan to an organ label. By registering the scan to segment with an annotated "atlas" scan, the labels of all voxels can be easily inferred by transferring the "atlas" labels to the new patient data as illustrated by fig.4.8. This approach, known as atlas-based registration, is considered as state-of-the-art for multiple organ segmentation. However, for large field-of-view scans, this task becomes very difficult due to the high inter-patient variability. Indeed, while affine registration lacks of flexibility, deformable registration may be difficult to regularize and thus can yield very large deformations which are not realistic considering the nature of the tissues.

Besides marginal space learning strategies [109] that show impressive results for the localization and segmentation of single anatomical structures, we demonstrated in the previous section that random forests and ferns can been successfully applied for the task of organs localization in multi-channel MR [73] as well as in CT [21]. Using regression forests and related techniques, we proposed to learn a statistical mapping relating each voxel to all organs of interest. Thereby we could: (1) discover key anatomical landmarks which provide best predictions, and (2) benefit of prior knowledge on the relative positions between all organs. Since in the case of anatomy localization, the goal is to predict the position and the size of each organ of interest, it can be easily formulated as a regression approach. In the case of segmentation, each voxel of a scan needs to be associated to an organ class. Hence the most natural way of tackling this problem is to formulate it as a classification task. However, classifying voxels based on their local visual context is very difficult in medical images, and yields predictions that lack of spatial consistency. There are two major advantages that are totally ignored when formulating the problem as a classification task: medical imaging follows often standard acquisition procedures, and the human anatomy offers a strong prior information on the global context such as the arrangement of organs, their size, shape, etc. Indeed, rich information beyond voxels labels is contained in annotated data and in the present section we propose to exploit this information to improve the consistence of predictions. Therefore, we introduce a novel random forest framework based on a joint classification-regression formulation. Each voxel is associated to an organ class label and to a vector containing its distance to all organ boundaries. By defining a joint classification-regression objective function, we encourage the selection of features leading to leaf clusters that are consistent in terms of **class** and **spatial location**. Thereby, spatial regularization is implicitely learned from the data and integrated directly within our forest model. In several experiments on synthetic and real data, we demonstrate the benefits of our approach which yields prediction with increased spatial consistency.

4.2.2 Problem statement

Let us consider a set of K organ classes represented by the labels $\mathcal{O} = \{\mathbf{O}_k\}_{k=1}^{K}$. In the general case, the goal of multiple organ segmentation is to assign an organ label $\mathbf{O}$ to each voxel $\mathbf{x} \in \mathbb{R}^3$ of a CT volume defined by an intensity function $\mathbf{I} : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$. In a probabilistic fashion, we can formulate this task as a maximum a posteriori problem:

$$\hat{\mathbf{O}} = \mathbf{argmax}_{\mathbf{O} \in \mathcal{O}} \, P(\mathbf{O}|\mathbf{x}, \mathbf{I}) \tag{4.7}$$

Given a set of observations and their associated labels, we need to learn the posterior distribution $P(\mathbf{O}|\mathbf{x}, \mathbf{I})$. As we saw in the previous sections, such probability distributions can be efficiently approximated by using random forests. Usually, classification forests aim at reducing the class uncertainty by maximizing at each node the information gain based on Shannon's entropy. While this objective may be solved during the training phase, forest predictions very often lacks of spatial consistency. In the present work, we propose to integrate additional spatial information within the same forest model by using a joint classification-regression formulation.

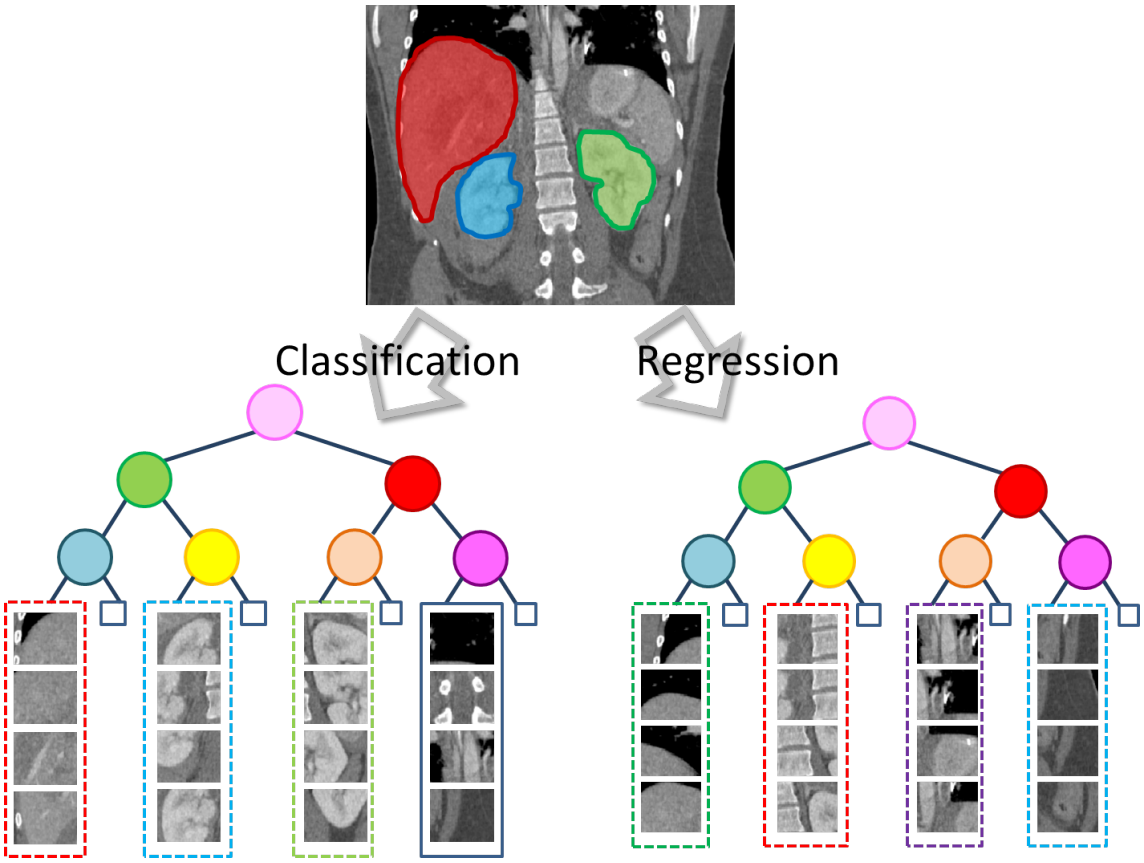

Figure 4.9: Classification vs. Regresion Forests: while classification forests (on the left) build leaf clusters that are consistent regarding the classes, regression forests (on the right) build leaf clusters that are consistent in terms of spatial location.

4.2.3 Joint Classification-Regression Forests

Classification and regression forests both aim at building leaf clusters that are homogenous according to the input feature space and to the output space. In the present work, we propose is to define a joint classification-regression objective to build clusters that have better characteristics for our task of multiple organ segmentation: (1) each voxel is associated to an organ class and (2) to its distances to all organ boundaries. This provides implicitely spatial regularization to our forest model, since leaf clusters will be consisting of training points that are: (i) homogenous in the feature space, (ii) belonging to the same class and (iii) have similar distances to the different organ boundaries as illustrated by fig. 4.9. Moreover, it benefits of implicit shape context information embedded in the regressed distances to the organ boundaries.

4.2.3.1 Joint Classification-Regression formulation

In the context of classification, each voxel $\mathbf{x} \in \mathbb{R}^3$ of a CT volume is associated to an organ label $\mathbf{O}$. Now, let us define by $\mathbf{B}_k$ the set of voxels belonging to the boundary of

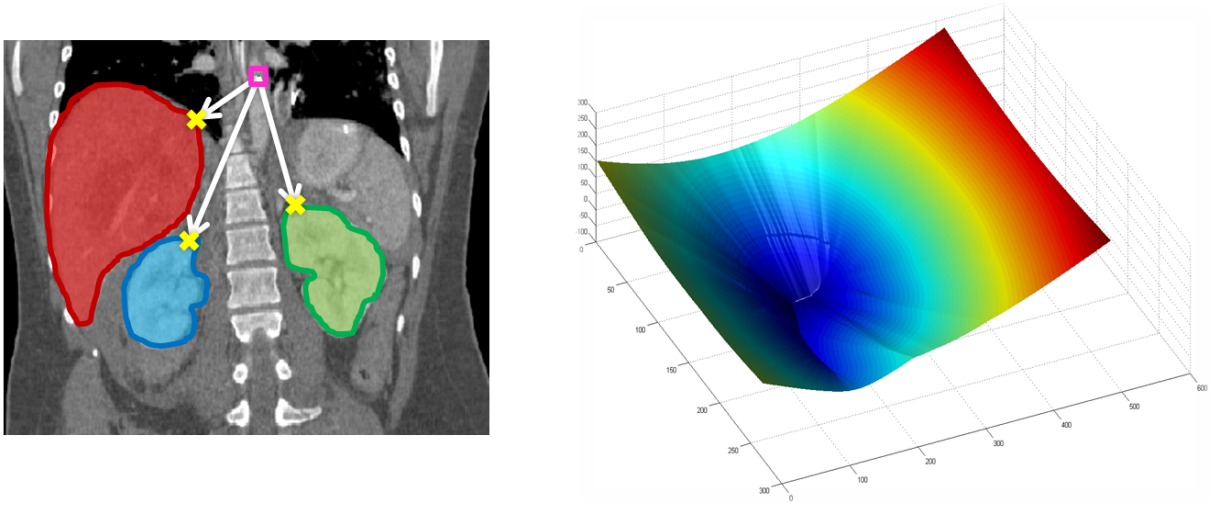

Figure 4.10: Regression objective: each voxel is associated to its distances to all organ boundaries. Thereby, we incorportate implicit organ shape information by using euclidean signed distances map.

organ $\mathbf{O}_k$ and its associated signed distance function D_k:

$$\begin{cases} D_k(\mathbf{x}) = \min_{\mathbf{x}' \in \mathbf{B}_k} ||\mathbf{x} - \mathbf{x}'||, \text{ if } \mathbf{x} \notin \mathbf{O}_k \\ D_k(\mathbf{x}) = -\min_{\mathbf{x}' \in \mathbf{B}_k} ||\mathbf{x} - \mathbf{x}'||, \text{ if } \mathbf{x} \in \mathbf{O}_k \end{cases} \tag{4.8}$$

As illustrated by Fig.4.10, we propose to associate each voxel $\mathbf{x}$ to a vector $\mathbf{D} = [D_1(\mathbf{x}), \cdots, D_k(\mathbf{x}), \cdots, D_K(\mathbf{x})]$, where $\mathbf{D} \in \mathcal{D} \subset \mathbb{R}^K$ contains the signed distance to all organ boundaries. Note that all these distances are computed in mm. Hence, we can formulate our joint classification-regression objective as the learning of the joint posterior $P(\mathbf{O}, \mathbf{D}|\mathbf{x}, \mathbf{I})$, which can be rewritten as:

$$P(\mathbf{O}, \mathbf{D}|\mathbf{x}, \mathbf{I}) = P(\mathbf{D}|\mathbf{O}, \mathbf{x}, \mathbf{I}) P(\mathbf{O}|\mathbf{x}, \mathbf{I}) \tag{4.9}$$

In the next section, we describe how to model both distributions $P(\mathbf{D}|\mathbf{O}, \mathbf{x}, \mathbf{I})$ and $P(\mathbf{O}|\mathbf{x}, \mathbf{I})$.

4.2.3.2 Classification-Regression Posteriors

Now let us define the posterior models we will use for our joint classification-regression task. Using a database of 3D CT scans, following training set can be constructed:

$$\{(\mathbf{x}^{(n)}, \mathbf{I}^{(n)}, \mathbf{O}^{(n)}, \mathbf{D}^{(n)})\}_{n=1}^{N}$$

where N = NbScans $\times$ NbVoxels, NbScans being the number of CT volumes in the database, and NbVoxels the number of voxels extracted (randomly) in each scan for training. Note that for the moment, as we have not defined the feature space yet, each voxel $\mathbf{x}$ is associated to the full CT data $\mathbf{I}$. To model the dependence of organ class and distances to organ boundaries, we need to model the class posterior as well as the conditional regression posterior. If we denote by $\mathcal{S}$ the subset of the training instances

that reach a given node, the class posterior can be simply modeled using a multinomial distribution as:

$$P(\mathbf{O} = \mathbf{O}_k|\mathbf{x}, \mathbf{I}) = |\mathcal{S}_k|/|\mathcal{S}| \tag{4.10}$$

where $\mathcal{S}_k$ represents the instances of $\mathcal{S}$ that belongs to class $\mathbf{O}_k$. We propose to model the conditional regression posterior for each class as:

$$P(\mathbf{D}|\mathbf{O} = \mathbf{O}_k, \mathbf{x}, \mathbf{I}) = \mathcal{N}_k^{(\mathcal{S}_k)}(\mathbf{D}|\mu_k^{(\mathcal{S}_k)}, \Sigma_k^{(\mathcal{S}_k)}) \tag{4.11}$$

where $\mathcal{N}_k^{(\mathcal{S}_k)}$ is a multivariate Gaussian with mean $\mu_k^{(\mathcal{S}_k)}$ and covariance matrix $\Sigma_k^{(\mathcal{S}_k)}$ estimated from the subset $\mathcal{S}_k$ that belong to class $\mathbf{O}_k$.

4.2.3.3 Robust statistics

During the learning phase, the training instances are iteratively split and the number of examples from a given class reaching the nodes/leaves is successively reduced. As the estimation of the conditional regression posterior is based on empirical means and covariances, it can become statistically problematic for small sample sizes. To overcome this problem, and to increase robustness against outliers, we replace the classical maximum likelihood estimation by a weighted Gaussian update, where the node's parent distribution plays the role of a prior. The mean can be then computed as follows:

$$\mu_k^{\text{child}} = \frac{\kappa}{\kappa + |\mathcal{S}_k^{\text{child}}|}\mu_k^{\text{parent}} + \frac{|\mathcal{S}_k^{\text{child}}|}{\kappa + |\mathcal{S}_k^{\text{child}}|}\bar{\mathbf{D}}_k^{\text{child}} \tag{4.12}$$

and the covariance matrix:

$$\begin{aligned}\Sigma_k^{\text{child}} =& \frac{\nu + n - 1}{\nu + n - 1 + |\mathcal{S}_k^{\text{child}}|}\Sigma_k^{\text{parent}} + \frac{|\mathcal{S}_k^{\text{child}}|}{\nu + n - 1 + |\mathcal{S}_k^{\text{child}}|}\Gamma_k^{\text{child}} \\ &+ \frac{\kappa|\mathcal{S}_k^{\text{child}}|}{(\kappa + |\mathcal{S}_k^{\text{child}}|)(\nu + n - 1 + |\mathcal{S}_k^{\text{child}}|)}\Lambda_k^{\text{child}}\end{aligned} \tag{4.13}$$

$\bar{\mathbf{D}}_k^{\text{child}}$ and Γ_k^{child} are respectively the empirical mean and covariance computed from the set of observations $\mathcal{S}_k^{\text{child}}$ from a given class reaching the child node. μ_k^{parent} and Σ_k^{parent} are the mean and covariance from the parent node, and Λ_k^{child} the covariance between the empirical mean and the prior (parent) mean $\Lambda_k^{\text{child}} = (\mu_k^{\text{parent}} - \bar{\mathbf{D}}_k^{\text{child}})(\mu_k^{\text{parent}} - \bar{\mathbf{D}}_k^{\text{child}})^\top$. The parameters κ and ν permit to control the trade-off between the prior and the empirical information. In fact, when the amount of training samples $|\mathcal{S}_k^{\text{child}}|$ is large enough, the empirical mean and covariance computed in the child node have more influence. In the case when the amount of training samples falls below a certain threshold defined by κ and ν, then the mean and covariance update relies more on the parent prior. In the following, we will discuss the node optimization which goal is to jointly reduce the uncertainty linked to the class and regression posteriors.

4.2.3.4 Node optimization

To characterize long range intensity context, we propose to use an infinite dimensional feature space where each feature basically compares the mean intensity in two different

regions Ω_1 and Ω_2. The corresponding splitting function $f_{\Omega_1,\Omega_2,\tau}$ is then defined by the position and the size of both regions and a threshold:

$$f_{\Omega_1,\Omega_2,\tau}(\mathbf{x},\mathbf{I}) = \left(\frac{1}{|\Omega_1|} \sum_{\mathbf{x}' \in \Omega_1} \mathbf{I}(\mathbf{x}') - \frac{1}{|\Omega_2|} \sum_{\mathbf{x}' \in \Omega_2} \mathbf{I}(\mathbf{x}') < \tau \right) \tag{4.14}$$

Having defined the type of splitting function, now we need an adapted objective function for our greedy optimization strategy. Given a subset of training points we denote $\mathcal{S}$, we define a joint entropy measure:

$$H(\mathcal{S}) = -\sum_{\mathbf{O} \in \mathcal{O}} \int_{\mathbf{D} \in \mathcal{D}} P(\mathbf{O},\mathbf{D}|\mathbf{x},\mathbf{I}) \log\left(P(\mathbf{O},\mathbf{D}|\mathbf{x},\mathbf{I})\right) d\mathbf{D} \tag{4.15}$$

Using the chain rule 4.9, this can be rewritten as:

$$\begin{aligned} H(\mathcal{S}) = & \underbrace{-\sum_{\mathbf{O} \in \mathcal{O}} P(\mathbf{O}|\mathbf{x},\mathbf{I}) \log\left(P(\mathbf{O}|\mathbf{x},\mathbf{I})\right)}_{H^{\mathrm{c}} = \text{Shannon's entropy}} \\ & + \underbrace{\sum_{\mathbf{O} \in \mathcal{O}} P(\mathbf{O}|\mathbf{x},\mathbf{I}) \left(- \int_{\mathbf{D} \in \mathcal{D}} P(\mathbf{D}|\mathbf{O},\mathbf{x},\mathbf{I}) \log\left(P(\mathbf{D}|\mathbf{O},\mathbf{x},\mathbf{I})\right) d\mathbf{D} \right)}_{H^{\mathrm{r}} = \text{weighted differential entropy}} \end{aligned} \tag{4.16}$$

where H^{c} is driving the classification objective, and H^{r} as conditional regression term, can be seen as a regularization: $H(\mathcal{S}) = H^{\mathrm{c}}(\mathcal{S}) + H^{\mathrm{r}}(\mathcal{S})$. As the conditional regression posterior is modeled using a multivariate Gaussian distribution, H^{r} can be rewritten as:

$$H^{\mathrm{r}}(\mathcal{S}) = \sum_{\mathbf{O}_k \in \mathcal{O}} P(\mathbf{O}_k|\mathbf{x},\mathbf{I}) \left(\frac{1}{2} \log\left((2\pi e)^K |\Sigma_k^{(\mathcal{S}_k)}| \right) \right) \tag{4.17}$$

where $\Sigma_k^{(\mathcal{S}_k)}$ is estimated from the points of $\mathcal{S}_k$ belonging to organ class $\mathbf{O}_k$. As both H^{c} and H^{r} may live in quite different ranges of values, we propose to normalize these entropies with respect to the root node:

$$H(\mathcal{S}) = \frac{H^{\mathrm{c}}(\mathcal{S})}{H^{\mathrm{c}}(\mathcal{S}_0)} + \frac{H^{\mathrm{r}}(\mathcal{S})}{H^{\mathrm{r}}(\mathcal{S}_0)} \tag{4.18}$$

where $\mathcal{S}_0$ represent the full training set at the root node. Finally, based on this entropy formulation, we compute the information gain Δ to measure the quality of a split. During node optimization, several decision function candidates are generated and the best is then chosen by maximizing Δ. This encourages the choice of features that permits in the end to build homogenous leaf clusters in terms of class and in terms of location within the anatomy.

4.2.3.5 Multiple organ segmentation

Now that we are able to estimate the joint probability $P(\mathbf{O},\mathbf{D}|\mathbf{x},\mathbf{I})$, we want to associate a class label to each unseen voxel $\mathbf{x}$. As inferring the distance from one voxel to the

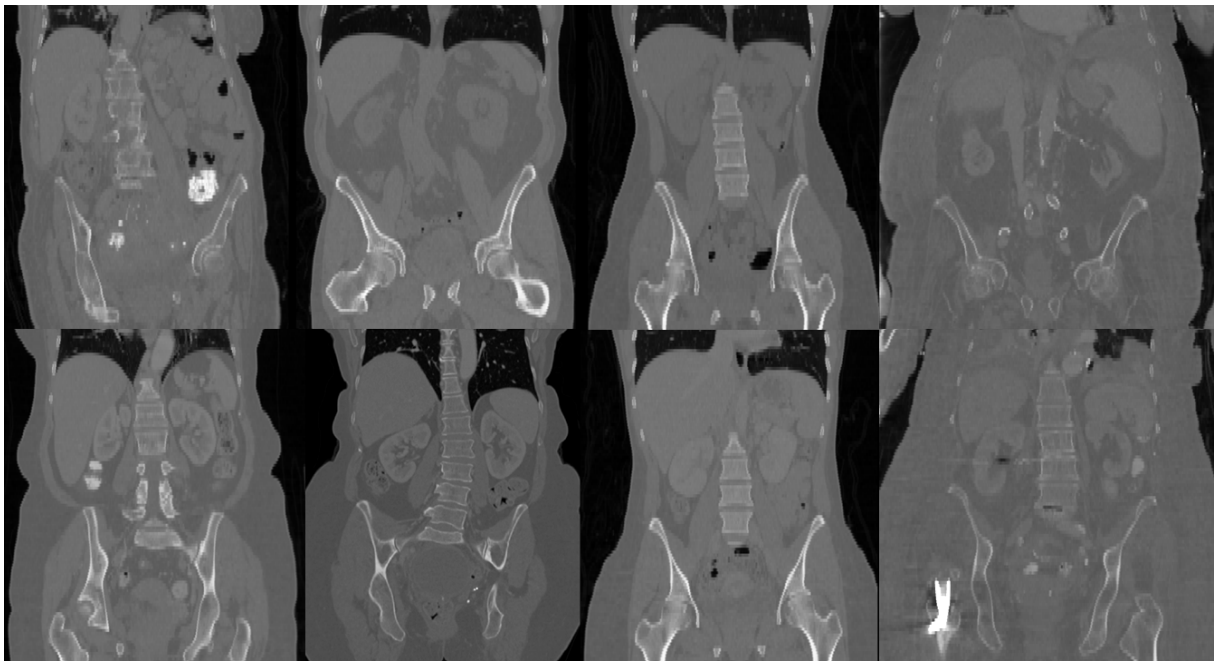

Figure 4.11: Database of 3D CT scans from 80 patients: high inter-patient variability, noise and artifacts such as contrast agents of metal implants make this database challenging for our segmentation experiments.

boundary of each organ based on the visual context is very challenging, we can expect the regression output to be very noisy. More generally, during the training of a tree, the classification objective for organ segmentation may be reached ealier in tree levels than the regression objective. Hence, we propose to consider only the **marginal** class posteriors for segmentation, as a more robust strategy since the regression posteriors remain uncertain:

$$\hat{\mathbf{O}} = \mathbf{argmax}_{\mathbf{O}\in\mathcal{O}}\, P(\mathbf{O}|\mathbf{x},\mathbf{I}) \tag{4.19}$$

Thereby, the regression term has an influence during the training phase for features and test selection, but not during the test phase. In the following section, we will investigate the performances of our joint classification-regression forests for the segmentation of multiple organs in real CT data.

4.2.4 Experiments and Results

Through exhaustive experiments on a 3D CT database of 80 patients, we propose to demonstrate the benefits of our approach over a classical multi-class classification method. As shown on fig.4.11, this database is really challenging due to the high inter-patient variability and also noise and artifacts such as metal implants. In all scans, 6 organs of interest have been manually segmented: liver, spleen, left and right kidneys, left and right pelvic bones. Note that this manual segmentation will be considered as gold standard in our evaluation.

4.2.4.1 Measuring the segmentation accuracy

To measure quantitatively the segmentation accuracy, we compute the DICE coefficient, the mean surface distance (MSD), the root mean square surface distance (RMS-SD) and the Hausdorff distance (HD). The DICE coefficient is defined as an overlap ratio between the gold standard segmentation $\mathbf{V}^{\text{gold}}$ and the forest segmentation output $\mathbf{V}^{\text{forest}}$:

$$\text{DICE}(\mathbf{V}^{\text{gold}}, \mathbf{V}^{\text{forest}}) = \frac{2.|\mathbf{V}^{\text{gold}} \cap \mathbf{V}^{\text{forest}}|}{|\mathbf{V}^{\text{gold}}| + |\mathbf{V}^{\text{forest}}|} \tag{4.20}$$

where $\mathbf{V}^{\text{gold}}$ represents the set of voxels belonging to the gold standard segmentation and $\mathbf{V}^{\text{forest}}$ to the forest segmentation output. Hence, the DICE coefficient tends towards 1 when the segmentation output has a large overlap with the gold standard. For the three other measures, we need to consider the 3D segmentation boundaries of the gold standard $\mathbf{S}^{\text{gold}}$ and of the forest output $\mathbf{S}^{\text{forest}}$. Thus, the mean surface distance is evaluated as:

$$\text{MSD}(\mathbf{S}^{\text{gold}}, \mathbf{S}^{\text{forest}}) = \frac{1}{|\mathbf{S}^{\text{forest}}|} \sum_{\mathbf{x} \in \mathbf{S}^{\text{forest}}} \min_{\mathbf{x}' \in \mathbf{S}^{\text{gold}}} ||\mathbf{x} - \mathbf{x}'|| \tag{4.21}$$

the root mean squared surface distance as:

$$\text{RMS-SD}(\mathbf{S}^{\text{gold}}, \mathbf{S}^{\text{forest}}) = \sqrt{\frac{1}{|\mathbf{S}^{\text{forest}}|} \sum_{\mathbf{x} \in \mathbf{S}^{\text{forest}}} \min_{\mathbf{x}' \in \mathbf{S}^{\text{gold}}} ||\mathbf{x} - \mathbf{x}'||^2} \tag{4.22}$$

and the Hausdorff distance as:

$$\text{HD}(\mathbf{S}^{\text{gold}}, \mathbf{S}^{\text{forest}}) = \max \left(\max_{\mathbf{x} \in \mathbf{S}^{\text{forest}}} (\min_{\mathbf{x}' \in \mathbf{S}^{\text{gold}}} ||\mathbf{x} - \mathbf{x}'||), \max_{\mathbf{x} \in \mathbf{S}^{\text{gold}}} (\min_{\mathbf{x}' \in \mathbf{S}^{\text{forest}}} ||\mathbf{x} - \mathbf{x}'||) \right) \tag{4.23}$$

Note that the three latest measures are all in mm, and tend towards zero if the predicted segmentation is ideally good.

4.2.4.2 Cross-validation experiments

To demonstrate the benefit of our joint classification-regression formulation, we propose to compare our approach to a classical classification strategy using random forest. To this end, we perform a two-folds cross-validation, *i.e.* the database is split in two subsets of 40 patient scans which are successively used as training and test set. For both approaches, we investigate the same range of parameters, using forest counting 40 trees and varying the tree depth until a maximum of 20. To construct the training set, we use bagging to select from each training scan a random subset of 5% of all voxels.

During the greedy optimization, at each node a set of 100 features is randomly generated, and 10 uniformly distributed thresholds are evaluated. The best split candidate is then chosen by maximizing the information gain. For the posterior computation, we use the Gaussian update for the mean and covariance estimation presented in the previous section, and we choose $\kappa = 10$ and $\nu = 10$.

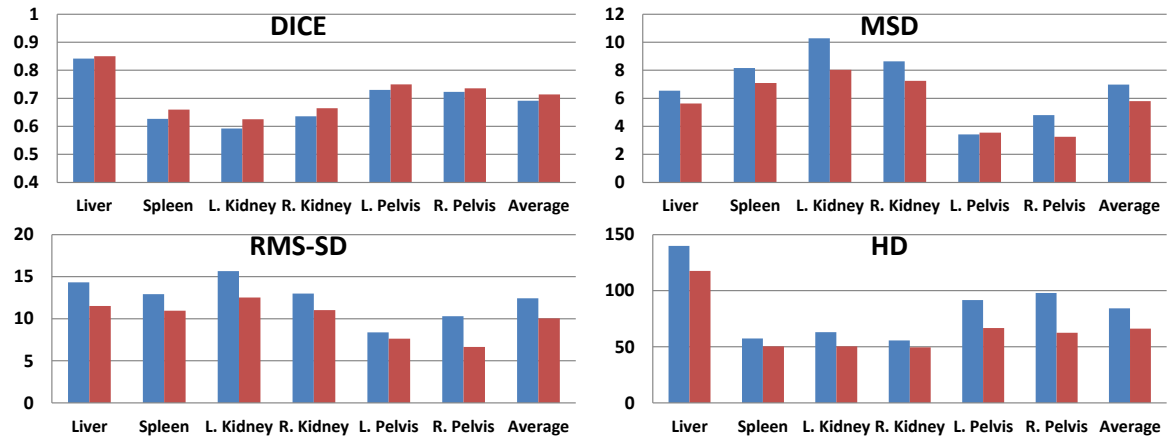

Figure 4.12: Overall segmentation results: Four different quality measures are shown in this figure: DICE measures the overlap agreement between the forest output and gold standard where 1 indicates perfect results. MSD, RMS-SD, and HD are different measures of surface distances in millimeters between prediction and gold standard where 0 indicates perfect results. Results for classification forests are the blue bars on the left, and for our approach the red bars on the right. All four measures confirm the benefits of our approach that yields better segmentation results.

4.2.4.3 Results

The quantitative results for individual organs and the average performance are summarized in fig.4.12. We report comparative results with respect to the gold standard annotations over the four different segmentation measures previously described, *i.e.* the DICE, MSD, RMS-SD and HD. All confirm the benefits of a joint classification-regression formulation by showing improvement for the segmentation of all organs of interest. In particular, improvements in RMS-SD and HD, that are more sensitive to large local errors, show that our approach permits to reduce the amount of outliers. While both classification and joint classification-regression forests perform partitioning in the same feature space, the joint objective function permits to select better feature tests that encourage the creation of clusters in the leaves that are consistent in terms of classes and spatial location. This yields prediction outputs that are spatially more consistent.

Further qualitative results showing the gold standard segmentation, the forest outputs for both approaches, as well as the probability maps can be found in fig.4.13 and 4.14. Again, one can see that our joint classification-regression approach permits to learn implicitely spatial regularization from the data: better segmentation results can be achieved while reducing the "tentacle-like" outliers at the bottom of the liver or kidneys, around the ribs or the vertebrae, and also to prevent the segmentation of the pancreas as being part of the kidney. By looking only at the visual context of pixels, all these outliers make sense, as for instance the left kidney and the pancreas are neighboring and have very similar intensities. Learning spatial regularization directly from the data encourage the disambiguation of such cases.

4.2.5 Conclusion

To conclude, in this section, we proposed an novel random forest approach to tackle the problem of multiple organ segmentation. When casting segmentation as a classification task, strong prior knowledge contained in the annotated scans such as organ positions, size or shape are not exploited. We proposed to take advantage of this rich information by formulating the segmentation problem as a joint classification-regression task: each voxel is associated to an organ class label and to a vector containing its distance to all organ boundaries. By defining a joint classification-regression objective function, our novel random forest model aims at creating leaf clusters of voxels being consistent regarding their class and spatial location. We could demonstrate the benefits of our approach in extensive experiments on real CT data. Indeed, results confirm the fact that better predictions can be obtained, reducing outliers due to ambiguous visual context. In this work, the regression output, which aims at predicting full organ distance maps, has not been used directly for segmentation. In the future, we need to investigate approaches to also take advantage of this regression output in order to further improve the segmentation results.

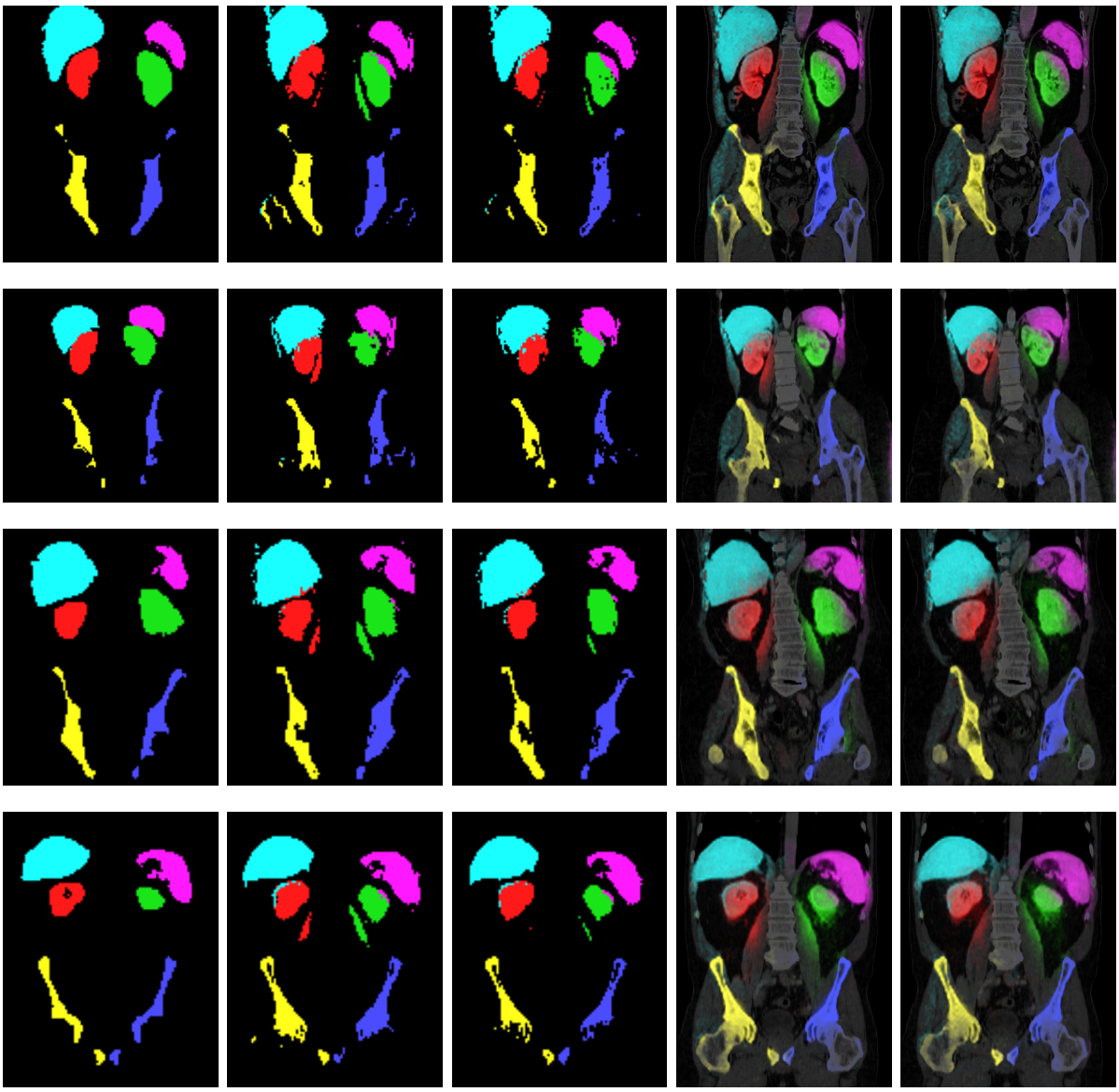

Figure 4.13: From left to right: Gold standard manual segmentation, MAP estimate of classification forest, MAP estimate of our joint approach, probability map of standard classification, probability map of our joint approach.

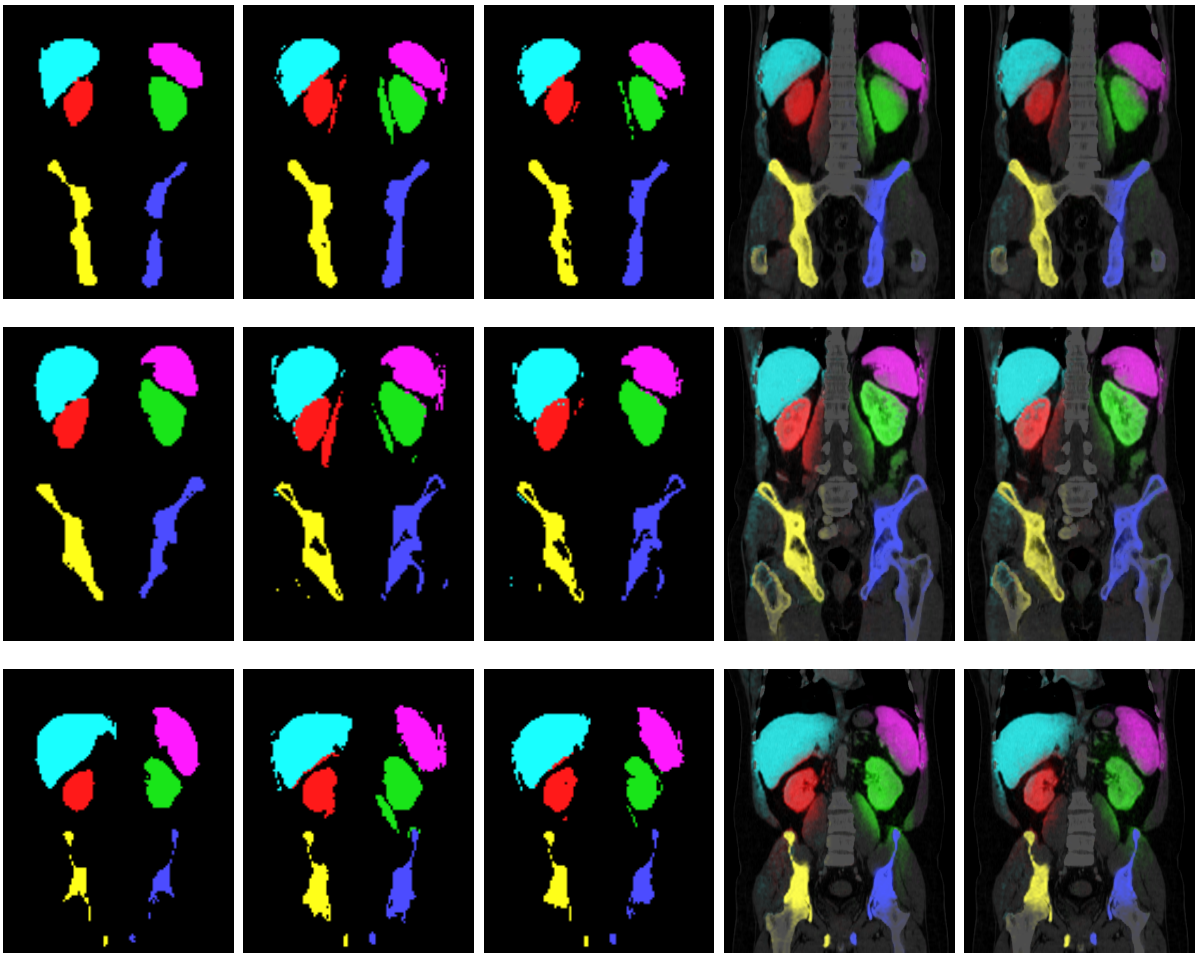

Figure 4.14: From left to right: Gold standard manual segmentation, MAP estimate of classification forest, MAP estimate of our joint approach, probability map of standard classification, probability map of our joint approach.

4.3 Detection of Substantia Nigra Echogenicities in 3D Transcranial Ultrasound towards Computer Aided Diagnosis of Parkinson Disease

Parkinson's disease (PD) is a neurodegenerative movement disorder caused by decay of dopaminergic cells in the substantia nigra (SN), which are basal ganglia residing within the midbrain area. In the past two decades, transcranial B-mode sonography (TCUS) has emerged as a viable tool in differential diagnosis of PD and recently has been shown to have promising potential as a screening technique for early detection of PD, even before onset of motor symptoms. In TCUS imaging, the degeneration of SN cells becomes visible as bright and hyper-echogenic speckle patches (SNE) in the midbrain. Recent research proposes the usage of 3D ultrasound imaging in order to make the application of the TCUS technique easier and more objective. In this section, we report our latest contribution in the development of learning-based tools to support the diagnosis of Parkinson disease, which has been published in [72]. For the first time, we propose an automatic 3D SNE detection approach based on random forests, with a novel formulation of SNE probability that relies on visual context and anatomical priors. On a 3D TCUS dataset of 11 PD patients and 11 healthy controls, we show that our SNE detection approach yields promising results that seem to correlate well with experts annotations.

4.3.1 Introduction and Medical Motivation

Parkinson's Disease (PD) is a neuro-degenerative movement disorder which has been the matter of increasing research in the medical and scientific community for the past decades. The primary symptoms of PD affect the motoric system, such as rigidity, shaking or slowness, but PD may also evoke non-motor symptoms such as dementia in later stages of the disease. The root cause of PD is the death of dopaminergic substantia nigra (SN) cells, which are located in the midbrain area. Although it is not known whether it is the cause or an effect of SN cell death, the progress of the disease is accompanied by a build-up of ferrite deposits within the SN. Over the past two decades, several studies have shown that these physiological changes can be visualized using transcranial ultrasound (TCUS), making this imaging technique a viable tool in differential diagnosis of PD [103]. Additionally, it has been shown recently that TCUS can be used as an early indicator of PD [8]. This result is particularly relevant, since it increases the hope that TCUS can be used as a cheap, quick and non-invasive early-detection and screening tool for large populations. The changes in SN are visible in TCUS in form of hyper-echogenicities, i.e. small bright speckle patches, within the midbrain. If performed by an expert sonographer with substantial experience in this technique, sensitivity and specificity of this technique can be as high as 90% [32]). However, the challenging nature of TCUS images causes high intra- and inter-rater variability and makes it difficult for less experienced groups to reach the diagnostic reliability of expert groups in this field [102]. Recently, the usage of three-dimensional (3D-) TCUS started being investigated for PD diagnosis, since it can make this promising PD screening technique easier, more objective, and more sig-

nificant due to the volumetric analysis of substantia nigra echogenicities (SNE). In this section, we introduce a fully automatic approach for the detection of SNE voxels within the midbrain, once the latter has been localized. There is little related work in literature concerning the automatic analysis of SNE, but similar to our work, all approaches we are aware of perform a midbrain ROI segmentation first and a SNE detection within the midbrain subsequently. Kier et al. [50] and Chen et al. [15] respectively perform SN pixel detection using morphological operators or image-feature-based SVM classification, both within a manually segmented midbrain in 2D. Engels et al. [28] use a hierarchical finite-element model and active contours to simultaneously segment the midbrain and SNEs in 2D. Despite early work on segmentation of midbrain area in 3D ultrasound [1], to our knowledge, there is no previous work on (semi-) automatic SNE analysis in 3DUS. The main contributions of our work are therefore to 1) propose a novel and *volumetric* SNE detection method based on random-forest, 2) formulate a detection paradigm mimicking human experts by using probabilistic modeling of visual and spatial SNE features and 3) demonstrate the reliability of our SNE detection approach on a database of 3D-TCUS volumes from 22 subjects.

4.3.2 Data acquisition and Midbrain Segmentation

4.3.2.1 Data acquisition:

For validation of our methods, we utilize a 3D-TCUS dataset acquired on 22 subjects, comprising 11 PD patients and 11 healthy controls. The acquisition was performed using 3D Freehand Ultrasound, i.e. by synchronized acquisition of 2D ultrasound images and 3D optical tracking data. Additionally, the acquisition was performed in a bi-lateral fashion, i.e. by reconstructing and combining US image information from both the left and the right bone window into a single bi-lateral 3D volume. The bi-lateral acquisition is an advantage in TCUS imaging and is not achievable in 2D. It partly allows for compensation of differing bone window qualities and leads to an information gain of TCUS image data in the midbrain area. In our study, only one subject had to be excluded from our evaluation due to an insufficient bone window, which is less than the typical 10% on whom this technique cannot be applied [103]. In total, the remaining dataset used in this study comprises data of 11 previously diagnosed PD patients and 11 healthy controls. The 3D volumes were reconstructed at an isotropic resolution of 0.45mm and labeled by a blinded expert into the regions "midbrain", "SNE left" and "SNE right". Manual segmentations of midbrain and SNEs are used as gold standard in this study, but one should note that even in the 2D method, intraclass correlation coefficients (ICC) of around ICC 0.85 are reported as the inter-rater variability [99] and variability in 3D is possibly higher than that. Hence, within our study, we assume that an SNE detection quality within the ranges of 2D variability can be argued as acceptable.

4.3.2.2 (Semi-)automatic midbrain segmentation:

In previous work [1], Ahmadi *et al.* proposed an easy-to-use, robust and accurate semi-automatic method for midbrain segmentation in B-Mode 3D-TCUS. The 3D-TCUS seg-

mentation method is based on a statistical shape model (SSM) of the midbrain, which was created using the above described dataset of 22 subjects. The segmentation method combines the SSM with a localized region-based cost function, an explicit active surface formulation and a gradient-descent optimization. The only required interaction for the user is to manually position the SSM mean shape with high overlap onto the midbrain region in the 3D-TCUS volume, which takes a few seconds only. In a five-fold cross-validation setup, the method achieved a high regional overlap with the manual expert segmentation (median DICE 0.85) and was able to retain a median of 95% of diagnostically relevant SNE voxels. This segmentation outcome will be used as a region of interest for our following detection approach.

4.3.3 Detection of Substantia Nigra echogenicities in 3D

As illustrated by fig.4.15, an experimented observer can detect PD-related hyperechogenicities in the left and right SN using 3D TC-US. Unfortunately, TCUS cannot visualize the SN regions themselves, but only the high-contrast SNE speckles located randomly within the area of SN. Thus, relying on prior knowledge of the midbrain anatomy and the known rough location of the SN within the midbrain, an experimented observer has to decide whether an echogenicity belongs to the SN or not based on location and intensity of speckle patches. This makes the detection of Parkinson-related SNEs quite challenging. In the present work, we aim at providing a reliable detection of PD-related SNEs in 3D by analoguously integrating two types of information: (i) visual context and (ii) spatial location within the midbrain.

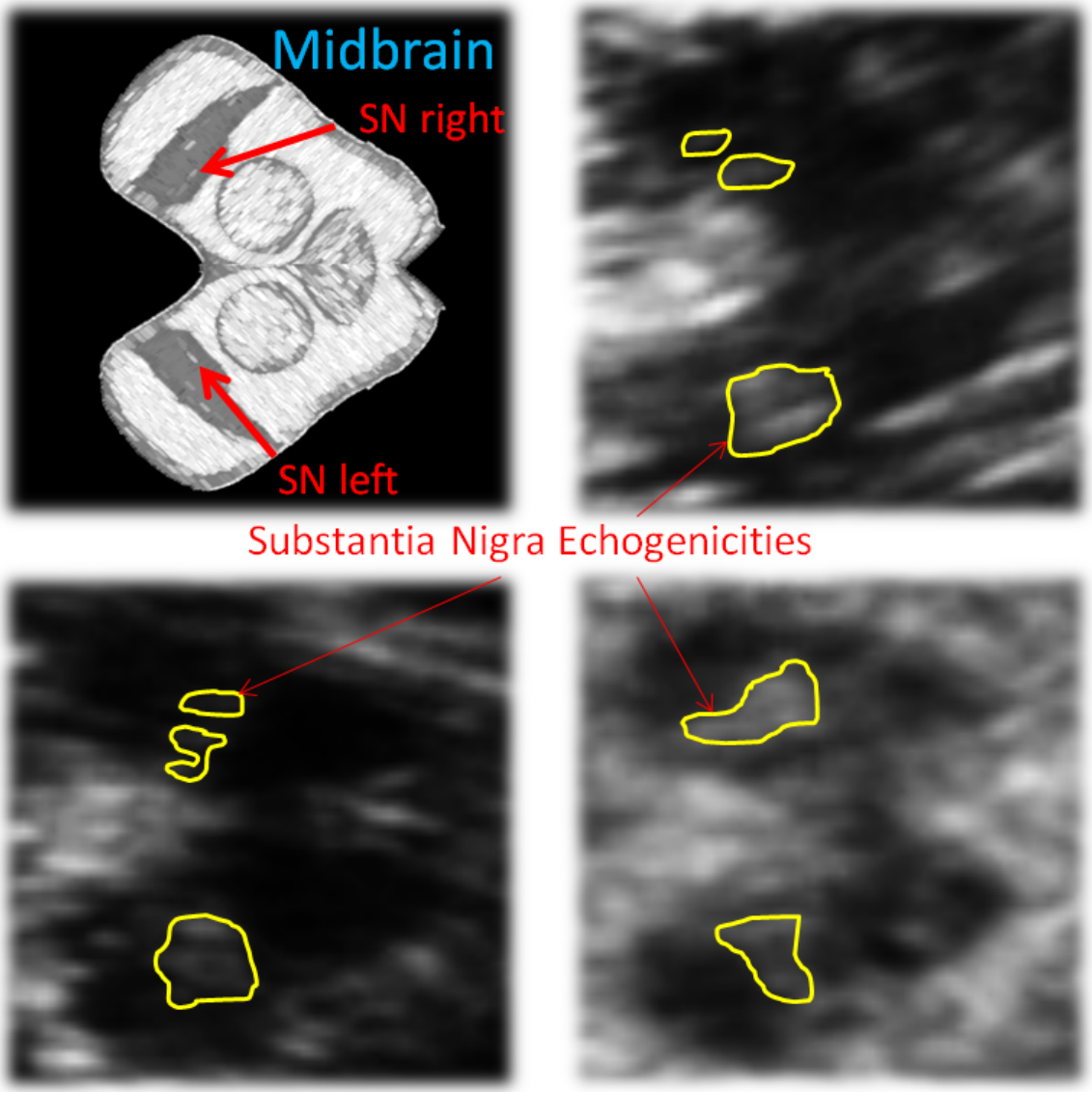

Figure 4.15: Goal of our approach: On the top left, the anatomy of the midbrain is detailed, showing the Substantia Nigra regions located at the front of both hemispheres. The other images show examples of typical SNE speckle patterns (in yellow) in 3D TCUS transversal slices.

4.3.3.1 Problem statement

Let us consider an intensity function denoted by $\mathbf{I} : \Omega \rightarrow \mathbb{R}$, where $\Omega \subset \mathbb{R}^3$ is the image domain representing the 3D ultrasound data. We assume that we are given a segmentation of the midbrain $\mathcal{M} \subset \Omega$, either from a manual expert segmentation or alternatively from the output of a ROI detection algorithm [1]. In this work, we propose to formulate the detection problem as a classification task in which each voxel $\mathbf{x} \in \mathcal{M}$ needs to be associated to a label $\mathbf{c} \in \{0, 1\}$, where 0 denotes the background and 1 the Substantia Nigra Echogenicities (SNE) class. In fact, $\mathbf{c}$ is the realization of 2 random variables $(\mathcal{E}, \mathcal{A})$ where $\mathcal{E}$ represents the observation of an echogenicity and $\mathcal{A}$ of the Substantia Nigra (SN), i.e. $\mathbf{c} = 1$ if and only if $\mathcal{E} = 1$ **and** $\mathcal{A} = 1$. Therefore, we aim at learning $P(\mathcal{E}, \mathcal{A}|\mathbf{x}, \mathbf{I})$, which represents the joint probability of observing an echogenicity $\mathcal{E}$ belonging to the SN $\mathcal{A}$ given the location $\mathbf{x}$ and the intensity function $\mathbf{I}$. It is important to note that (1) it is not the SN itself which causes hyper-echogenicities but only potential acoustic micro-scatterers residing within it and (2) echogenicities can happen in the whole skull in TCUS due to tissue boundaries and micro-scatterers present in the entire brain tissue. Hence, we can assume the independence of the random variables $\mathcal{E}$ and $\mathcal{A}$, and decompose this joint probability as follows:

$$P(\mathcal{E}, \mathcal{A}|\mathbf{x}, \mathbf{I}) = P(\mathcal{E}|\mathbf{x}, \mathbf{I})P(\mathcal{A}|\mathbf{x}) \tag{4.24}$$

The first term $P(\mathcal{E}|\mathbf{x}, \mathbf{I})$ is a data term, encoding the probability of observing an echogenicity given some visual information at location $\mathbf{x}$, and the second term $P(\mathcal{A}|\mathbf{x})$ is an anatomical prior not depending on $\mathbf{I}$, i.e. the ultrasound data. As learning these probability distributions is challenging due to the dimensionality of the problem, we propose to use two discriminative models based on random forests. Geremia et al. in [35, 34] demonstrated state-of-the-art results for the segmentation of multiple-sclerosis lesions based on multi-channel MRI data. In addition to a forest using visual context, we propose to learn a novel spatial prior based on two hemisphere-specific coordinate systems. In the following, we describe how to use random forests for learning: (1) the data term $P(\mathcal{E}|\mathbf{x}, \mathbf{I})$ and (2), the prior $P(\mathcal{A}|\mathbf{x})$.

4.3.3.2 Learning the data term $P(\mathcal{E}|\mathbf{x}, \mathbf{I})$

In TCUS, echogenicities are characterized by higher intensities and higher contrast. Therefore, we propose to describe the visual context of a voxel at location $\mathbf{x}$ by extracting a set of simple features that encode the mean intensities in cuboidal regions of different sizes in the neighborhood of $\mathbf{x}$ similarly as in [35, 34]. Let us denote by $\mathcal{X}$ the space spanned by these simple features, and $\mathbf{X}$ the feature representation associated to a voxel at location $\mathbf{x}$. We consider a training set $\left(\mathbf{X}^{(n)}, \mathcal{E}^{(n)}\right)_{n=1}^{N}$, where each feature vector $\mathbf{X}^{(n)}$ is associated to a label $\mathcal{E}^{(n)}$ which is equal to 1 if there is an echogenicity at location $\mathbf{x}^{(n)}$ and 0 if not. To efficiently partition this high-dimensional space $\mathcal{X}$, we propose to use a random forest. Each tree is a directed acyclic graph, and each node consists in a decision function $f_{\mathbf{v},\tau}$ defined as:

$$f_{\mathbf{v},\tau}(\mathbf{X}) = (\mathbf{X} \cdot \mathbf{v} \geq \tau) \tag{4.25}$$

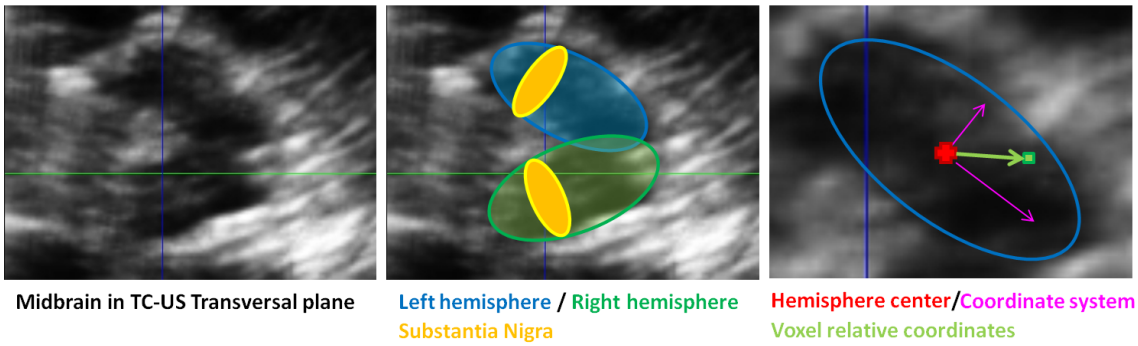

Figure 4.16: Midbrain anatomy: in the transversal plane, the midbrain has a characteristic butterfly shape. The Substantia Nigra are thin structures located at the front of both hemispheres. A hemisphere-specific coordinate system is computed to express voxel spatial location accounting for inter-patient asymmetric changes of scales and orientation.

$\mathbf{v}$ being a vector of dimensionality $\dim(\mathcal{X})$, and $\tau \in \mathbb{R}$ a threshold. Here, we use axis-aligned splits in $\mathcal{X}$, by generating $\mathbf{v}$ having only one non-zero entry. At each node, the choice of $\mathbf{v}$ and τ is optimized following a greedy optimization strategy. From a set of functions that are drawn randomly, the best candidate is selected by maximizing the information gain based on Shannon's class entropy. The posterior distribution can be estimated from the set of training instances $\mathcal{S}$ reaching the current node as:

$$P(\mathcal{E} = e|\mathbf{x}, \mathbf{I}) = \frac{|\left\{\mathbf{X}^{(n)} \in \mathcal{S},\ \mathcal{E}^{(n)} = e\right\}|}{|\left\{\mathbf{X}^{(n)} \in \mathcal{S}\right\}|} \tag{4.26}$$

By optimizing the information gain, the tree aims at minimizing the uncertainty on the random variable $\mathcal{E}$, encouraging thereby the creation of leaves containing either mostly echogenicities, or mostly background. Nodes are grown until a maximal tree depth has been reached, or when the number of feature points falls below a given threshold. Finally, in each leaf, the posterior distribution $P(\mathcal{E}|\mathbf{x}, \mathbf{I})$ is computed on the set of features points reaching this leaf using eq.4.26 and stored. Now, to predict the probability of observing an echogenicity at a location $\mathbf{x}$ for an unseen ultrasound volume of the midbrain, one just needs to first extract its associated feature vector $\mathbf{X}$, to push it downward the tree until it reaches a leaf, and to use the stored posterior distribution. Considering a random forest consisting of T trees, predictions can be simply computed by averaging tree posteriors: $P(\mathcal{E}|\mathbf{x}, \mathbf{I}) = \frac{1}{T}\sum_t^T P_t(\mathcal{E}|\mathbf{x}, \mathbf{I})$.

4.3.3.3 Learning the prior $P(\mathcal{A}|\mathbf{x})$

As shown on fig.4.16, the midbrain has a characteristic butterfly shape in the transversal plane, which does not vary much along the longitudinal axis. The Substantia Nigra are thin structures located at the front of both hemispheres and do not vary much along the longitudinal axis either. Hence, we propose to express the location of each voxel using patient-specific coordinate systems that represent the left and right midbrain hemispheres in the transversal plane. By doing so, we can easily account for asymmetric changes of scales and orientation of the midbrain anatomy, which can occur in TCUS imaging. Let

us denote by $\{\mathbf{x}^{(m)}\}_{m=1}^{M} = \mathcal{M}$, the finite set of M voxels belonging to the midbrain. First, the centers of the left and right hemispheres are computed by performing a K-means clustering on $\mathcal{M}$. Then, each voxel is associated to its nearest cluster center to create the 2 hemisphere subsets $\mathcal{H}^{\text{left}}$ and $\mathcal{H}^{\text{right}}$. Finally, principal component analysis is applied to each of these subsets to compute a hemisphere-specific transversal coordinate system, and the location of each point is expressed in the normalized coordinate systems of the hemisphere it belongs to. The in-plane location of each voxel $\mathbf{x}^{(m)}$ can then be encoded by a vector $\mathbf{x}'^{(m)} = \left[x'^{(m)},\ y'^{(m)},\ h^{(m)}\right]$, where $x'^{(m)}$ and $y'^{(m)}$ are the in-plane components in the hemisphere coordinate system, and $h^{(m)}$ is a categorical variable encoding the left/right side. To summarize, each voxel $\mathbf{x}^{(m)}$ is associated for the training phase to a couple $\left(\mathbf{x}'^{(m)},\ \mathcal{A}^{(m)}\right)$, where $\mathcal{A}^{(m)}$ is equal to 1 if $\mathbf{x}^{(m)}$ belongs to the Substantia Nigra and 0 if not. As in the previous section, we use a random forest to learn the prior $P(\mathcal{A}|\mathbf{x})$ using a training set of 3D TCUS from different patients. During the training, each tree aims at separating the SN from the rest of the midbrain, and creates clusters in its leaves that are consistent in terms of spatial location $\mathbf{x}'$.

4.3.3.4 SNE detection

Once the data term and the prior have been learned from a set of labelled midbrains, a new unseen patient data can be processed as follows:

1. the midbrain $\mathcal{M}$ is segmented,
2. the hemisphere coordinate systems are determined using K-means followed by a PCA on the voxels belonging to $\mathcal{M}$,
3. the probability $P(\mathcal{E}|\mathbf{x}, \mathbf{I})$ and the prior $P(\mathcal{A}|\mathbf{x})$ are computed for each $\mathbf{x} \in \mathcal{M}$,
4. the joint probability $P(\mathcal{E}, \mathcal{A}|\mathbf{x}, \mathbf{I})$ can be predicted using Eq.4.24.

Hence, we obtain for each voxel a probability of belonging to an SNE, and we can use a threshold $\mathcal{T} \in [0,\ 1]$ to create a binary segmentation of the ferrite deposits: $\mathbf{c} = 1$ if $P(\mathcal{E}, \mathcal{A}|\mathbf{x}, \mathbf{I}) \geq \mathcal{T}$, and $\mathbf{c} = 0$ otherwise.

4.3.4 Experiments and Results

In this section, we evaluate our SNE detection approach on the bi-lateral 3D TCUS dataset volume of 22 subjects, consisting of 11 PD patients and 11 healthy controls. The 3D volumes were reconstructed at an isotropic resolution of 0.45mm and labeled by a blinded expert into the regions "midbrain", "SNE left" and "SNE right". For our validation, we will consider this labeling as gold standard. We conduct comparative experiments to evaluate our SNE detection approach based on 2 discriminative models (VisForest-PriorForest) against the simple forest without spatial prior (VisForest), and a forest with a spatial prior constructed using a Gaussian model for each hemisphere (VisForest-GaussianPrior). We perform a leave-one-patient-out cross-validation, i.e. we train all models on 21 labeled midbrains and test on the remaining one. As the outputs from our system are probabilities

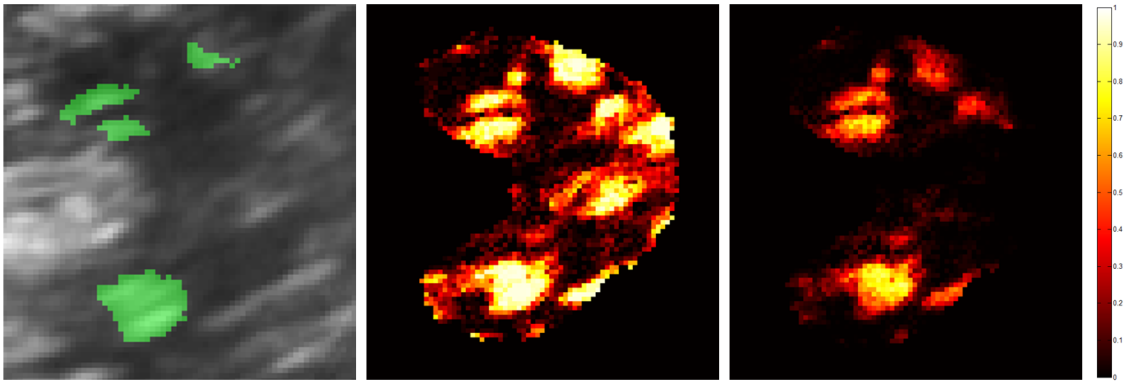

Figure 4.17: The effect of our spatial prior: From left to right, (i) the manual segmentation overlayed on the US data, (ii) the predicted posterior using the data term forest and (iii) the output after combining with the forest-based spatial prior. All outputs are probabilistic and can be thresholded to provide a binary segmentation.

	F-measure			Specificity			Sensitivity		
	Mean	**Std**	**Median**	**Mean**	**Std**	**Median**	**Mean**	**Std**	**Median**
VisForest	0.456	0.115	0.463	0.775	0.060	0.779	**0.845**	**0.081**	**0.859**
VisForest-GaussianPrior	0.508	0.155	0.547	0.819	0.045	0.812	0.829	0.113	0.844
VisForest-PriorForest	**0.519**	**0.148**	**0.574**	**0.835**	**0.043**	**0.832**	0.828	0.099	0.829

Table 4.2: Overall SNE Detection results on 22 patients: The proposed prior permits to achieve better detection by improving the specificity, i.e. by better rejecting echogenicities that do not belong to the estimated SN. Moreover, using a forest-based prior provides slightly better results.

between 0 and 1, we perform a ROC analysis, i.e. we vary the threshold's value to compute a binary segmentation, compute the corresponding confusion matrices for each run and derive different quality measures: f-measure, specificity and sensitivity.

The number of trees is set to 10 for all experiments, and best results were obtained for a depth = 15 for the VisForest, and for a depth = 10 for the PriorForest. Overall results are summarized in tab. 4.2. On the left, the best f-measure are reported by using threshold values of 0.5, 0.1 and 0.2 respectively for the VisForest, VisForest-GaussianPrior and VisForest-PriorForest models. By including our hemisphere-specific spatial prior, the f-measure is increased from **0.456** (VisForest) to **0.518** (VisForest-PriorForest). Moreover, learning this prior distribution using a random forest provides slightly better results than with Gaussian prior achieving **0.508**. On the right, the best compromise between sensitivity and specificity are computed from the ROC analysis for all approaches. As illustrated by fig. 4.17, the proposed prior permits to achieve improved specificity by better rejecting echogenicities that do not belong to the estimated SN. By varying the segmentation threshold, we also compute the area under curve which is $\mathbf{AUC = 0.903}$ for our approach, compared to a VisForest alone $\mathbf{AUC = 0.879}$ or with a simple Gaussian prior $\mathbf{AUC = 0.891}$. Detailed segmentation results are presented for each patient in fig. 4.18, and additional visual results comparing gold standard expert segmentation with our approach are pictured in fig.4.19 and 4.20.

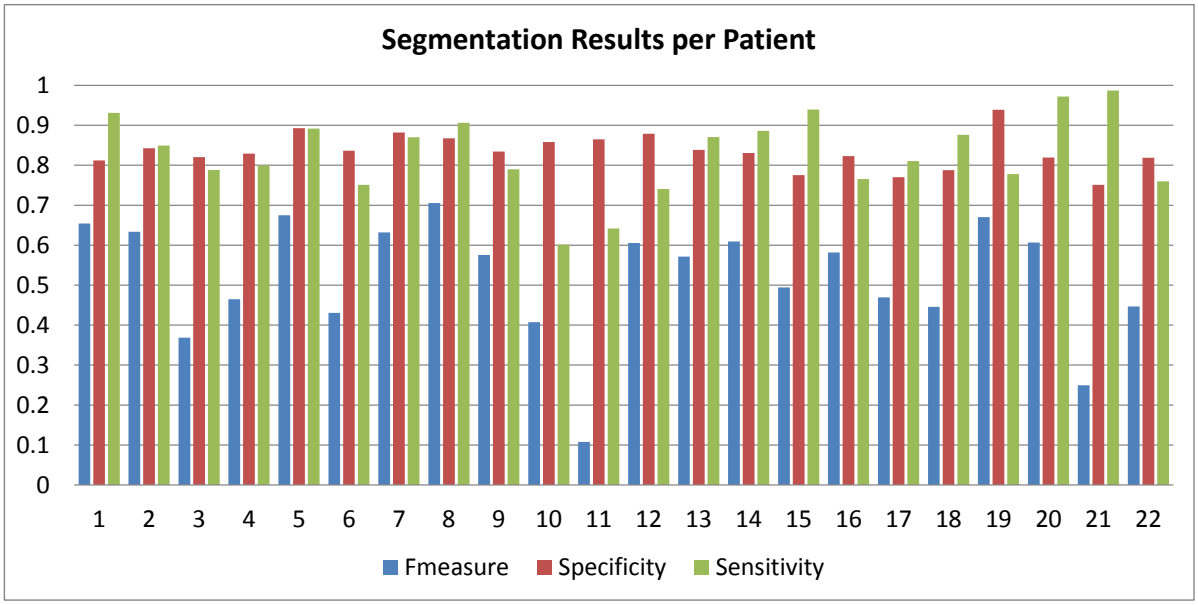

Figure 4.18: Evaluation of our SNE detection approach on 22 patients

4.3.5 Discussion and Conclusion

In this section, we presented the first approach for the automatic detection of Substantia Nigra Echogenicities in 3D TCUS. As the interpretation of such data is very difficult and yields high inter and intra-observer variability, our aim is to provide an objective and reliable segmentation of such Parkison-related speckle patches. Inspired by the way medical experts recognize SNE, we proposed a probabilistic formulation combining two discriminative models: (1) a "visual" random forest specialized on the detection of echogenicities and (2) a "spatial" random forest modeling a location prior within the midbrain. Therefore, voxel locations are parametrized within hemisphere-specific coordinate systems in order to account for asymmetric changes of orientation and scale in the midbrain anatomy. Through experimentations conducted on 22 patients data, we show promising segmentation results that seem to correlate well with expert labeling. From the segmentation output of our system, we can quantify automatically the volumetric amount of hyper-echogenicities in each hemisphere. Currently in [81], we propose to integrate this information within the very first computer aided diagnosis system for Parkinson disease based on 3D TCUS.

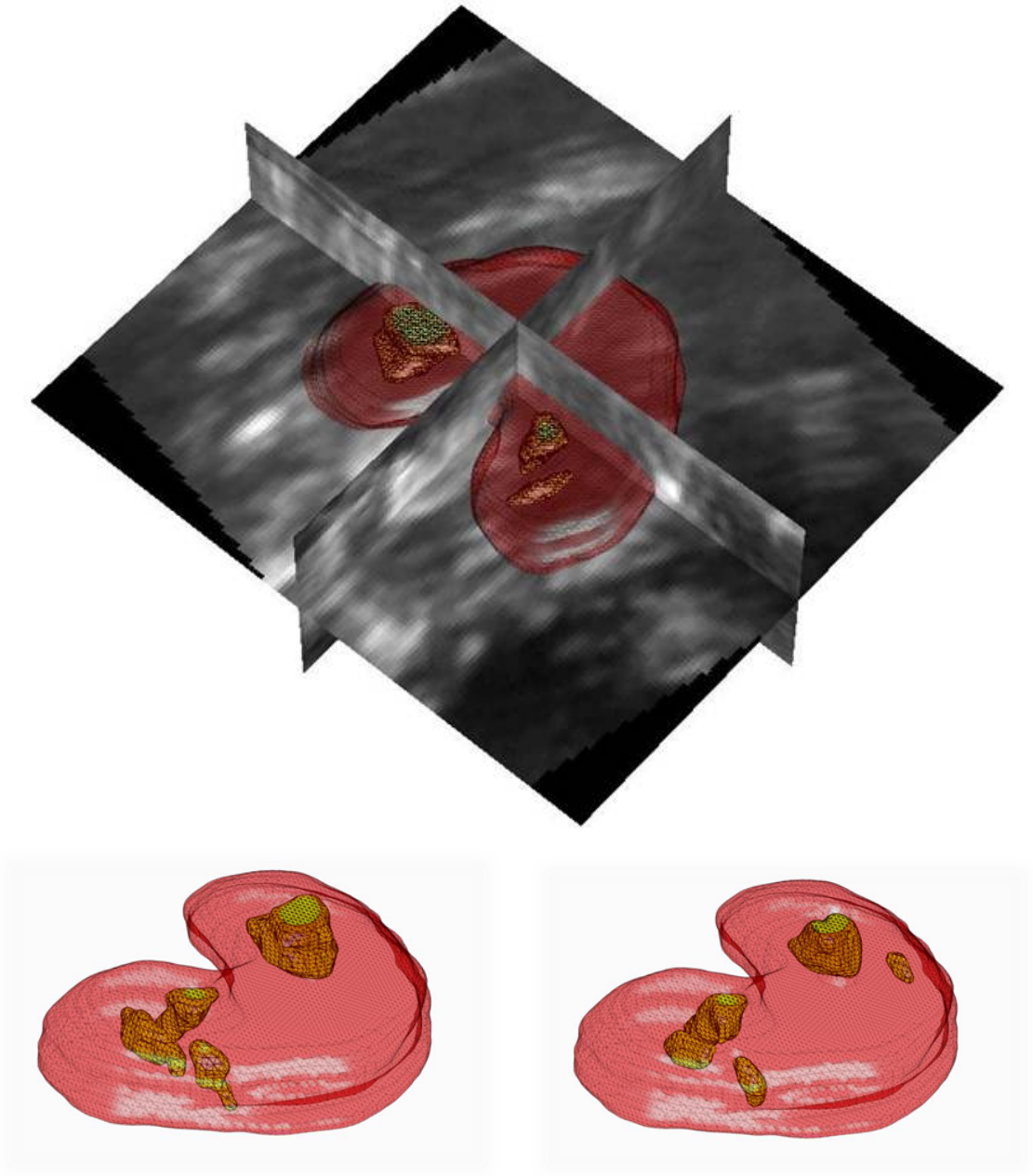

Figure 4.19: 3D visualization of the results: On top, in situ visualization of the detection results. The midbrain and the lesions are represented respectively by red and yellow 3D meshes. Below, experts annotations (left) are compared to the output of our approach (right). Our detection results seem to correlate well with experts annotations.

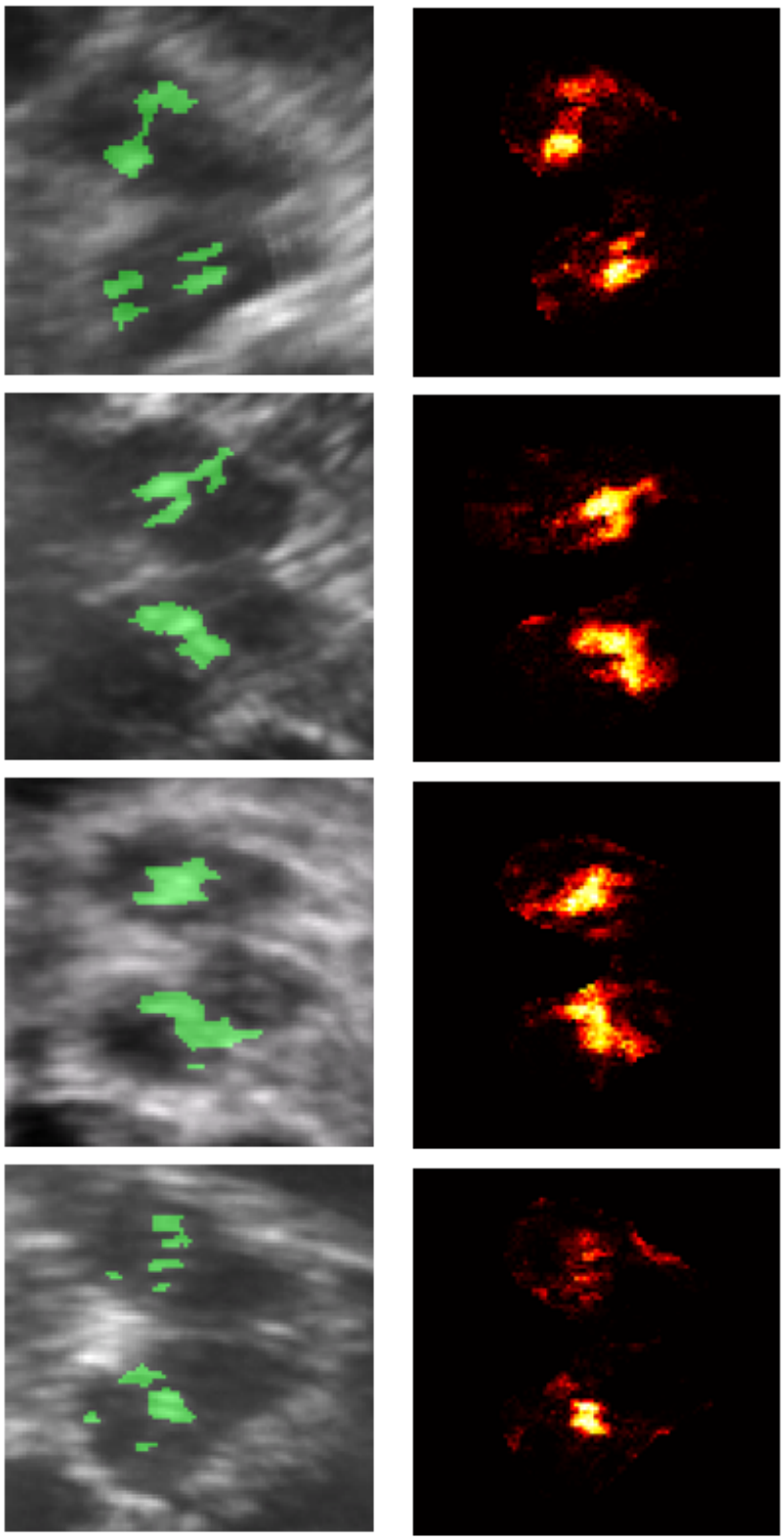

Figure 4.20: More SNE detection results: Left the manual segmentation overlayed on the US data and right, the output of our detection approach. All outputs are probabilistic and can be thresholded to provide a binary segmentation.

4.4 Content-based Modality Recognition

Introduced as a new subtask of the ImageCLEF 2010 challenge, we aim at recognizing the modality of a medical image based on its content only. Therefore, we propose to rely on a representation of images in terms of bag of words from a visual dictionary. In this section, we describe our very fast approach that allows the learning of implicit visual dictionaries which has been published in [75]. Instead of a unique computationally expensive clustering to create the dictionary, we propose a multiple random partitioning method based on Extreme Random Subspace Projection Ferns. By concatenating these multiple partitions, we can very efficiently create an implicit global quantization of the feature space and build a dictionary of visual words. Taking advantages of extreme randomization, our approach achieves very good speed performance on a real medical database, and this for a better accuracy than K-means clustering.

4.4.1 Introduction

With the goal of promoting multi-modal information retrieval, ImageCLEF proposes each year a medical retrieval challenge [65]. Made accessible by the Radiological Society of North America (RSNA), the database for this challenge contains more than seventy thousands images taken from publications that appeared in the journals Radiology and Radiographics. Consisting of 2D images in JPEG format, this collection counts medical images from different modalities, but also photographs, drawings and graphics. Moreover, many have been "processed" *e.g.* zoomed, cropped or annotated by medical experts. Because of their high variability, retrieval of medical images in such multi-modal database is a challenging task. In [49], Kalpathy-Cramer *et al.* demonstrated the importance of recognizing first the modality of images in order to improve the precision of image retrieval. Motivated by this, a new subtask was organized last year at the ImageCLEF 2010 challenge. In this section, we propose to tackle the problem of recognizing the modality of an image based on its visual content only. Since similar anatomies appear in the different classes, we can not rely on semantic information to discriminate the modalities. However, since each imaging system is based on a different physical phenomenon, resulting images show particular local visual signatures such as textural and noise patterns at small scales. For instance, ultrasound images contain particular speckle patterns while relevant information is contained in low fequency edges. Hence, to recognize modalities independently of organs or anatomical structures appearing in the images, we propose to rely on local textural and noise patterns information extracted at random positions of the image. To efficiently represent the global statistics of appearance of such local signatures in an image, a bag of visual words (BoW) can be constructed based on a visual dictionary. While K-means clustering is a classical approach to build visual dictionaries, it suffers from several limitations such as its computational cost and its dependence on the quality of its initialization.

In this section, our main technical contribution is an extreme random clustering approach to build efficiently implicit dictionaries and construct discriminative BoWs. As shown on fig.4.21, our approach begins with the random sampling of the input feature space by extracting low level visual features at random positions in the images. Then,

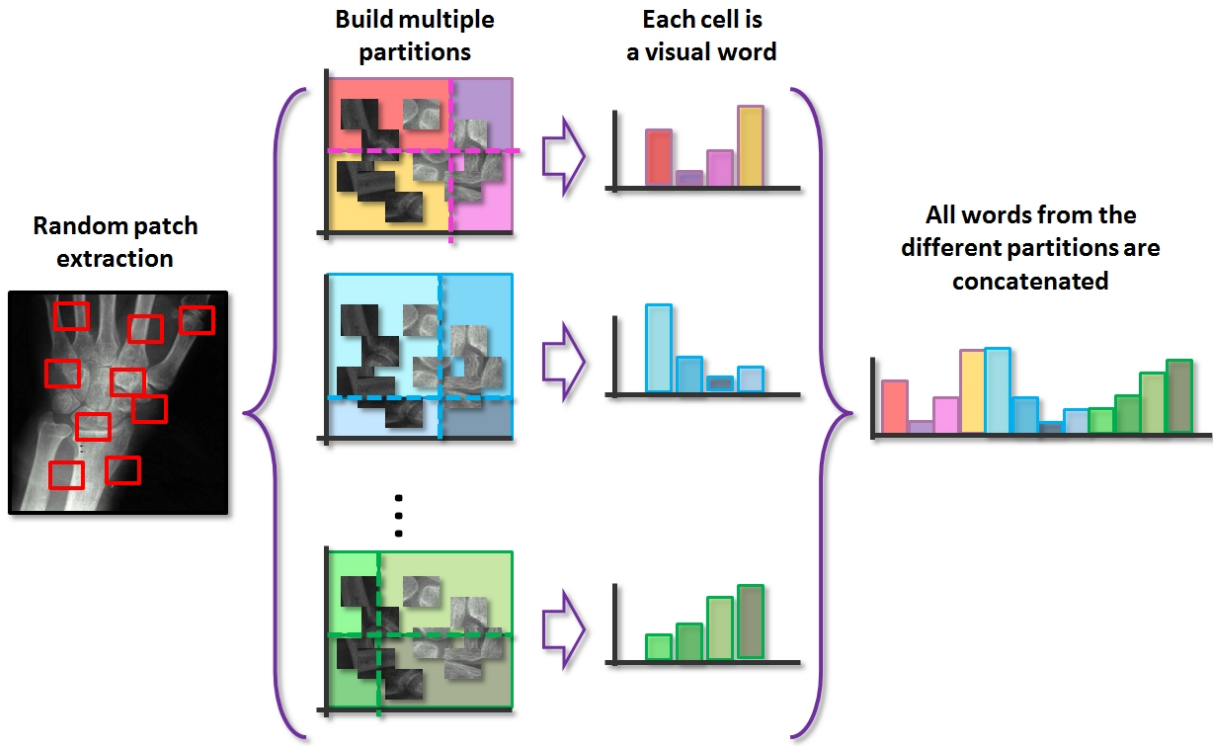

Figure 4.21: Dictionary Learning Overview: First, visual features are extracted at random positions in the images. Then, multiple independent partitions of the feature space are built. Finally, each cell of these partitions are associated to a visual words.

multiple random partitions are built using Extreme Random Subspace Projection Ferns. Finally, each cell of these partitions is associated to a visual word to form what we call an implicit dictionary. Experiments conducted on CT, MR, PET, US and X-ray images taken from the ImageCLEF 2010 database show that our approach is a fast alternative to K-means clustering which provides better performance in terms of accuracy and speed.

4.4.2 Related Work

In computer vision, bag of visual words (BoW) have become a standard representation tool for multi-class image recognition tasks. BoWs describe the content of an image in terms of the frequency of appearance of the so-called visual words. The extraction of these visual words relies on a quantization of the high dimensional space spanned by low-level visual features [106]. A classical approach to quantize the feature space is K-means clustering [90]. While K-means is highly dependent on the quality of its initialization, its major drawback is its computational cost. Indeed, performing several runs of K-means clustering in a high dimensional space may last a few days, which is not suitable for updating the BoW representation of a medical database that changes on a daily basis.

During the last decade, efforts have been made to overcome the limitations of K-means clustering to learn dictionaries. For instance, Nister *et al.* [67] introduced a

tree-based approach for CD-cover recognition relying on hierarchical K-means that shows better performance in terms of speed. Using mean shift, Jurie *et al.* [48] overcame K-means' tendency to draw cluster centers towards denser regions of the feature space. Winn *et al.* [105] generated compact dictionaries by merging some of the visual words, and this, without loss of discriminativity. In order to improve the discriminativity of dictionaries, Perronin *et al.* [80] introduced an approach that combines universal and class specific dictionaries and uses generative models. Yang *et al.* [106] proposed instead to unify the unsupervised clustering with the training of a classifier. In the same direction, Mairal *et al.* [60] presented an optimization framework to learn simultaneously a sparse representation from a dictionary and its associated classifier.

As the papers cited above, we aim at improving the learning of dictionaries. The focus of the proposed method differs from those in [80, 106, 60], as we propose to replace a single but complex clustering step by the construction of multiple random partitions to very efficiently quantize the feature space and thereby learn an implicit dictionary.

Since trees are able to identify natural clusters in high-dimensional spaces, random forests have been recently applied to dictionary learning. For instance, Moosmann *et al.* [64] and later Shotton *et al.* [89] proposed to build visual dictionaries for object categorization by using leaves and nodes from all trees of a random forest as visual words. Such dictionaries have shown state of the art performances while benefiting from fast learning and evaluation. To identify clusters with a higher resolution, Perbet *et al.* proposed in [79] to compute the intersections of all partitions and to represent them with the nodes of a graph, which is then clustered with a Markov Cluster algorithm. If this method leads to an explicit global partition of the feature space, its construction requires to solve a second clustering problem. Moreover, reducing the redundancy of the dictionary may lead to a loss of discriminativity. Following the idea of using leaves and nodes as visual words from Moosmann *et al.* [64], we propose a very efficient dictionary learning approach based on extreme randomized clustering using ferns we call ***Extreme Random Subspace Projection Ferns***, which provides a compact structure and benefits from very fast training and evaluation.

4.4.3 Proposed Method

We formulate the imaging modality recognition problem as an instance of a multi-class classification problem. Our contribution is a method to learn a visual dictionary based on extreme randomization in order to construct bag of visual words (BoWs) to represent the images we want to classify. With an extreme random partitioning algorithm we call ***Extreme Random Subspace Projection Ferns*** (ERSP), we can very efficiently construct multiple quantizations of the feature space that we then use to build an implicit dictionary. As shown on fig.4.21, our approach consists in the following steps:

1. **Extract** random points from the visual feature space.
2. **Build** efficiently multiple random partitions of the feature space with ERSP.
3. **Concatenate** these multiple random partitions.

4. **Associate** each cell to a visual word to construct an implicit dictionary.

5. **Build** bag of visual words (BoWs).

6. **Classify** using SVM with RBF kernel [19].

In contrast to K-means clustering, our approach is very fast and efficient, it is neither dependent on initialization nor requires the number of clusters to be known beforehand. Moreover, introducing randomization in the clustering phase permits to gain independence of the available training set, which in turn provides better generalization in the case of undersampled feature space, unbalanced data or noisy labeling. Finally, the proposed method can be used in a supervised as well as semi-supervised setting. Next, we describe the steps of the method enumerated above in details.

4.4.3.1 Visual Feature Space

The choice of suitable low-level visual features is crucial. In classical object recognition, features are especially designed for recognizing an object subject to different imaging conditions. Recognizing imaging modalities contrasts from classical recognition since similar objects may appear in several classes, e.g. bone structures appearing in CT as well as in X-ray images, or arteries and blood vessels that are visible in X-ray angiography and MR Time of Flight. Fortunately, the observation of medical images from different modalities shows particular textural and noise patterns at small scales. For instance, ultrasound images contain particular speckle patterns while relevant information is contained in low fequency edges. Hence, to recognize modalities independently of organs or anatomical structures appearing in the images, we propose to rely on local textural and noise patterns information extracted at random positions of the image. Hence, we propose the extraction of the following low-level visual features: Patch colors/intensities, Local Binary Patterns (LBP) [68], as texture operator to encode local color/intensity changes, and Histograms of Oriented Gradients (HOG) [23], to encode local appearance with local distributions of color/intensity gradient directions. These local visual features are computed on a set of patches that can be extracted densely or at particular keypoints of the image, and that may have different size. In the present work, we use patches of size 17×17 extracted at random positions. This patchsize allows us to capture small scale patterns independently of edges, corners or keypoints locations.

4.4.3.2 Extreme Random Subspace Projection Ferns

As explained in chapter 3, a random fern can be seen as the intersection of decision stumps which permits to partition efficiently the feature space. As shown on fig. 4.22, recall that while a tree is a set of random decision functions that split feature vectors at each node towards the left or the right branch, a fern systematically applies the same decision function for each node of the current level. This means that, in contrast to random trees, the decision function is defined in the whole feature space. Results of these random tests are finally stored as binary values, leading to a more compact and simple

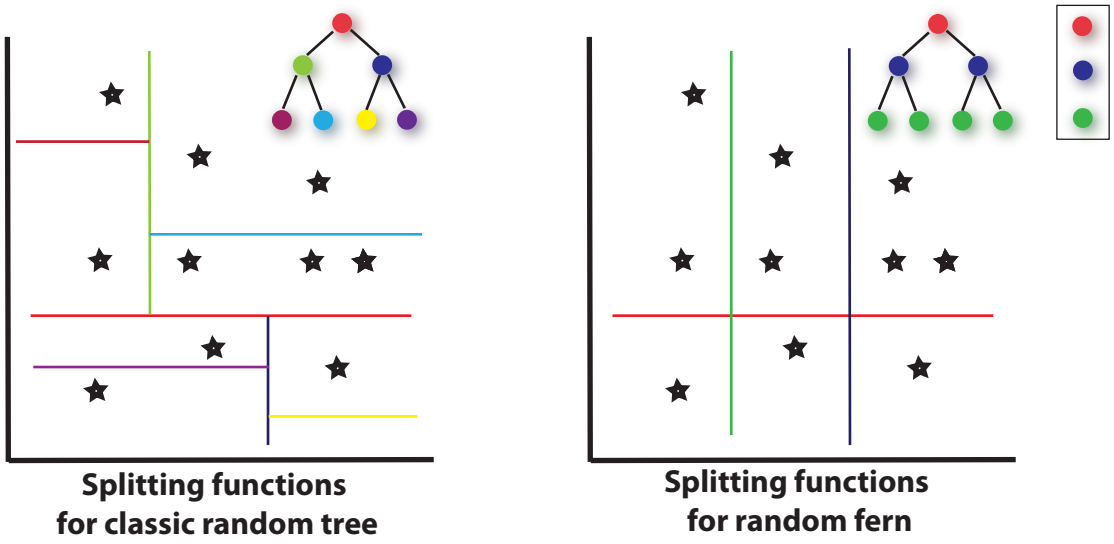

Figure 4.22: Random trees and random ferns: In contrast to a tree, a fern applies only one decision function per level. It induces splitting functions which traverse the whole feature space.

structure for performances that are similar to those of random trees [71]. Motivated by their performance we choose to use random ferns for partitioning the visual feature space.

A ferns ensemble can be built as follows: we denote by $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^{T}$ the random ferns ensemble. Each fern $\mathbf{F}_t$ is defined as a set of L binary decision functions $f_{t,l}$. The output of evaluating a function $f_{t,l}$ on a visual feature vector $\mathbf{X} \in \mathbb{R}^D$ is binary, that is $f_{t,l}(\mathbf{X}) : \mathbf{X} \mapsto \{0, 1\}$. We denote the result of the evaluation $b_{t,l}$. Hence, to an input feature vector $\mathbf{X}$ corresponds a binary vector $\mathbf{b}_t = [b_{t,1}, \ldots, b_{t,l}, \ldots, b_{t,L}]^\top$ encoding the cell of the partition where the vector falls.

In our Extreme Random Subspace Projection Fern (ERSP) approach, we combine random dimension selection and random projections [31] at each node test. Moreover, we propose to investigate the effects of pushing the randomization one step further. Instead of searching for the best threshold according to the information gain as in Extreme Randomized Trees [36], we study the use of purely random splits. Thereby, the clustering becomes independent from the training data, providing robustness to outliers or undersampled feature spaces and which permits to generate set of thresholds that better cluster non-binary separable data once they are combined.

Let us now formally describe our Extreme Random Subspace Projection Fern (ERSP) approach. We denote $\{N_l\}_{l=1}^{L}$ the nodes of a given fern. As shown on fig. 4.23, at each node N_l, we first randomly select d dimensions from the visual feature space. This means at each node we consider the set of Q "subvectors" $\{\mathbf{X}_{q,l}^{\text{sub}}\}$ from the subspace $\mathbb{R}^d$ computed from the full training set, where $d < D$. Then, each subvector $\mathbf{X}_{q,l}^{\text{sub}}$ is projected to $\mathbb{R}$ using a randomly generated unit vector $\mathbf{v}_l \in \mathbb{R}^d$: $\mathbf{X}_{q,l}^{\text{proj}} = \mathbf{v}_l^\top \cdot \mathbf{X}_{q,l}^{\text{sub}}$. The binary splitting is performed with a threshold τ_l on each projected vector $\mathbf{X}_{q,l}^{\text{proj}}$. Usually, this threshold is optimized according to the data. For instance, τ_l can be defined as the median of the projected data. In this work, we also investigate the effects of randomizing

this threshold. To summarize, the following decision function is defined on the random subspace as $f_l(\mathbf{X}^{\text{sub}}) : \mathbf{X}^{\text{sub}} \mapsto \{0, 1\}$:

$$f_l(\mathbf{X}^{\text{sub}}) \doteq \max(0, \text{sign}(\mathbf{v}_l^\top \cdot \mathbf{X}^{\text{sub}} - \tau_l)).$$

The binary partition produced at node N_l is then stored in $b_l \in \{0, 1\}$. Finally, a random fern outputs a binary vector $\mathbf{b}$ encoding the index of the cell in which a feature vector falls, *i.e.* $\mathbf{b} = [b_1, \ldots, b_l, \ldots, b_L]^\top$. Note that once the training has been performed, all nodes operations are frozen. The pseudocode describing the growing of a fern is detailed in Alg. 5.

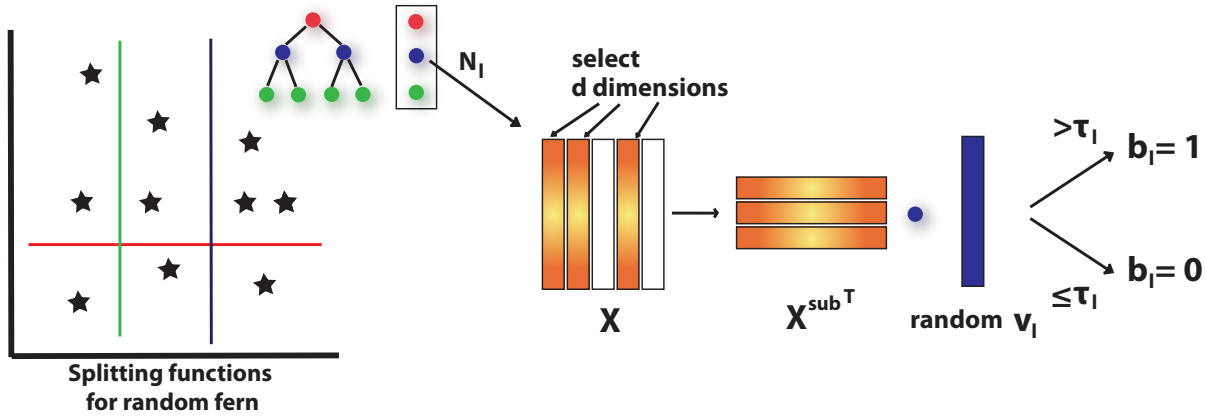

Figure 4.23: ERSP algorithm: At each node, subdimensions of the original feature space are randomly selected. Then subvectors are projected using a random vector and finally, a random threshold operation is applied.

4.4.3.3 From Multiple Independent Partitions to an Implicit Dictionary

Let us denote $\{\mathcal{P}_t\}_{t=1}^T$ the T independent partitions of the feature space built with a ferns ensemble. Each partition is defined as the set of cells:

$$\mathcal{P}_t = \left\{\mathcal{C}_t^{(1)}, \cdots, \mathcal{C}_t^{(z)}, \cdots, \mathcal{C}_t^{(Z)}\right\} \tag{4.27}$$

with cardinality $Z = 2^L$. $\mathcal{C}_t^{(z)}$ represents the cell indexed by a unique binary vector $\mathbf{b}_t = [b_{t,1}, \ldots, b_{t,l}, \ldots, b_{t,L}]^\top$ resulting from the splitting operations induced by the fern $\mathbf{F}_t$. To construct our dictionary $\mathbf{D}$, all random partitions are concatenated and each of their cells are associated to a visual word of the dictionary:

$$\mathbf{D} = \{D_m\}_{m=1}^M = \left\{\mathcal{C}_1^{(1)}, \cdots, \mathcal{C}_1^{(Z)}, \cdots, \mathcal{C}_T^{(1)}, \cdots, \mathcal{C}_T^{(Z)}\right\} \tag{4.28}$$

Since these visual words are not induced by an explicit global quantization of the space, but from overlapping partitions, the resulting dictionary is called "implicit". Now that the dictionary has been defined, each new feature vector $\mathbf{X}$ can be associated to a visual word as follows: first, $\mathbf{X}$ is passed through each ferns of the ERSP ensemble and corresponding

Algorithm 5: Pseudocode for Extreme Random Subspace Projection Fern

Input: $\{\mathbf{X}_q\}$, $(q \in \{1, \cdots, Q\})$ {input feature vectors}
Output: $\{\mathbf{b}_q\}$, {output binary vectors}
loop over the nodes
for each node N_l, $(l \in \{1, \cdots, L\})$ **do**
  select randomly d dimensions
  $\left\{\mathbf{X}_{q,l}^{\text{sub}}\right\} \leftarrow$ **selectRandomSubspace**$(\{\mathbf{X}_q\}, d)$
  generate randomly a random unit vector of dimension d
  $\mathbf{v}_l \leftarrow$ **generateRandomProjection**
  project all subvectors
  for each subvector $\mathbf{X}_{q,l}^{\text{sub}}$ **do**
    $\mathbf{X}_{q,l}^{\text{proj}} \leftarrow \mathbf{v}_l^{\top} \cdot \mathbf{X}_{q,l}^{\text{sub}}$
  end for
  generate randomly a threshold in the range of the projections values
  $\tau_l \leftarrow$ **generateRandomThreshold**$\left(\left\{\mathbf{X}_{q,l}^{\text{proj}}\right\}\right)$
  perform binary test for each projection value
  for each projection value $\mathbf{X}_{q,l}^{\text{proj}}$ **do**
    if $\mathbf{X}_{q,l}^{\text{proj}} > \tau_l$ **then**
      $\mathbf{b}_q[l] = 1$
    else
      $\mathbf{b}_q[l] = 0$
    end if
  end for
end for

output cell indexes $\{\mathcal{P}_t(\mathbf{X})\}_{t=1}^{T}$ are gathered. Then, the BoW is updated by incrementing the frequency of appearance of visual words according to these indexes. Finally, an image $\mathbf{I}$ is then represented by a the bag of words defined as:

$$\mathbf{H}(\mathbf{I}) = [P(D_1|\mathcal{X}) \cdots P(D_m|\mathcal{X}) \ \cdots \ P(D_M|\mathcal{X})] \tag{4.29}$$

where $\mathcal{X} = \left\{\mathbf{X}_q^{\mathbf{I}}\right\}_{q=1}^{Q}$ is a set of features extracted from Q random patches of image $\mathbf{I}$, and $P(D_m|\mathcal{X})$ the probability of a visual word D_m knowing $\mathcal{X}$. Finally, $\mathbf{H}(\mathbf{I})$ can be fed to a SVM classifier with a RBF kernel for modality classification.

4.4.4 Experiments and Results

In this section, we propose to recognize the modality of medical images taken from the ImageCLEF 2010 database [65]. While modality recognition was a new subtask from the medical image retrieval challenge at ImageCLEF 2010, this application brings new challenges as the database given as training set consists of classes with heterogenous content and very high variability. The same organs may appear in each modality, and different kinds of organs or anatomical structures appear within the same class. Moreover,

CLASSIFICATION RESULTS				
K-means				
Nb of clusters	1000	2000	5000	10000
F-measure	75.1%	**75.2**%	75%	74.2%
Our approach				
ferns/nodes/clusters	8/8/2048	8/10/2560	10/8/8192	10/10/10240
F-measure	76.8(74.9)%	75.8(75.1)%	76.8(75.9)%	**76.9**(**76**)%

Table 4.3: Classification results: our approach against K-means for different numbers of ferns and nodes. We investigate the effects of randomizing the choice of the threshold (results in parenthesis). Our approach provides slightly better results, even by using extreme randomization.

the database contains some multi-modal images such as PET/CT, which create overlap between classes. Note that we constrain here the problem to the most interesting and challenging modalities from the database:

- **CT**: Computerized tomography (314 images).
- **MR**: Magnetic resonance imaging (299 images).
- **PET**: Positron emission tomography including PET/CT (285 images).
- **US**: ultrasound including (color) doppler (307 images).
- **XR**: x-ray including x-ray angiography (296 images).

This leads to a total of 1501 images. **GX** (Graphics,drawings,...) and **PX** (Photographs,...) classes have been discarded for our study.

To build the dictionaries, 5000 random patches are extracted for each class which make a total of 25000 visual features. During the test phase, 1000 random features are computed to construct the BoW of an image. For each test, classes are rebalanced by using random subsampling. A grid-search is performed to find the best hyperparameters for the SVM classifier. Note that all experiments are performed with MATLAB and we use the fast Kmeans++ implementation proposed by Arthur *et al.* [6] to perform K-means clustering with a clever initialization.

Tab. 4.3 compares the classification results and tab. 4.4 the times needed to cluster the data points to create the dictionary. Our approach performs slightly better than K-means even if thresholds are randomly chosen. Moreover, while K-means needs several hours to perform one clustering run, our method clusters the data in less than two seconds. Fig. 4.24 compares the confusion matrices our approach against K-means, and fig. 4.25 presents the classification results for each class. For both approaches, most of the confusion occurs between the CT and MR, and between the two and some of the X-ray images. This is expected as CT and MR may sometimes be difficult to discriminate using only local information. Indeed, they both contain patterns showing high variability according to the chosen feature representation. Moreover, images may suffer from artifacts

CLUSTERING TIME				
K-means				
Nb of clusters	1000	2000	5000	10000
Time in ***hours***	2.3 h	**3.5** h	6.6 h	11.3 h
Our approach				
ferns/nodes/clusters	8/8/2048	8/10/2560	10/8/8192	10/10/10240
Time in ***seconds***	1.08 s	1.16 s	1.28 s	**1.41** s

Table 4.4: Clustering time: our approach against K-means for different numbers of ferns and nodes. While K-means requires a few hours to get create a good dictionary, our approach needs less than 2s.

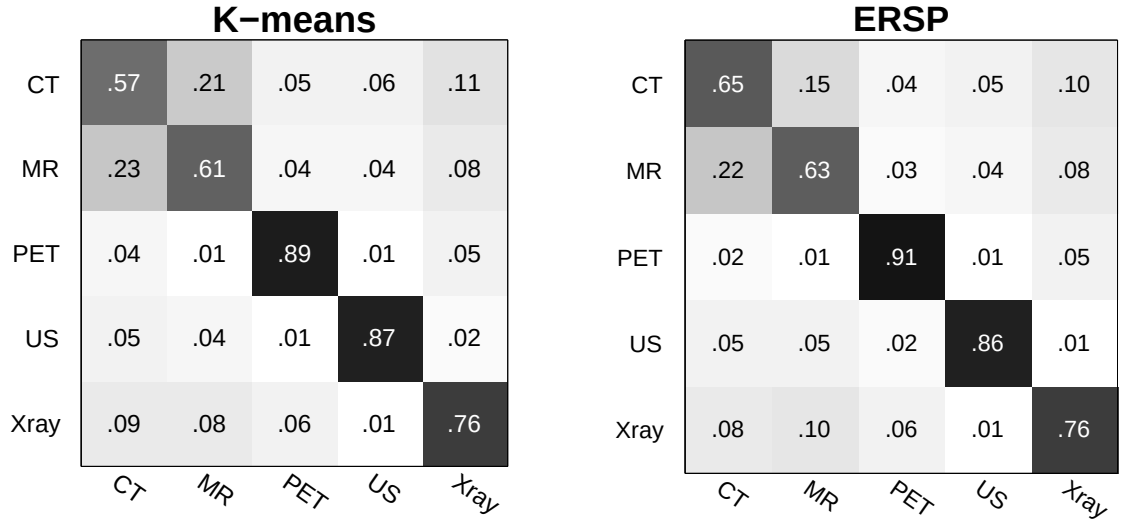

Figure 4.24: Confusion matrices: K-means (left) compared to our approach (right). Our approach outperforms K-means for almost all modality classes. For both, most of the confusion happens between CT and MR, while PET and US are well recognized due to their particular appearance.

due to the imaging system itself or to jpeg compression. Such artifacts may alter intensity patterns and distributions or worst, create artificial structures that look very similar in CT and MR. Confusions between CT and X-ray can be explained from the fact that they are based on the same physical phenomenon. Concerning MR and X-ray, confusions occur for instance in the case of cropped images of the knee which, except from the cartilage, are very dark. On the other hand, PET and US images show very discriminant patterns that can be very well separated. In the case of PET-CT images, since colors are used to represent PET signals, they can be be also well recognized. In fig. 4.26, we compare the influence of increasing the number of ferns on the f-measure, and this for both ERSP methods with and without threshold randomization. These figures suggest that with extreme randomization: (1) more ferns are needed to achieve comparable performance, and (2), the performance converges towards a limit while in the other case, we can expect further increase in the f-measure.

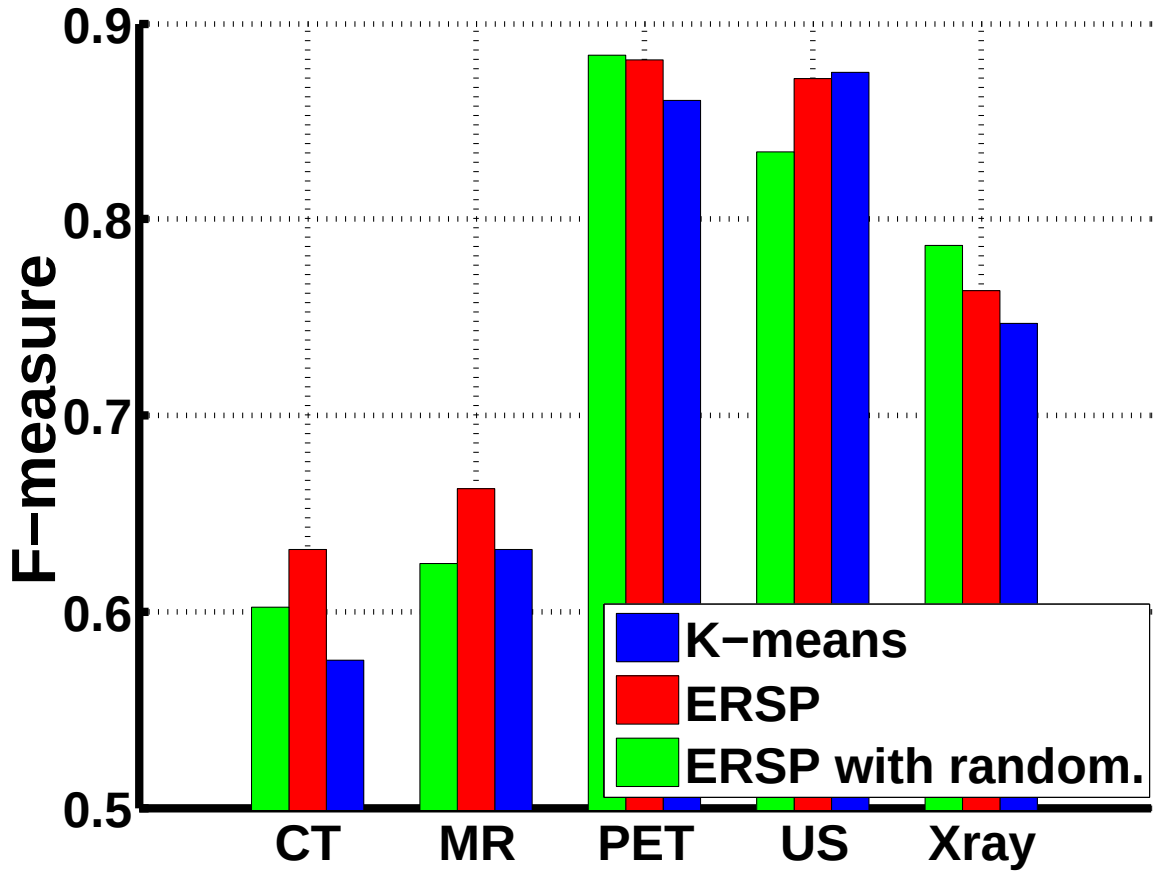

Figure 4.25: Overall classification accuracy: Comparative results for the different modalities between K-means and our approach. Our approach outperforms K-means for almost all modality classes. Again, most of the confusion happens between CT and MR, while PET and US are well recognized.

4.4.5 Discussion and Conclusion

In this section, our contribution is an approach to construct implicit dictionaries for modality recognition using extreme randomization. The backbone of our method is a clustering algorithm based on random ferns we call Extreme Random Subspace Projection (ERSP) ferns. Our approach is very fast, it provides independence from the available training set through extreme randomization, it is not highly dependent on the initialization, and it does not require the a-priori knowledge of the number of clusters. Experiments conducted on medical images from ImageCLEF 2010 database show that our approach is a fast alternative to K-means clustering for building efficiently dictionaries for multi-class classification.

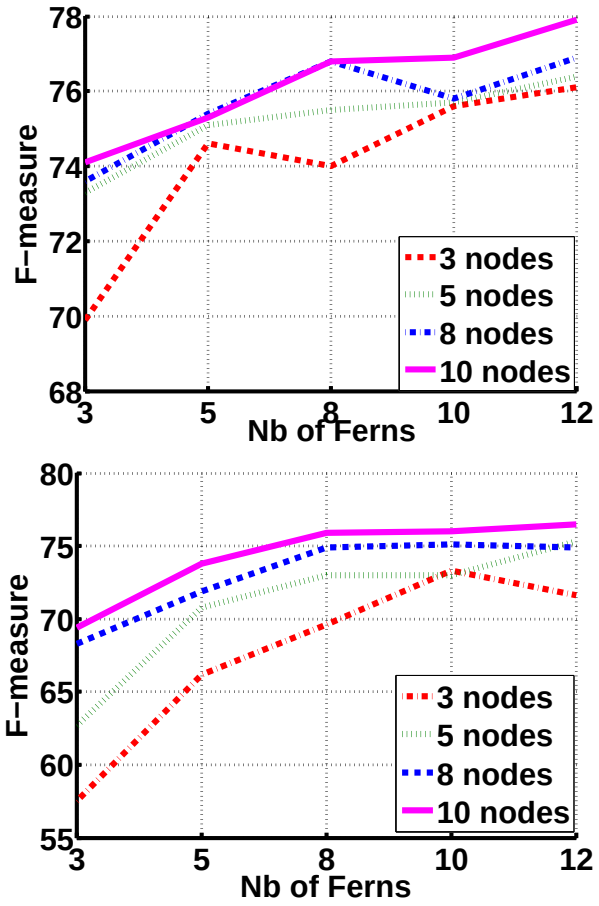

Figure 4.26: Threshold randomization: Classification results for ERSP approach without (top) and with (bottom) threshold randomization according to the number of ferns and to the number of nodes. If extreme randomization is used (bottom), more ferns are needed to achieve comparable performance. Moreover, the performance converges towards a limit while in the other case (top), we can expect further increase in the f-measure.

4.5 STARS: Several Thresholds on a Random Subspace

In this section, we propose a novel random partitioning approach we call ***STARS: Several Thresholds on a Random Subspace*** that has been published in [76]. Instead of modeling directly the posterior distribution over the entire space, we propose to divide the problem by creating multiple partitions in different random directions of the feature space. The novelty of our STARS approach resides in the fact that they consist of **multi-decisions stumps** (see fig.4.27), which permits to extract more information from each subspace. By aggregating the predictions of multiple independent STARS elements, a strong ensemble learner can be constructed. In the following, we start by motivating and defining our STARS model, and we demonstrate that it can be very efficiently implemented. Afterward, we show that STARS ensemble can be instantiated for different tasks such as classification or clustering, and this in an offline or online fashion. Furthermore, we analyze their behaviour on a few toy examples and adapt them for dictionary learning to tackle the problem of modality recognition of a medical image.

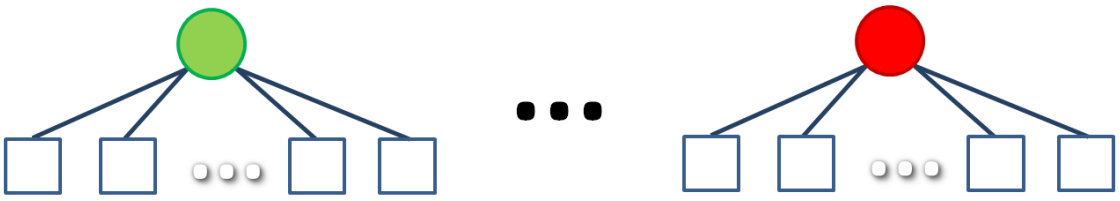

Figure 4.27: STARS ensemble: They can be seen as an ensemble of multi-decisions stumps.

4.5.1 Motivation

A wide range of computer vision applications such as face detection, object recognition or tracking can be formulated as supervised learning tasks. These latests can be modeled in a probabilistic fashion as a maximum a posteriori problem, which requires learning the class posterior distributions in a specific feature space in order to perform predictions for new incoming observations.

A major challenge is to find an efficient and memory-friendly approach for partitioning the full space and evaluating the posterior in each resulting "cell". A simple and well studied partitioning approach is the decision tree. If a tree benefits of fast training and evaluation, it suffers from some limitations. Indeed, training an optimal tree is a NP-complete problem, and with a high risk of overfitting. However an ensemble of independent trees, namely random forest, can achieve state-of-the-art performance, and this, in several applications such as tracking [54], object categorization [10], or dictionary learning [64], [89]. To achieve such performances, binary decisions are performed at each node of the trees based on a simple linear operation, which is very often a projection on a random subspace. While this permits to very fast partition the feature space, binary decisions

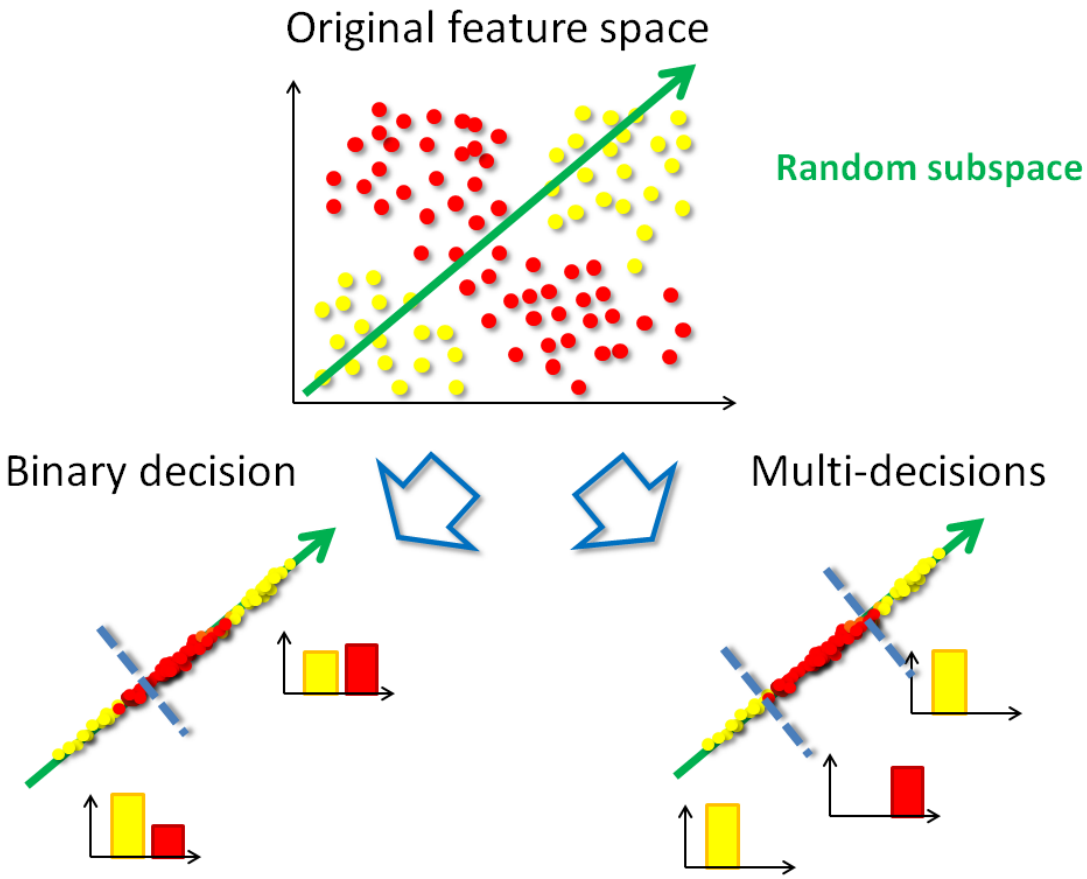

Figure 4.28: STARS motivation: Using multi-decisions permits to capture more information in a random subspace than a binary decision.

bring only limited information on the input data after projection on this random subspace. To take full advantage of the information contained in each random subspace, we propose in this section a novel partitioning structure we call ***STARS: Several Thresholds on A Random Subspace***.

In contrast to trees, a STARS can be seen as a **multi-decisions stump**, *i.e.* a single node using multiple decisions instead of a single binary decision. As illustrated by fig.4.28, the motivation comes from realizing that a single binary decision is not well adapted to handle multiple clusters. By simply increasing the number of thresholds on the subspace induced by a random projection, a full partition can be built and posteriors can be learned in each cell of this subspace. While a single STARS constitutes a weak learner, combining an ensemble of independent STARS permits to construct a strong learner. In this section, we will define the STARS model, explain how to implement it efficiently and finally discuss how to derive them for classification or clustering.

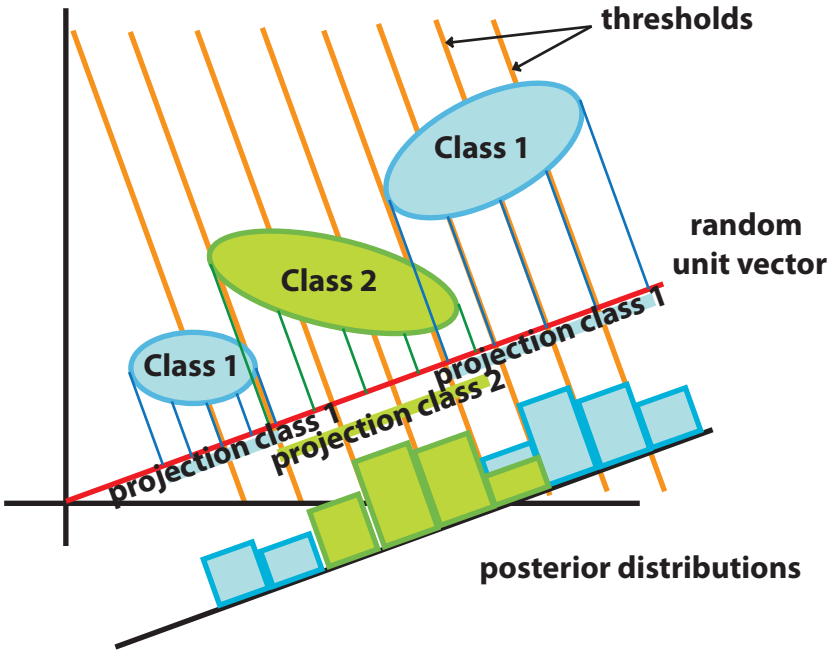

Figure 4.29: STARS model: provides a fast partitioning on a random one-dimensional subspace.

4.5.2 STARS Model

To fully exploit the information contained in each random subspace, we propose to use **multiple** decisions to partition the random subspace and fast approximate the posterior as shown on fig.4.29. In this section, we start with the formal definition of STARS elements, and then combine STARS to build a strong classifier.

4.5.2.1 Formal Definition of a STARS

From multiple decisions to a partition:
Let us denote by $\mathbf{F}$ a STARS model. $\mathbf{F}$ is defined by a random unit vector $\mathbf{v}$ of dimensionality D and a vector $\mathcal{T}$ whose entries are ordered thresholds $\mathcal{T} = (\tau_1, \cdots, \tau_b, \cdots, \tau_B)^\top$, where $\tau_1 < \tau_2 < \cdots < \tau_b < \cdots < \tau_B$ and B is the number of thresholds. Intuitively, a STARS creates a partition $\mathcal{P}$ of the subspace defined by $\mathbf{v}$ using the thresholds $\mathcal{T}$. $\mathcal{P}$ is represented by a set of "cells" or "bins" of this subspace:

$$\mathcal{P} = \left\{ \underbrace{]-\infty, \tau_1]}_{\mathcal{C}^{(1)}}, \underbrace{]\tau_1, \tau_2,]}_{\mathcal{C}^{(2)}} \cdots, \underbrace{]\tau_{B-1}, \tau_B]}_{\mathcal{C}^{(Z-1)}}, \underbrace{]\tau_B, +\infty[}_{\mathcal{C}^{(Z)}} \right\} \tag{4.30}$$

where $\left\{\mathcal{C}^{(z)}\right\}_{z=1}^{Z}$ denote the cells and Z is the total number of cells.

STARS definition:
Consider an input point $\mathbf{X}$ of feature space $\mathcal{X}$. Passing $\mathbf{X}$ through STARS $\mathbf{F}$ associates

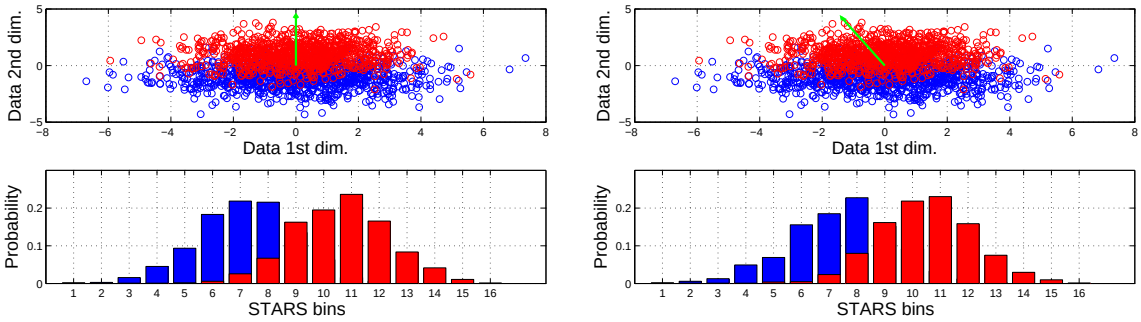

Figure 4.30: Influence of random projection: 2 overlapping classes are generated from 2 Gaussian distributions. A random direction is represented by a green vector. Below, the approximated probability distribution of the data points is plotted after projection on these 2 different directions. The performance of a STARS structure depends on the chosen direction according to the kind of data to classify.

this point with an index $z \in \{1, \cdots, Z\}$. This index corresponds to the cell $\mathcal{C}^z$ in which the projection $\mathbf{X}$ on $\mathbf{v}$ falls. We define $\mathbf{X}^{\text{proj}} \in \mathbb{R}$ to be the projection:

$$\mathbf{X}^{\text{proj}} = \mathbf{X}^\top \cdot \mathbf{v}, \tag{4.31}$$

Then $\mathbf{X}^{\text{proj}}$ is compared to each threshold contained in $\mathcal{T}$. To vectorize this operation, we use a all-ones vector $\mathbf{1}_B$ of dimension B and simultaneously compare $\mathbf{X}^{\text{proj}}$ to all thresholds as follows:

$$\mathbf{X}^{\text{bin}} = \left((\underbrace{\mathbf{1}_B}_{B\times 1} \cdot \underbrace{\mathbf{X}^{\text{proj}}}_{1\times 1}) \geq \underbrace{\mathcal{T}}_{B\times 1} \right) \tag{4.32}$$

where $\geq$ is the operator comparing the entries of two vectors. $\mathbf{X}^{\text{bin}}$ is a binary vector, whose b^{th} entry is equal to 1 when the condition $\mathbf{X}^{\text{proj}} \geq \tau_b$ is fulfilled and 0 otherwise. The cell in which the point falls is determined by 2 thresholds such that $\tau_{b-1} \leq \mathbf{X}^{\text{proj}} \leq \tau_b$. These 2 thresholds are easily detected as they corresponds to the 2 consecutive entries in $\mathbf{X}^{\text{bin}}$ containing a 1 and then a 0. The index z of the cell in which the point falls can be efficiently determined by summing the entries of $\mathbf{X}^{\text{bin}}$ and add 1:

$$z = \left\| \mathbf{X}^{\text{bin}} \right\|_1 + 1 \tag{4.33}$$

Note that since thresholds are ordered, vectors $\mathbf{X}^{\text{bin}}$ are always composed of ones entries followed by zero entries. Therefore, summing all one elements (and adding 1) allows to unambiguously determine the cell index. Finally to summarize, the STARS operation associates each point $\mathbf{X}$ to the index of the cell z in which its projection falls:

$$\begin{cases} \mathbf{F} : (\mathbf{X}) \longmapsto z \\ \mathbf{F}(\mathbf{X}) \doteq \left\| \left((\mathbf{1}_B \cdot (\mathbf{X}^\top \cdot \mathbf{v})) \geq \mathcal{T} \right) \right\|_1 + 1 \end{cases} \tag{4.34}$$

Choice of v:
$\mathbf{v}$ is chosen to have same dimensionality as the input feature space. Its entries are randomly generated in the range $[-1, 1]$. To add some randomization to the feature selection,

a subset of d dimensions are randomly selected. This permits to analyze dependencies between different subset of features and can be done by keeping only d non-zero entries in the vector $\mathbf{v}$. Finally, $\mathbf{v}$ is normalized.

Choice of $\mathcal{T}$:
While there are plenty of possibilities for setting the thresholds in $\mathcal{T}$, we use the training set to generate them in the interval defined by their projections. This is a reasonnable choice permitting to restrict the thresholds to the part of the feature space that is effectively occupied by the data points. Then, in the present work thresholds are chosen such that they create an uniform binning. While using such an uniform binning is equivalent to performing histogramming in random subspaces, thresholds can be generated on this interval using any kind of distribution.

Estimation of the posterior:
To estimate the posterior distribution on the random subspace defined by the vector $\mathbf{v}$, all points of the training set $\left\{(\mathbf{X}^{(n)}, \mathbf{Y}^{(n)})\right\}_{n=1}^{N}$ are passed through the STARS structure $\mathbf{F}$ and to compute their cell indexes. Posterior can then be learned in each cell using their corresponding training points:

$$p(\mathbf{Y}|\mathbf{X} \in \mathcal{C}^{(z)}, \mathcal{P}) \tag{4.35}$$

As shown in fig. 4.29, a STARS structure permits to build a fast partition in one direction of the feature space and to approximate the posterior in each of its cell. The choice of this direction according to the training set seems to be crucial to ensure good performance. Indeed, by using only one partitioning element, a random choice of the projection may lead to a high variance. For this reason, we propose to use STARS in an ensemble learning fashion. As stated by Freund *et al.* in [31], with a large set of random projections, the probability of capturing interesting directions of the data as is high. Moreover, using random projections is computationally very efficient.

By using multiple STARS learners, one can construct a strong learner, where each STARS builds a partition $\mathcal{P}_t$ over a different random projection of the same feature space. Once the training phase finished, posterior estimates of all individual STARS can be combined using averaging:

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} p(\mathbf{Y}|\mathbf{X}, \mathcal{P}_t) \tag{4.36}$$

In the following, we will show how STARS ensemble can be very efficiently implemented for fast learning or prediction.

4.5.2.2 STARS Ensemble: an Efficient Implementation

In this section, we detail our algorithm to create an ensemble of T STARS $\mathcal{F} = \{\mathbf{F}_t\}_{t=1}^{T}$. Then, we discuss the advantages provided by our approach compared to methods based on Fisher's linear discriminant.

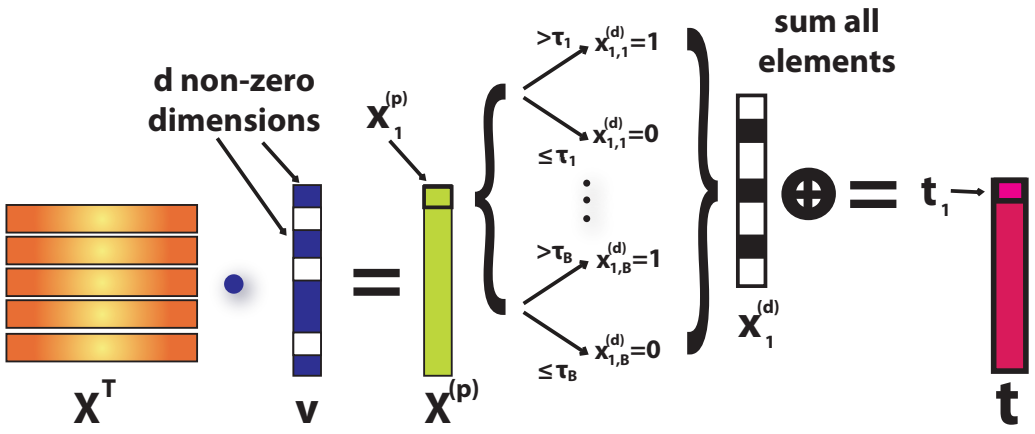

Figure 4.31: STARS algorithm: associating input feature vectors to the cells in which they falls. First the data is projected on a set of random directions. Then each projection is compared to a set of thresholds producing a binary vector. The cell index is determined by summing all entries of this binary vector and adding 1.

Efficient Ensemble learning:
One main advantage of having a simple structure such as a STARS is the possibility of high vectorization of the code permitting an efficient implementation. First a $D \times T$ matrix $\mathbf{V}$ containing all T random projection vectors $\mathbf{v}_t$ as columns is generated, where D is the dimensionality of the feature space $\mathcal{X}$. The training data contained in the matrix $\mathcal{S} \in \mathbb{R}^{N \times D}$, where each row is an observation, is projected on $\mathbf{V}$ to obtain a matrix $\mathcal{S}^{\text{proj}} \in \mathbb{R}^{N \times T}$ containing the projections of data points in all random directions. In the next step, the minimum and the maximum of $\mathcal{S}^{\text{proj}}$ are computed for each one-dimensional subspaces to setup the intervals for each STARS. A $T \times B$ matrix $\mathcal{T}$ containing all thresholds is generated using these intervals and all projections $\mathcal{S}^{\text{proj}}$ are thresholded resulting in a $N \times T \times B$ binary matrix $\mathcal{S}^{\text{bin}}$. The cell indexes stored in a $N \times T$ matrix $\mathbf{z}$ are finally computed by summing all elements of $\mathcal{S}^{\text{bin}}$ along the third dimension and adding 1 to ensure that first cell indexes are equal to 1. The pseudocode for the training of a STARS ensemble is summarized in alg.6 and illustrated in fig. 4.31. Note that all functions in alg.6 can easily be vectorized/parallelized.

After gathering the output indices $\mathbf{z}$, it is possible to estimate the posterior probabilities on each partition $\mathcal{P}_t$ associated to $\mathbf{F}_t$. As discussed in the previous chapters, there are several possibilities to combine their posteriors. In the present work, we assume all partitions are equiprobable and use a simple averaging:

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T} p(\mathbf{Y}|\mathbf{X}, \mathcal{P}_t) \tag{4.37}$$

STARS ensemble parameters:
As for trees and ferns, STARS ensemble possess only a few hyperparameters, the most important being (1) the **number of STARS** and (2), the **number of decision bins**. As STARS are weak learners, increasing their number permits to increase the prediction performance. The number of decisions per STARS is a crucial parameter as it directly

Algorithm 6: Pseudocode for STARS Ensemble

Input: $\mathcal{S} = \left\{\mathbf{X}^{(n)}\right\}, \ (n \in \{1, \cdots, N\})$ {input feature vectors}
Output: $\mathbf{z} = \left\{z^{(n)}\right\}$, {cell indexes}
create a matrix $\mathbf{V}$ *containing the random projections as columns*
$\mathbf{V} \leftarrow$ **generateRandomProjections**
project all points of the training set on all different random vectors
$\mathcal{S}^{\text{proj}} =$ **performProjections**$(\mathcal{S}, \mathbf{V})$
first compute ranges for thresholds
$\mathcal{S}^{\text{proj}}_{\text{min}} =$ **min**$(\mathcal{S}^{\text{proj}})$
$\mathcal{S}^{\text{proj}}_{\text{max}} =$ **max**$(\mathcal{S}^{\text{proj}})$
create B thresholds for each STARS so that intervals defined by the min and max are uniformly binned
$\mathcal{T} =$ **generateThresholds**$(\mathcal{S}^{\text{proj}}_{\text{min}}, \mathcal{S}^{\text{proj}}_{\text{max}}, B)$
perform thresholding and output binary vectors
$\mathcal{S}^{\text{bin}} =$ **performThresholding**$(\mathcal{S}^{\text{proj}}, \mathcal{T})$
compute the cell indexes
$\mathbf{z} =$ **sum**$(\mathcal{S}^{\text{bin}})$
$\mathbf{z} = \mathbf{z} + 1$ {ensure that first cell index is 1}

controls the number of cells or bins of the partition induced by each STARS. Consequently, it needs to be optimized to achieve a good generalization. STARS have an "optimization-free" nature and their performance is directly linked to their random projection. Hence, to provide a good partitioning of the full feature space, a STARS ensemble needs more weak learners than a forest or random ferns ensemble.

Comparison to Fisher's linear discriminant (FLD) based approach:
In an ensemble fashion, our STARS approach constitutes a strong learner even if it is based on random projections. We propose here to illustrate this by comparing it to an approach which searches for an optimal subspace before performing a decision. As FLD aims at finding the best subspace to discriminate classes, we propose to build a FLD-based ensemble learner and to compare it with a STARS ensemble on a few synthetic datasets. Note that to train such a FLD ensemble, we use different bootstraps of the training set to ensure the construction of different FLD weak learners. As shown on fig 4.32, FLD fails to discriminate multi-clusters or non-convex classes. Indeed, while STARS do not make any assumption on the class distribution, FLD makes the assumptions that classes are linearly separable, uni-modal and Gaussian distributed.

In the following, we will briefly explain how STARS can be used for different learning tasks such as classification and clustering.

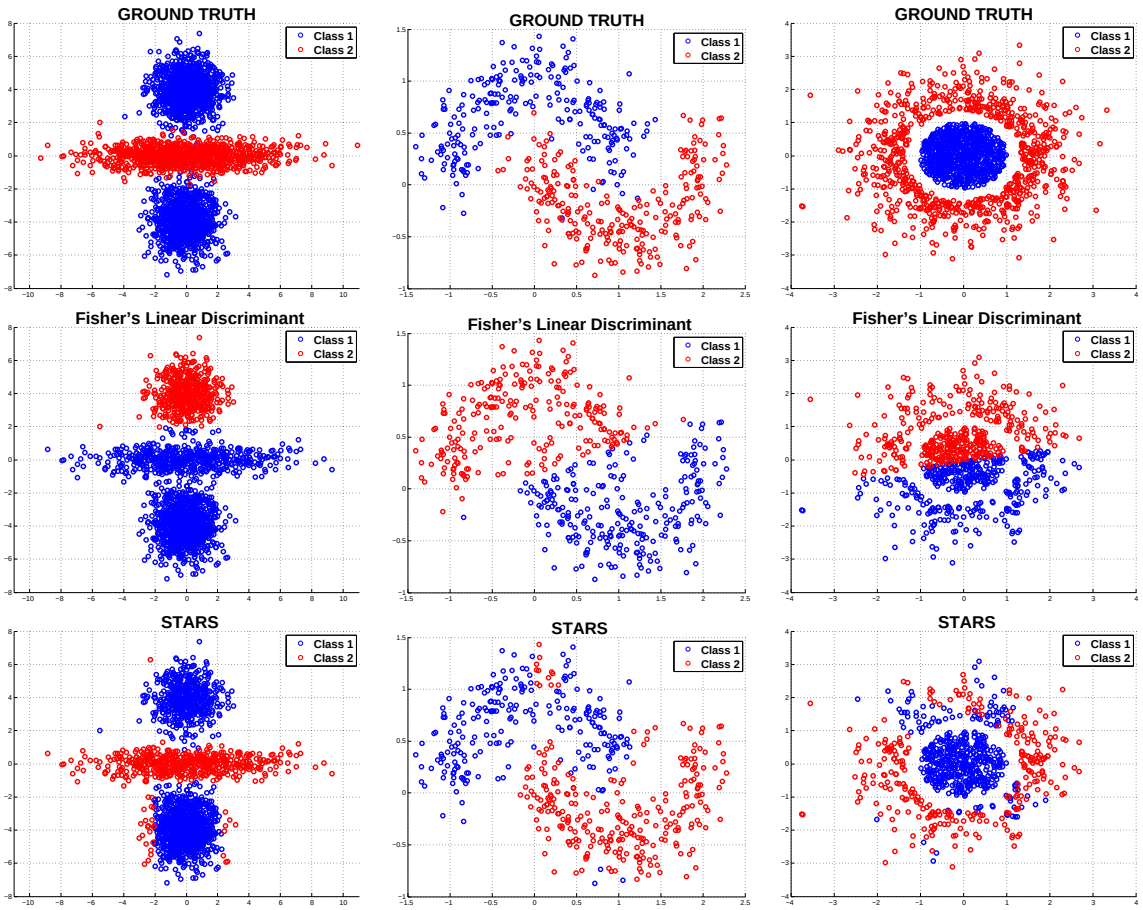

Figure 4.32: Comparison of FLD based ensemble and STARS: Clustering of synthetic datasets. In contrast to FLD, a few STARS show better ability to separate multi-clusters and non-convex classes.

4.5.3 STARS for Classification and Clustering

4.5.3.1 STARS for Classification

Let us consider a training set $\left\{\mathbf{X}^{(n)}, \mathbf{Y}^{(n)}\right\}_{n=1}^{N}$ of N observations from the feature space $\mathcal{X} \subset \mathbb{R}^D$ and their associated class labels from the finite set of labels denoted by $\mathcal{Y} = \{y_k\}_{k=1}^{K}$. The goal of multi-class classification is to learn the posterior $P(\mathbf{Y}|\mathbf{X})$, to be then able to perform predictions for an unseen observation $\mathbf{X}$ by using maximum a posteriori:

$$\hat{\mathbf{Y}} = \mathbf{argmax}_{\mathbf{Y}\in\mathcal{Y}}\, p(\mathbf{Y}|\mathbf{X}) \tag{4.38}$$

To learn efficiently this posterior, a partition $\mathcal{P}$, defined as an ensemble of Z cells $\mathcal{P} = \left\{\mathcal{C}^{(z)}\right\}_{z=1}^{Z}$, is constructed by using a STARS. It is then possible to approximate the posterior in each cell $\mathcal{C}^{(z)}$.

Pruning: For classification purposes, STARS ensemble can be efficiently pruned. Indeed, after training, STARS elements having a low discriminativity can be discarded.

Intuitively, the smaller the overlap of the posterior distributions of each class, the more discriminative is the STARS. Different measures can be designed to determine the discriminativity of each STARS, for instance using classical distribution distances such as the Kullback-leibler divergence (KL) measured over a full partition:

$$\mathrm{KL}(\mathcal{P}_z) = \sum_{\substack{\mathbf{Y},\mathbf{Y}'\in\mathcal{Y} \\ \mathbf{Y}\neq\mathbf{Y}'}} \sum_{z=1}^{Z} p(\mathbf{Y}|\mathbf{X}\in\mathcal{C}^{(z)},\mathcal{P}) \log\left(\frac{p(\mathbf{Y}|\mathbf{X}\in\mathcal{C}^{(z)},\mathcal{P})}{p(\mathbf{Y}'|\mathbf{X}\in\mathcal{C}^{(z)},\mathcal{P})}\right) \tag{4.39}$$

Intuitively, this measure takes into account the divergences between all classes and results in a high value for a discriminative partition. Finally, we simply perform pruning by discarding STARS having a KL measure which are lower than a pre-defined threshold.

Influence of STARS parameters: In this part, we propose to show the influence of the main STARS parameters, *e.g.* the number of decisions and the number of STARS. Therefore, we will use the same 3 toy examples as for forests and ferns: the "cross", "sun" and "two moons" datasets (see fig.3.5).

Let us start with a single STARS, where its projection is randomly generated and the bins are uniformly distributed within the interval defined by the features of the data points. STARS are optimization-free weak learners that approximate the posteriors on random direction. We vary only the number of bins between 8 and 32. We propose to plot the resulting posterior over the feature space using a color code varying from deep blue to red according to the posterior values for the blue and the red class.

As shown on fig. 4.34, when the number of decision bins is increasing, it builds partition with a higher resolution yielding more stripes perpendicularly to the projection vector. Again, as there are no optimization, the partition construction is not making use of the data, which explains the high variability of a single STARS. As for ferns, when the number of decision bins increases, the risk of creating empty bins is higher. Hence, a good compromise has to be found for the number of bins as it definitely has a big influence on the generalization.

Now let us set the number of bins equal to 16 and vary the number of STARS. As illustrated by fig.4.35, STARS are weak learners which performance highly depends on the random projection. Thus, increasing the number of STARS is crucial as it permits to better fit the data, achieve better generalization and get smoother posteriors, *i.e.* smoother boundaries between the classes. Fig. 4.33 demonstrates that STARS ensemble compare well with forests or random ferns.

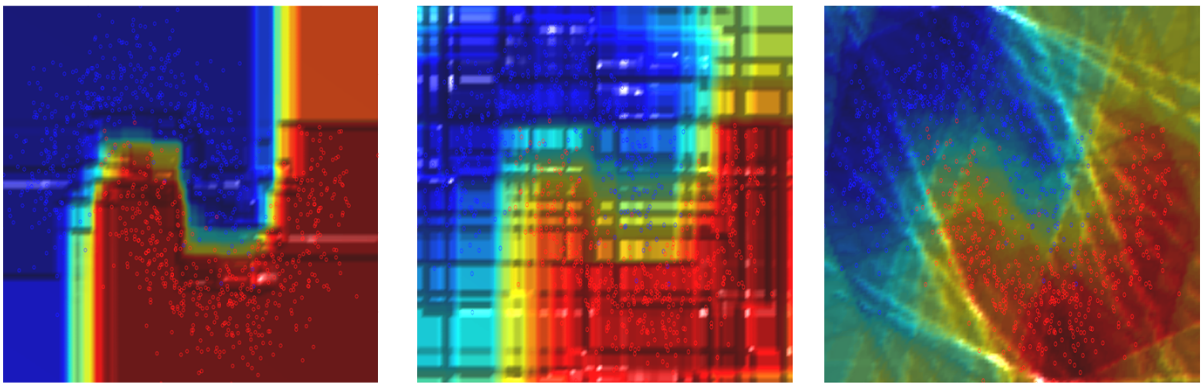

Figure 4.33: Comparison to forests and random ferns: despite its very simple structure, STARS provide a good class posterior which compares well with the other methods

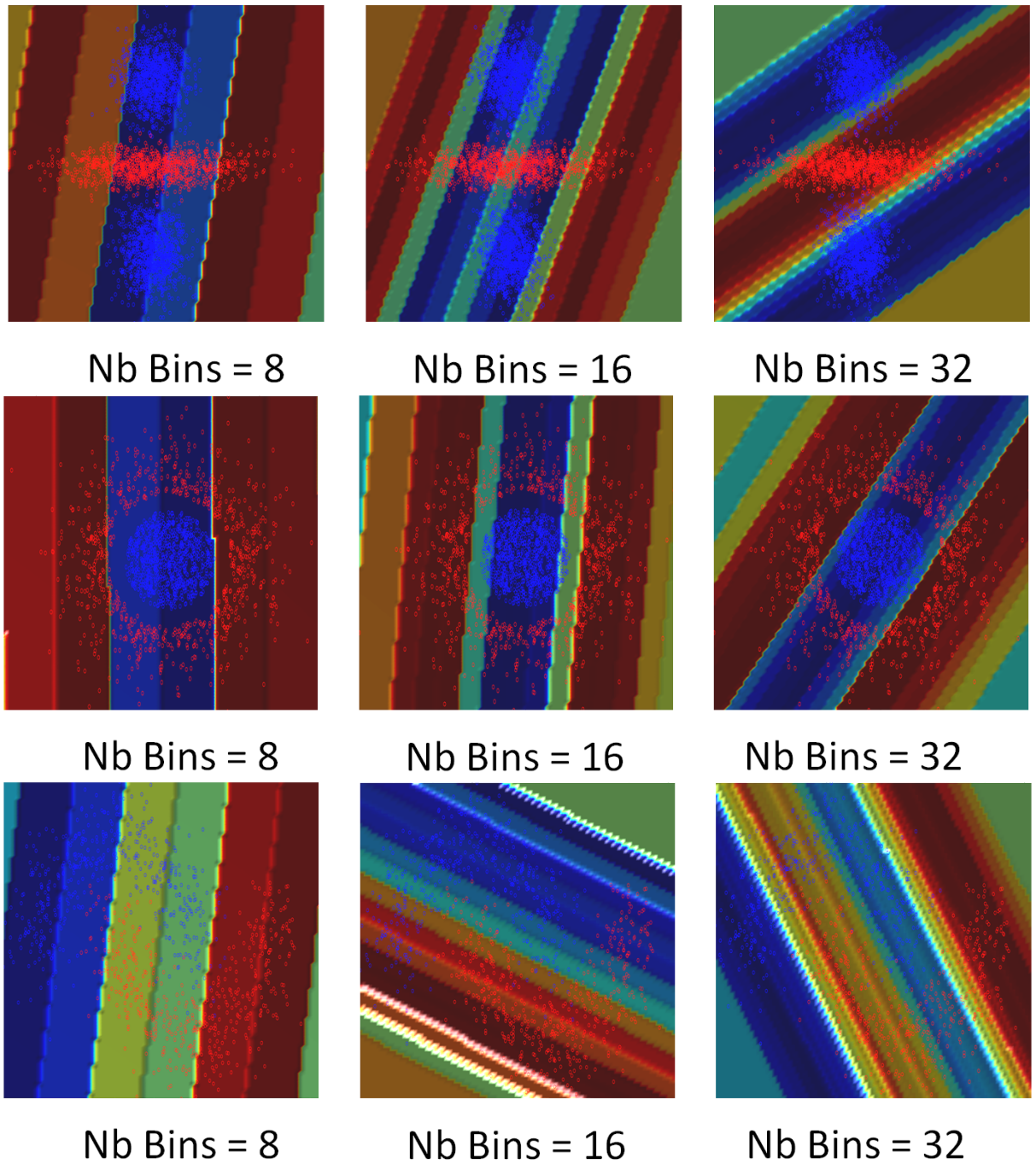

Figure 4.34: Classification posterior of a single STARS: we propose here to study the influence of the number of decision bins parameter.

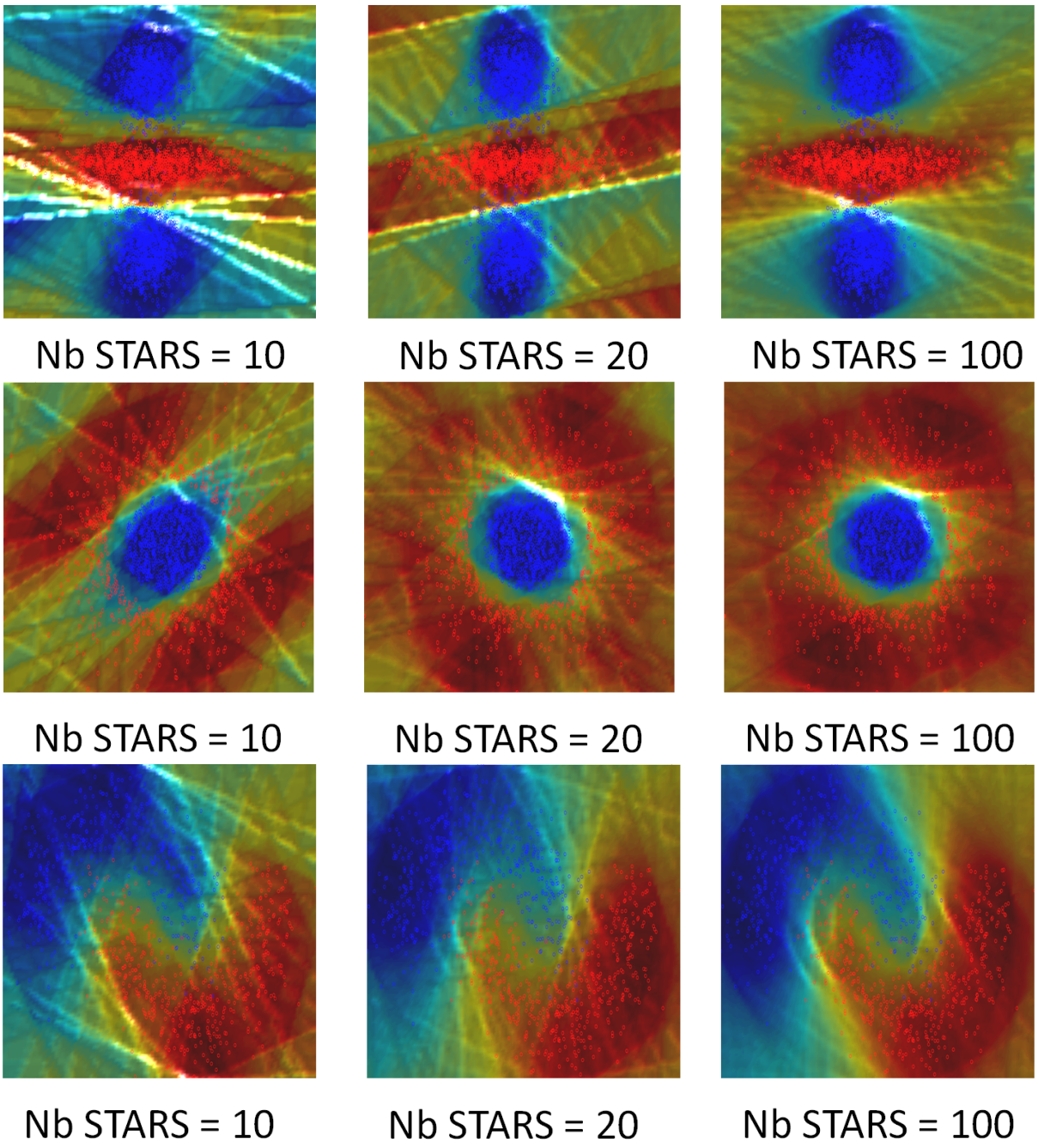

Figure 4.35: Classification posterior of a STARS ensemble: the number of bins is set to 16, we propose here to study the influence of the number of STARS.

4.5.3.2 STARS for Clustering

Considering an input feature space $\mathcal{X} \subset \mathbb{R}^D$, each cell of the partitions induced by STARS can be associated to a cluster. Hence, as for random forests and ferns, each STARS maps a point $\mathbf{X} \in \mathcal{X}$ to a cluster, and this happens by simply looking at the cell it falls in:

$$\mathbf{F}_t(\mathbf{X}) = \mathcal{C}_t^{(z_t)} \tag{4.40}$$

Again, each observation is associated to a set of clusters coming from multiple partitions in different projections of the same feature space. They can be merged into one global clustering using the 2 approaches presented in the previous chapter which are: (1) perform **intersection** between the different partitions to create a global partition and thereby global clusters, (2) keep the vectors of cells as an **implicit** representation of the global clusters.

4.5.3.3 Discussion

In this section, we presented a novel fast random partitioning approach called STARS, which can be seen as an ensemble of multi-decisions stumps. They permit to create efficiently multiple partitions in random projections of the same feature space, to finally approximate the posterior probability distribution in each cells of each random subspace. Due to their simple structure, STARS ensemble can be very efficiently implemented in a highly vectorized fashion, and they can be derived for multiple learning tasks such as classification, regression or clustering. One interesting property of STARS is that they keep track of cells neighborhood in contrast to trees or ferns. Indeed, in these latests, it is impossible to know from their structure which cells are neighboring in the feature space. In STARS, their structure reflects their cell neighborhood of the subspaces. This nice property permits for instance to perform regularization between neighboring cells while estimating the posteriors, or to easily perform soft cell assignment for incoming observations. Another idea that needs to be investigated would be to compute the cell indexes by using an intersection strategy as in ferns. While this would increase the complexity of the approach, the created cells would be better localized and provide improved posteriors.

4.5.4 STARS: Application to Content-based Modality Recognition

Now we propose to apply STARS to the problem of learning visual dictionaries to compactly represent images with bag-of-visual-words. Remind that a Bag of Visual Words (BoW) representation considers a set of feature vectors extracted locally from patches of an image. To construct a compact image representation, this feature space is quantized in ***cells***. Each cell is represented by a visual word and the whole set of visual words is called dictionary. Each local feature vector can be associated to one of the visual words of this dictionary depending on the cell of the space it belongs to. A BoW is then defined as the histogram of appearance of the visual words of the image. A classical way of building a dictionary is to use hierarchical K-means clustering [67] on the set of visual features extracted from a training set of images. Based on a hierarchical tree structure,

this approach proposes to perform a simple K-means clustering with a small amount of clusters at each node of the tree. In the end, the resulting cluster centers in the leaves of the tree define the visual words of the dictionary. Each feature vector is pushed through the whole hierarchical structure and associated to its nearest cluster center. By replacing one single hierarchy by an ensemble of trees and by getting rid of the K-means clustering step, Moosmann *et al.* showed in [64] impressive results for dictionary learning. Following this idea, we propose to use our STARS approach to create multiple partitions of the feature space, and to associate each cell of each STARS to a visual word. Finally, by passing a set of local visual features through the STARS ensemble and by gathering all output indexes, a BoW is constructed as the histogram of appearance of each cell index over the training dataset.

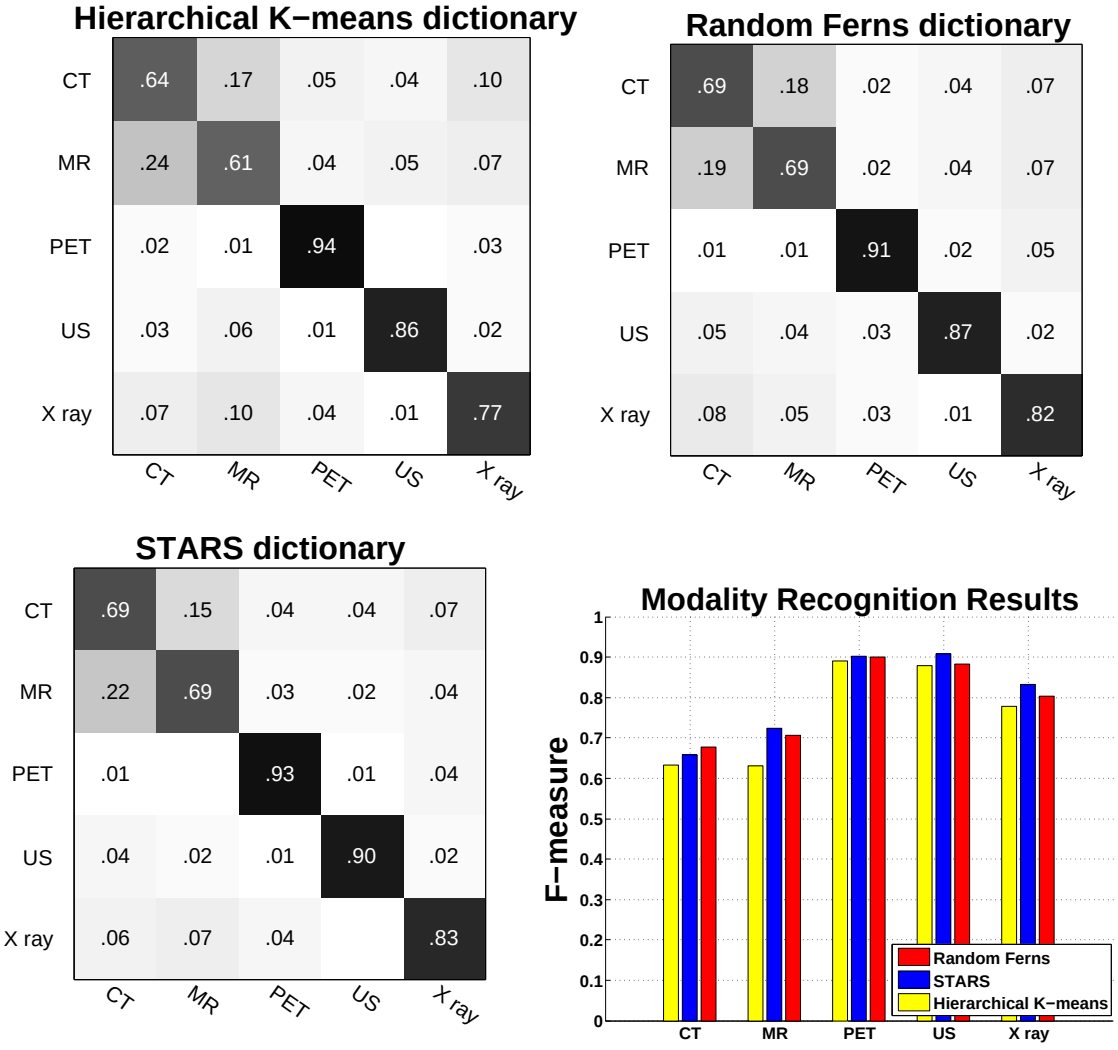

Figure 4.36: Overall classification results for modality recognition of medical images: our STARS approach (overall **81.7%**) performs mostly better than hierarchical K-means (**76.1%**) and Random Ferns (**79.43%**)

In contrast to classical image categorization, we propose here to recognize the modality

of medical images taken from the ImageCLEF2010 database [65] as in the previous section.

Assuming that local intensity patterns, texture and noise information permit to discriminate between modality classes, we use the same low-level visual features as before: patch colors/intensities, Local Binary Patterns (LBP) [68] as texture operator to encode local color/intensity changes, and Histograms of Oriented Gradients (HOG) [23] to encode local appearance with local distributions of color/intensity gradient directions. In these experiments, we compare our approach to hierarchical K-means and random ferns to build visual dictionaries. The branching factor has been set to 3 for hierarchical K-means, which means that at each node, K-means clustering is performed with $K = 3$. We investigated performances for hierarchy having different depths, note that the resulting number of visual words is then K^{depth}. For ferns, we tested different numbers of ferns/nodes and best performances have been reached with 10/10 Ferns/Nodes. For our STARS ensemble, we tested the effects of increasing the number of STARS and of bins. In the training phase, 5000 random patches are extracted for each class which make a total of 25000 visual features. During the test phase, 10000 random features are computed to construct the BoW of an image. Finally, 5-folds cross-validation has been performed on the constructed BoWs.

As shown on fig.4.36, the dictionary learning approach based on STARS clustering (overall **81.7%**) performs mostly better than hierarchical K-means (**76.1%**) and random ferns (**79.43%**). According to the confusion matrices, all three methods have similar behavior as observed already in the previous section: most of the confusion occurs between the CT and MR, and between the two and some of the X-ray images. On the other hand, PET and US images show very discriminant patterns that can be very well separated.

CHAPTER

FIVE

CONCLUSION AND OUTLOOK

"This is the end."
Jim Morrison

Along the different chapters of this thesis, our goal was twofold as reflected by its title: "Random Forests for Medical Applications."

First, we aimed at presenting random forests, a fascinating and multi-task ensemble learner, as a partitioning approach. While they have been mainly used for classification, random forests gain more and more interest for solving other learning tasks such as regression or clustering. The partition formalism we use in this thesis permits to fully understand the philosophy of random forests: **divide and conquer**. Indeed, they construct piece-wise posterior models by, (1) creating a partition over the full feature space using simple binary decisions, and (2) model the posterior distribution in each cell of this space. In fact, by defining the right **objective function** and designing an appropriate **posterior model** within the leaf, one can adapt the random forests to tackle any kind of learning problem. Afterward, we presented and analyzed the random ferns, often considered as ensemble of constrained trees. We propose instead a new interpretation of random ferns as **intersection of decision stumps**, and this permits us to better understand their advantages and pitfalls. Benefitting of a highly randomized nature, they are very fast, compact and robust to noisy data. However, as they are optimization-free, they need to be deeper than trees and may create empty cells. Finally, we proposed our own ensemble partitioning approach, namely the STARS. Constructed as ensemble of **multi-decisions stumps**, they permit to create partitions in multiple random projections of the same feature space. Due to their simple structure, they can be very efficiently implemented and derived for different learning tasks. Moreover, they respect the neighborhood structure of the cells in the feature space, which would allow for instance to perform neighborhood regularization during the learning phase, or to perform soft cell assignment.

Second, we demonstrated the great potential of forests-related approaches in different

medical applications such as multiple organ localization, segmentation, lesion detection or content-based modality recognition of medical images. While very often, a classification formulation of such tasks seems to be a natural choice, the lesson learned here is that simple classification is not always the most appropriate formulation in a field such as medical imaging. Indeed, classification relies often on local visual context only, which is in medical imaging often noisy or ambiguous. Since the human body consists of a generic anatomy, there is a huge prior information on the global context, *i.e.* on the position, shape and appearance of the different organs. By formulating our tasks as regression, joint classification-regression, or by combining visual information and spatial prior, we could demonstrate major improvement in performances and robustness. Random forests embody the perfect framework for learning-based approaches in medical applications, as they permit to integrate **application specific** objective functions and posteriors. In all medical applications, a lot of rich information remains hidden, ignored or unused while formulating the problem of interest. In future work, we aim at modeling and integrating more application-specific prior information in forest models to further demonstrate their great potential. Moreover, the different sources of information embedded in medical imaging data do not live in completely independent spaces. Indeed, they are inter-connected and should have very interesting structures that need to be explored. For instance, while formulating organ segmentation as a classification task, each voxel is only associated to the target organ label. However, human anatomy has been exhaustively studied and decomposed in multiple hierarchical systems and structures. So why not taking advantage of this hierarchical semantic information to redefine the organ segmentation problem as a hierarchical classification? Each voxel would be associated to a hierarchy of labels, and new objective functions could be designed to drive the hierarchical classification, by giving successively priority to the different levels in the hierarchy. Furthermore, most learning-based approaches that have been proposed in medical imaging follow discriminative strategies. Indeed, given an observation such as a medical data, the goal is very often to infer the desired output. Beyond this, the creation of anatomical generative models is of great interest, as they would permit to generate random anatomical samples from the learned distributions, the far goal being to learn a human anatomy generative model. While such generative models become reality for human body shape by using 3D laser scanners, the construction of anatomical models raise many challenges to capture all the variability across the population, *e.g.* by matching each anatomical structure perfectly. However, the impact of such models for medical imaging applications would be dramatic for numerous tasks such as semantic parsing, segmentation, registration or abnormality detection. Several strategies could be elaborated for the creation of modality-specific models, multi-modal models or real anatomical models. While the two firsts would be built from imaging data and could be used only in application-specific setups, the latest would require high-resolution optical data such as the one shared by the Visible Human Korean project and could be used as a latent model for all types of imaging modalities.

Finally, we also implicitely argued in this thesis for the application of machine learning approaches in the field of medical imaging. While machine learning suffered a lot from its image of "sorcery", we hopefully convinced the reader along this thesis that at least random forests and related techniques are transparent models with a fully understandable

behaviour. As many tasks in medical imaging still need to be defined in a supervised way, human expert annotations remain a crucial requirement. On one side, we would like to encourage the community to build shared databases of labelled patient data, even if we are aware of the difficulties that need to be overcome. On the other side, further efforts can be done in learning-based medical imaging to move from fully supervised approaches towards semi-supervised or unsupervised formulations. Furthermore, in the context of medical image analysis, new strategies need to be explored to facilitate the convergence between human and machine predictions. First, fast learning approaches such as the few presented in this thesis could be pushed towards efficient online learning to enable systems to evolve as new observations come in. Second, we strongly believe that novel approaches based on active learning need to be explored to open the doors for interactive human-machine learning. Indeed, transferring fully automatic frameworks into the clinical routine is challenging. Often specialized for particular applications or machine setups, such approaches might suffer from low acceptance. Designing novel human-machine interactive frameworks could lead to very efficient solutions that would be very flexible to different imaging conditions or system setups and would get better acceptance in the clinical routine.

APPENDIX

A

SIMILARITY LEARNING: CONTRIBUTIONS IN MEDICAL APPLICATIONS

In this appendix, we present our learning-based contributions for multi-modal image registration and guide-wire tracking in fluoroscopic sequences that have been published respectively in [77] and [74]. We propose a novel framework based on support vector regression to learn data-driven similarity measures. First, we show how to use this framework in the context of multi-modal image registration by learning a function which relates the space of joint intensity distributions to the registration error. Then, we adapt this framework to a deformable problem, namely the tracking of a guide-wire in fluoroscopic sequences. We first learn the distribution of guide-wire motions to reduce the complexity of the problem. Then by generating random samples from this distribution, we build a training set of local visual features and their corresponding tracking error. The data term is finally learned using support vector regression.

A.1 Similarity Learning for Multi-modal Registration of Medical Images

In multi-modal registration, similarity measures based on intensity statistics are the current standard for aligning medical images acquired with different imaging systems. In fact, the statistical relationship relating the intensities of two multi-modal images is constrained by the application, defined in terms of anatomy and imaging modalities. In this chapter, we present the benefits of exploiting application-specific prior information contained in one ***single*** pair of registered images. By varying the relative transformation parameters of registered images around the ground truth position, we explore the manifold described by their joint intensity distributions. An adapted measure is fitted using support vector regression on the training set formed by points on the manifold and their respective geometric errors. Experiments are conducted on two different pairs of modalities, MR-T1/MR-TOF and MR-T1/SPECT. We compare the results with those obtained using mutual information and Kullback-Leibler distance. Experimental results show that the proposed method presents a promising alternative for multi-modal registration.

A.1.1 Introduction

Image registration is a crucial processing step in all image analysis tasks in which information from various imaging sources needs to be combined. Establishing correspondences between images acquired with different medical imaging modalities is a challenging task known as multi-modal registration. Objective functions that evaluate the quality of alignment, known as similarity measures, are optimized to identify the geometric transformation that maps the coordinate system of one modality to the other [115]. The choice of the appropriate measure is not straightforward, because it implicitly models the relationship between the different images to register [84]. Classical measures such as sum of square differences (SSD) or correlation coefficient (CC) make the assumption of a linear functional mapping between the intensities of the images to align. But this hypothesis is far from being realistic according to the physics of different imaging systems. Modeling the real relationship between different imaging modalities is very difficult and this explains why statistical measures have become more and more popular. Since its introduction by Viola and Wells [101] and Collignon et al [18], mutual information remains the state of the art of multi-modal registration of medical images.

Even though the statistics relating intensities of two multi-modal images is modality-specific, there were only few attempts to incorporate prior knowledge in such similarity measures. Chung et al. [17] proposed to use as prior information a reference joint probability distribution of registered images from different modalities. Images are then aligned by minimizing the Kullback-Leibler distance between an observed and the expected joint histogram. Leventon et al. [55] compared two methods to model this reference histogram from a training set of registered images, namely a mixture of Gaussians and Parzen windowing. The distance to this expected histogram is then estimated by using log likelihood. In these works however, the use of prior information remains limited to one reference joint distribution.

Zhou et al.[113] propose an approach based on Adaboost to learn local similarity measures for anatomic landmarks detection in echocardiac images. It uses an atlas of the left ventricle containing pairs of local patches with their relative displacements. In a mono-modal scenario, the method shows that incorporating prior information can improve the detection results. This approach requires however extensive initial supervision.
Joint histograms of multi-modal images warped with different relative transformations describe a manifold embedded in the joint distribution space. Our contribution is to define a similarity measure relating the topology of such manifolds to the registration error. This yields an application-specific similarity measure, which requires one single pair of registered images as prior information. Using a set of relative transformations between the two images, we generate a training set of data points from the corresponding joint histograms and their associated geometric error values defined in section A.1.2. The similarity measure is then learned by performing a support vector regression on this data.

The remainder of the chapter is organized as follows: Section A.1.2 presents our regression approach to define an application-specific similarity measure. Section A.1.3 reports experiments performed in two different and challenging applications in comparison to classical methods such as mutual information and Kullback-Leibler distance. Results show that our approach presents a promising alternative for multi-modal registration. Section

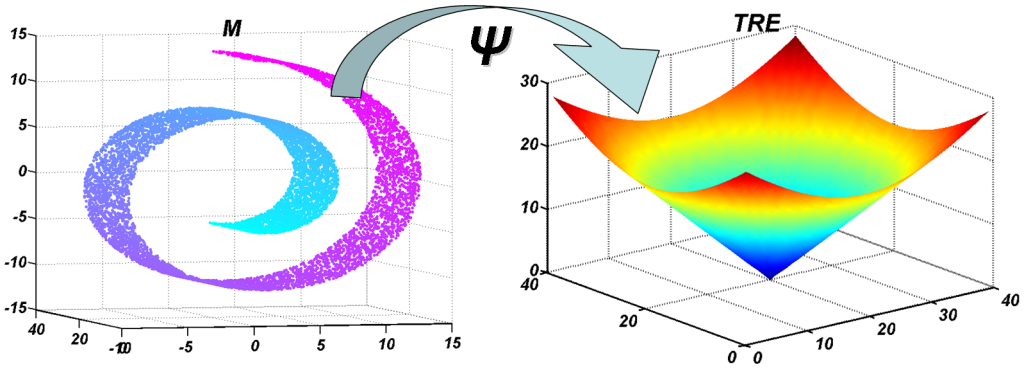

Figure A.1: Our regression approach: learn a similarity Ψ mapping each point of the manifold $\mathcal{M}$ (abtract representation on the left) to a value of the geometric error (on the right).

A.1.4 concludes the paper and gives an outlook on future work.

A.1.2 Methods

A.1.2.1 Problem statement

The goal of multi-modal image registration is to identify the geometric transformation that maps the coordinate system of one modality to the other. Let us consider two 2D images defined on the domains Ω_1 and Ω_2 with intensity functions $I_1 : \Omega_1 \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ and $I_2 : \Omega_2 \subset \mathbb{R}^2 \rightarrow \mathbb{R}$. The two dimensional case is discussed for better readability, the extension to three dimensions being straightforward. The registration task can be defined as a maximization problem, in which we want to estimate the best transformation T according to a chosen similarity measure S computed on the discrete overlap domain $\Omega = \Omega_1 \cap T(\Omega_2)$:

$$T = \mathbf{argmax}_T \; S_\Omega(I_1, T(I_2)). \tag{A.1}$$

The joint intensity distribution $p(I_1, I_2)$ of both images can be evaluated by histogramming or parzen windowing. In most of statistical measures, the similarity S_Ω is a mapping from the joint distribution space $\mathbb{J}$ into $\mathbb{R}$. While Mutual Information (MI) gives a measure of the distance between the joint histogram of both images and what it would be if their intensity distributions were independant, the Kullback-Leibler distance (KL) [17] evaluates the distance between an observed p_o and an expected p_e joint histogram:

$$MI(I_1, I_2) = D(p(I_1, I_2)||p(I_1)p(I_2)) \tag{A.2}$$
$$KL(I_1, I_2) = D(p_o(I_1, I_2)||p_e(I_1, I_2)), \tag{A.3}$$

where D in its general form is defined on two histograms p and q as:

$$D(p||q) = \sum_x p(x) log\left(\frac{p(x)}{q(x)}\right). \tag{A.4}$$

The statistical relationship relating the intensities of two different multi-modal images is constrained by the application. With “application", we mean the combination of the

modalities to relate and the different tissues appearing in the imaged anatomy, e.g. blood, bones or muscles. Joint histograms of images warped with different relative transformations describe a manifold $\mathcal{M}$ embedded in $\mathbb{J}$ which is application-specific. In [17], Chung et al. makes use of one expected joint histogram, corresponding to one single reference point on such a manifold. The used Kullback-Leibler divergence is however not adapted to its topology.

Instead, we propose to model an application-specific similarity Ψ taking into account how the topology of $\mathcal{M}$ relates to the registration error. By using a set of relative geometric transformations $\{T_i\}_{i\in\mathbb{N}}$ between a source and a target image, we sample $\mathcal{M}$ by the joint histograms $\mathcal{J}_{I_1,T_i(I_2)}$. Each of these "points" is then associated to a geometric error derived from the corresponding transformation parameters, generating thereby a set of data points. Finally, the similarity Ψ is defined by performing a regression on these points. The following section presents how to generate data points to relate this manifold $\mathcal{M}$ to the geometric error.

A.1.2.2 Data points generation

Our objective is to model a similarity Ψ learned on the full manifold $\mathcal{M}$:

$$\Psi : \mathcal{M} \to \mathbb{R}, \tag{A.5}$$

which has favorable characteristics for registration purposes, namely convexity, smoothness and the ability to estimate the geometric error. To model an accurate mapping Ψ, the manifold $\mathcal{M}$ must be sampled thoroughly as a function of the transformation T, whose space is parameterized as follows:

$$T(t_x, t_y, \theta) \text{ where } \begin{cases} t_x \in [-M, +M] \\ t_y \in [-N, +N] \\ \theta \in [-\phi, +\phi] \end{cases} \tag{A.6}$$

By sampling the space of transformations, a set $\{T_i\}_{1\leq i\leq Q}$ of Q transformations is generated. Then, by using a pair of registered images from different modalities, joint histograms are computed according to these $\{T_i\}_{1\leq i\leq Q}$. As illustrated by Fig. A.1, each joint histogram is then associated to a geometric error value. In medical image registration, the *target registration error* (TRE) permits the evaluation of error in translation and orientation between corresponding structures or organs appearing in both modalities. The TRE is computed by comparing the positions of a set of points $\{p_i,\ 1 \leq i \leq P\}$ after being mapped by the estimated transformation T and by the ground truth transform G:

$$\mathcal{E}(T) = \frac{1}{P} \sum_{i=1}^{i=P} \|T(p_i) - G(p_i)\| . \tag{A.7}$$

This procedure permits us to generate following couples:

$$\left\{(\mathcal{J}_{I_1,T_i(I_2)}, \mathcal{E}(T_i))\right\}_{1\leq i\leq Q}, \tag{A.8}$$

which we denote $\{(\mathcal{J}_i, \mathcal{E}_i)\}_{1\leq i\leq Q}$ for better readability.

A.1.2.3 Fitting the similarity model through support vector regression

We propose to learn the similarity by approximating the function Ψ with the previously generated data points. Since this function is a high dimensional non-linear mapping, we use support vector regression for its ability of modeling complex non-linear functions. We consider the problem of fitting a similarity function on the set of Q data points $\{(\mathcal{J}_i, \mathcal{E}_i)\}_{1 \leq i \leq Q}$. The $\{\mathcal{J}_i\}$, as discrete approximations of the joint intensity distributions, consist of $B \times B$ bins. They are linearized into a vector of dimensionality B^2. Let φ be a non-linear mapping from $\mathcal{M}$ into a hidden feature space $\mathcal{H}$ with dimensionality $\mathbf{dim}(\mathcal{H}) > B^2$ used to model non-linear relationships between joint histograms and their corresponding geometric error values. The mapping Ψ is modeled by the following function:

$$\Psi(\mathcal{J}) = w \cdot \varphi(\mathcal{J}) + b, \tag{A.9}$$

where w is a linear separator of dimensionality $\dim(\mathcal{H})$ and b a bias. The optimal regression function is then given by the minimum of the following functional [91]:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{Q} \left(\xi_i^+ + \xi_i^-\right), \tag{A.10}$$

where C controls the flexibility of the model. This functional aims at minimizing the norm of w and the regression errors on the data points, characterized by the slack variables ξ_i^+ and ξ_i^-. The optimal vector w_0 can be written as a linear combination of the training vectors in $\mathcal{H}$ with weights $\{\alpha_i\}_{1 \leq i \leq Q}$:

$$w_0 = \sum_{i=1}^{Q} \alpha_i \ \varphi(\mathcal{J}_i). \tag{A.11}$$

The regression function becomes then:

$$\Psi(\mathcal{J}) = \sum_{i=1}^{Q} \alpha_i \ \varphi(\mathcal{J}_i) \cdot \varphi(\mathcal{J}) + b = \sum_{i=1}^{Q} \alpha_i \ \mathbf{K}(\mathcal{J}_i, \mathcal{J}) + b, \tag{A.12}$$

where $\mathbf{K}$ is the kernel associated to φ in $\mathcal{H}$. To handle non-linear relations between the manifold $\mathcal{M}$ and the TRE, $\mathbf{K}$ is chosen as a RBF kernel, giving thus the following similarity model:

$$\Psi(\mathcal{J}) = \sum_{i=1}^{Q} \alpha_i \ \exp\left(\frac{-\left|\mathcal{J}_i - \mathcal{J}\right|^2}{\sigma^2}\right) + b. \tag{A.13}$$

A.1.3 Experiments and Results

Our regressed similarity measure is evaluated on two challenging applications for multi-modal registration: rigid registration of MR-T1 and MR-TOF (Angiography) images of the carotid artery, and of MR-T1 and SPECT images of the brain. In this paper, we focus on 2D rigid-body experiments to prove the concept of our novel approach.

This permits in particular to show that the approach is not limited to pairs of images with a tissue distribution similar to the image pair used for training. Indeed, in the

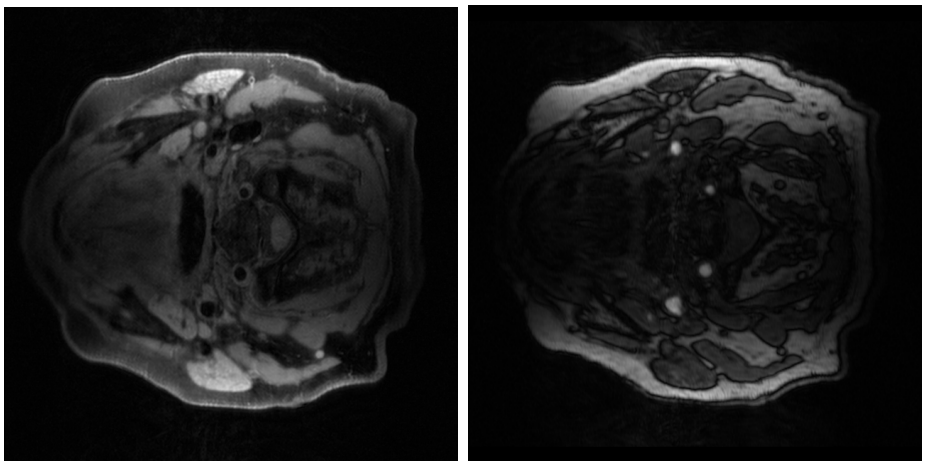

Figure A.2: Multi-modal Images used in our experiments: T1 and TOF MR Angiography of the neck of the same patient.

following experiments, a pair of corresponding images from a 3D dataset is used for training. The obtained similarity measure is then evaluated on pairs of images taken from the 3D datasets of the other patients. For statistical relevance, the pairs are chosen randomly and the tests are repeated. It must be noted that the tissue distribution varies depending on the randomly chosen slices, which can originate from the neck or from the head.

A.1.3.1 Experimental Setup

Our similarity measure will be compared to normalized mutual information (NMI) and Kullback-Leibler distance (KL) in terms of *success rate*, *accuracy* and *capture range*. We consider a registration experiment as *successful* when the final target registration error is inferior to a given threshold t_e. In fact, this permits to quantify the ability of an approach to converge in the neighborhood of the right solution. We then define the *accuracy* as the mean target registration error on all registered images after the removal of such outliers. *Capture range* is evaluated by assessing the success rate as function of an increasing initial TRE. Knowing the ground truth position of each dataset, an initial random perturbation is applied to each pair of images according to a given value of TRE. Experiments are then repeated with an increasing initial target registration error.

The objective of our experiments is to highlight the benefits of a similarity measure taking advantage of prior information. Since the convergence to the right solution depends on the topography of the search space offered by a similarity measure, we use a Downhill-Simplex optimizer, that does not require any gradient information. For fair comparison, all measures have the same number of joint histogram bins (32×32) and are tested in the same conditions.

In both experimental setups T1/TOF and T1/SPECT, a cross-validation of N tests is performed on a set of P patients. A test consists of one ***regression step*** performed on a random pair of slices from a given patient and one ***validation step*** consisting of

$P-1$ evaluations performed on the other $P-1$ patients. During the regression step, our similarity measure and the expected joint histogram needed by KL are computed on the same pair of images. During an evaluation, all measures are tested in the same conditions on a random pair of slices taken from another patient with the same initial perturbation. By using 10 initializations with an increasing TRE per evaluation, we can investigate the ability of each measure to converge towards the right solution and thereby assess their capture range.

The transformation space is sampled as follows: $-40 \leq t_x \leq +40$ (in pixels), $-40 \leq t_y \leq +40$ and $-40 \leq \theta \leq +40$ (in degrees) with a step of 4 for each parameter, generating thereby 9261 data points. For the choice of the hyperparameters σ and C, a grid-search has been performed. All experiments are performed with the Spider environment for MATLAB on an Intel Core 2 Duo CPU 2.40 GHz.

MR-T1 and MR-TOF Angiography images: experiments are conducted on images (refer to Fig. A.2) taken from $P = 8$ patients (48 pairs of images) with different staging of atherosclerosis. Both sequences were consecutively acquired, patients were positioned on a vacuum pillow and the acquisition was ECG gated to ensure perfect alignment. Images have a resolution of 128x128 with a pixel size of 2.5mm x 2.5mm. The threshold t_e is set to 1cm which corresponds to 4 pixels. A cross-validation of $N = 32$ tests has been performed, which then corresponds to $N \times (P-1) \times 10 = 2240$ registration experiments.

MR-T1 and SPECT-Tc images: experiments are conducted on images taken from $P = 5$ patients (73 pairs of images): a healthy patient, one with a glioma, one with a carcinoma, one with a stroke and finally one with an encephalopathy. These already registered datasets are taken from the publicly available Whole Brain Atlas database. Images have a resolution of 128x128 with a pixel size of 1.67mm x 1.67mm. The threshold t_e is set to 1cm which corresponds to 6 pixels. A cross-validation of $N = 40$ tests has been performed, which then corresponds to $N \times (P-1) \times 10 = 1600$ registration experiments.

A.1.3.2 Results

The objective of our experiments is to show the benefits of a similarity measure taking full advantage of prior information. As shown on Fig. A.3, the optimal regression model provides a smooth and convex search space, which is very close to the original TRE surface to approximate. Moreover, the global optimum has been preserved at the right position. In fact, smoothness and convexity are crucial characteristics to prevent the optimizer of being stuck in a local optimum and to ensure its convergence to the global one. The great advantage of our approach is its ability to model the convexity, the smoothness and the capture range of the similarity measure. Indeed, its convexity can be changed by choosing another function of the geometric error. The choice of hyperparameters C and σ influences the flexibility of the regression and thus the smoothness of the resulting function. During the regression process, increasing the sampling range of the transformation space permits to increase the capture range of the trained similarity. A high capture range is crucial when no good initialization parameters are available. Results presented in Fig. A.3 shows the overall success rate and the final TRE as functions of the initial TRE. While

the success rate of other measures sinks with an increasing initial TRE, our regressed similarity measure shows a good behaviour. This highlights its greater capture range and this, for a better accuracy. In the T1-TOF experiments, KL provided once a better accuracy for an initial TRE of 22.5 mm. This comes from the fact that KL was only successful on three registration experiments: Ψ and MI were actually better than KL in these specific experiments, but in the displayed results their accuracy is averaged on many more experiments as they have much higher sucess rates.

Our method was robust face to different tissue distributions, e.g. coming from patients affected by different kinds of disease or from different locations of the head that were not learned during the regression phase. For example, while slices from the top of the skull contain mostly skin, bone, cerebrospinal fluid, grey and white matter, slices in the middle of the head also consists of muscles and eyes. This could suggest that the manifold on which the similarity was learned is not strictly dependent on the anatomy. This needs however to be extensively studied with further experimentations.

A.1.4 Discussion and Conclusion

In this work, we propose to take advantage of prior information, namely a registered pair of images, in order to improve results in multi-modal registration. Our contribution is to define, with a regression approach, a new similarity measure relating the manifold described by joint histograms of two different modalities to the registration error. Experiments conducted on MR-T1/MR-TOF and MR-T1/SPECT images show that the presented method is a promising alternative for multi-modal registration. We empirically demonstrated that these manifolds are not dependant on the choice of the particular training pair within the dataset. This means that such an adapted application-specific measure can be defined by using a single pair of manually registered images from the specific application. Moreover, its robustness to different or new tissue distributions suggests that such manifolds could be modality-specific. In future work, we will further study their dependence to the variations of tissue distribution within the images.

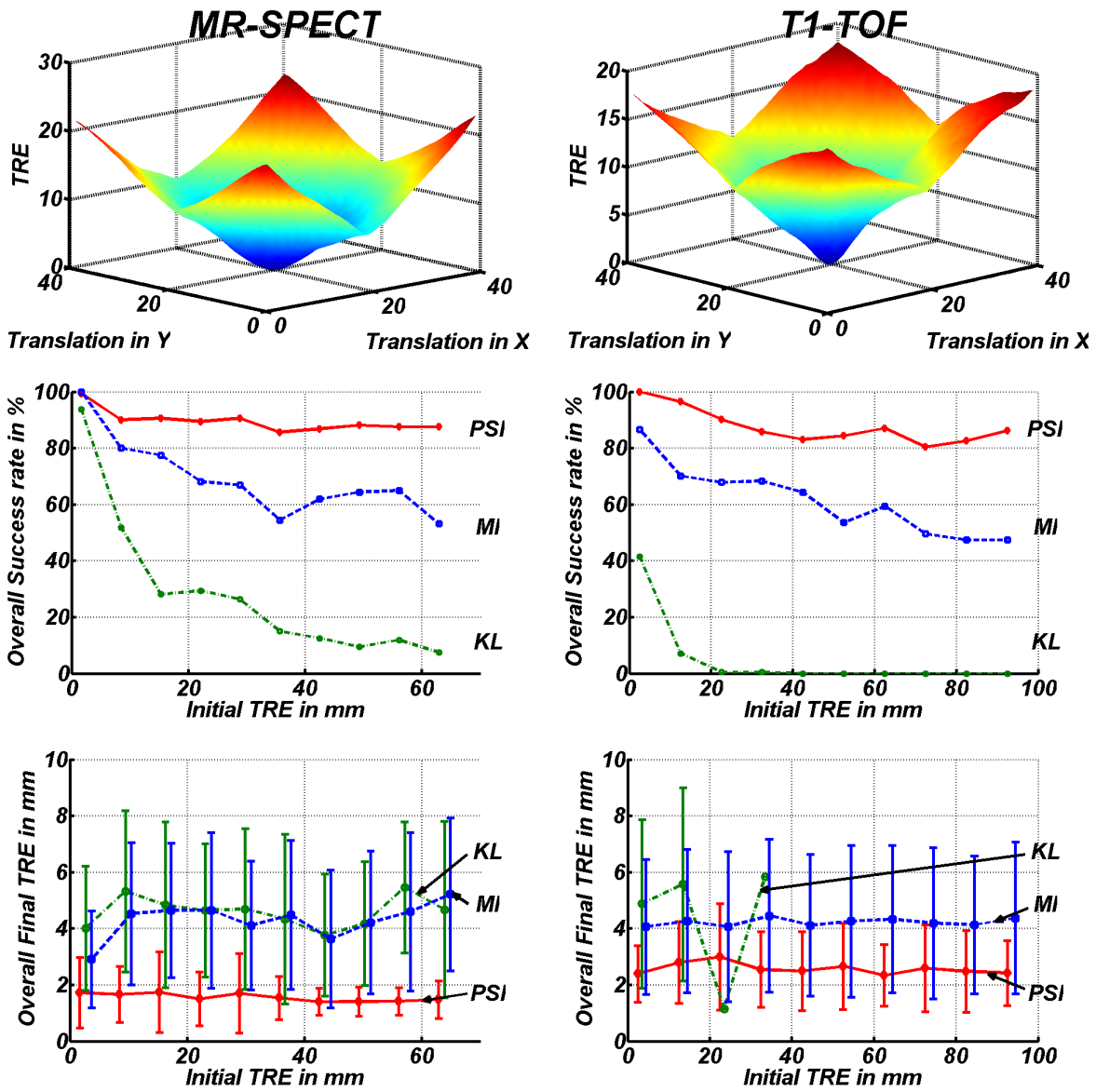

Figure A.3: Registration experiments: On top, plot of the similarity Ψ for variations in translation in x and y between -20 and $+20$ pixels - in the middle, plot of the success rate (in percent) and at the bottom final TRE (mean and standard deviation in mm) according to an increasing initial TRE. Left MR-SPECT, right T1-TOF

A.2 Similarity Learning for Guide-wire Tracking in Fluoroscopic Sequences

Deformable guide-wire tracking in fluoroscopic sequences is a challenging task due to the low signal to noise ratio of the images and the apparent complex motion of the object of interest. Common tracking methods are based on data terms that do not differentiate well between medical tools and anatomic background such as ribs and vertebrae. A data term learned directly from fluoroscopic sequences would be more adapted to the image characteristics and could help to improve tracking. In this work, our contribution is to learn the relationship between features extracted from the original image and the tracking error. By randomly deforming a guide-wire model around its ground truth position in one *single* reference frame, we explore the space spanned by these features. Therefore, a guide-wire motion distribution model is learned to reduce the intrisic dimensionality of this feature space. Random deformations and the corresponding features can be then automatically generated. In a regression approach, the function mapping this space to the tracking error is learned. The resulting data term is integrated into a tracking framework based on a second-order MAP-MRF formulation which is optimized by QPBO moves yielding high-quality tracking results. Experiments conducted on two fluoroscopic sequences show that our approach is a promising alternative for deformable tracking of guide-wires.

A.2.1 Introduction

During the last decade, the success of angiographic interventions relied on the ability of physicians to navigate in the patient's anatomy based only on their mental three-dimensional representation of the human body as well as on the haptic feedback from the instruments. Recent advances in computer aided planning and navigation techniques offer great potential of minimizing the risk of complications and improving the precision. In the case of angiographic applications, the most common imaging modality is X-ray fluoroscopy A.4. Currently, in order to monitor guidance procedures, a roadmap, e.g. a digital subtracted angiography (DSA) showing vessel anatomy, is computed during the intervention. Unfortunately, such roadmaps cannot directly be fused with the intra-operative fluoroscopic sequence due to misalignment caused by respiratory motion. A fundamental step toward a successful integration of any navigation application into clinical routine is the estimation and compensation of such respiratory motion. Determining this spatio-temporal information is a challenging task due to the fact that fluoroscopic X-ray images have a low signal to noise ratio, are subject to big changes in contrast and suffer from background clutter in the abdominal area. Moreover, the apparent motion of the guide-wire is a combination of multiple components. The major motion in the chest is caused by patient breathing. A second, deformable component results from forces applied to the guide-wire by the physician and by surrounding organs which are subject to non-uniform motions during the breathing cycle. Furthermore, the guide-wire may sometimes partially vanish.

A recent approach dealing with the problem of guide-wire tracking in fluoroscopy is [40]. In this work, Heibel et al. proposed a scheme for deformable tracking based on a

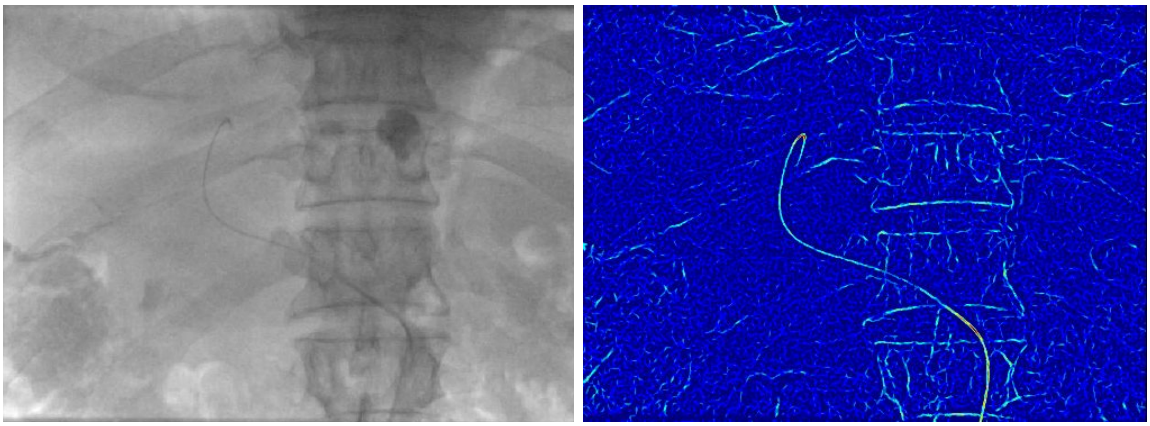

Figure A.4: Fluoroscopic X-ray: Tracking the guide-wire is challenging task in these images having a low signal to noise ratio and suffering from background clutter in the abdominal area.

MAP-MRF formulation. However, their data term does not differentiate well between medical tools and anatomic background such as ribs and vertebrae. A learned data term being more robust and adapted to the image characteristics of fluoroscopic sequences could help to further improve the tracking. Since MRF formulations are derivative free optimization procedures, they ease the integration of such learning based energies for which analytical derivatives are hard to derive if possible at all. Learning permits to model complex relationships between the information contained in the images and the quality of alignment. In the context of guide-wire tracking, we can distinguish two kinds of learning approaches: First, methods for the detection of the guide-wire in each frame and second, methods used for learning a data driven energy. A learning-based tracking approach by detection based on marginal space learning was presented by Barbu *et al.* in [7]. Later, Wang *et al.* proposed in [104] the combination of learning-based detectors and online appearance models. In the case of energy learning, Nguyen et al. [66] addressed the problem of modeling the error surface of parametric appearance models in order to minimize the number of local minima for image alignment and recently Pauly et al. suggested in [77] to learn the statistical relationship between two different imaging modalities to model a data term for multi-modal rigid registration.

In this work, our contribution is a learning approach for deformable tracking: we propose to learn a data term based on the relationship between features extracted from the original image and the tracking error. As illustrated in Fig.A.5, we introduce novel features, namely the local mean orthogonal intensity profiles that represent information contained in the original image. Since deformable transformations have a high number of degrees of freedom, the intrisic dimensionality of the space spanned by these features is high. However, typical guide-wire deformations are lying on a subspace we propose to learn to reduce the complexity of our problem. A set of random deformations is then generated automatically and applied to the ground truth position of the guide-wire on a single reference image. A training set of data points from the corresponding local mean orthogonal profiles and their associated tracking error values is thereby created. Learning is then performed on this dataset with a support vector regression. The resulting data

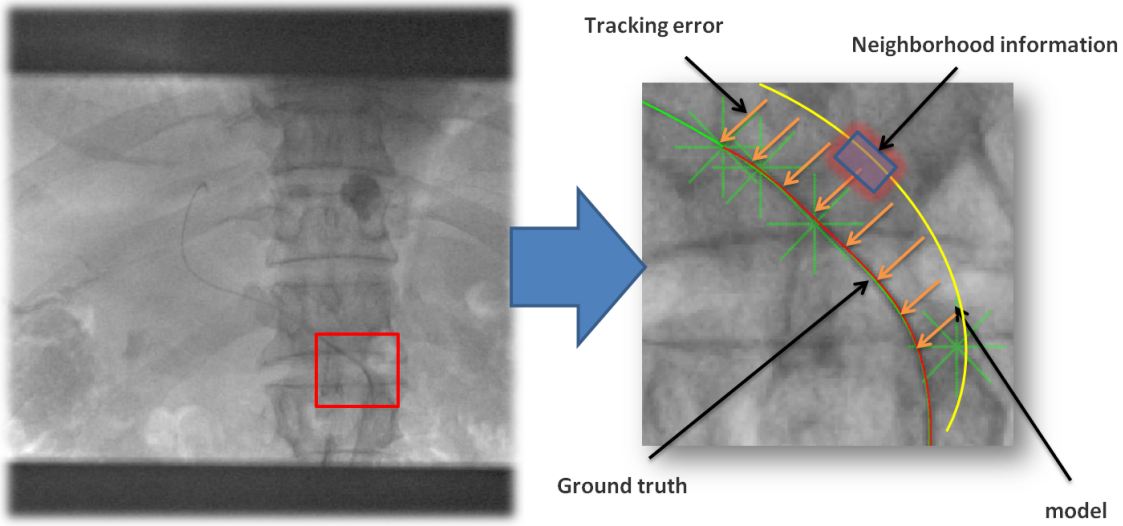

Figure A.5: Similarity learning: robustness of tracking can be improved by learning a data term directly from fluoroscopic images

term is integrated into a tracking framework based on a MAP-MRF formulation which is solved with higher-order clique reduction techniques. Due to the higher-order nature of our problem and since we are dealing with non-submodular energy functions we chose a combination of the recently proposed reduction scheme of Ishikawa [45] and the QPBO [39] optimizer supporting improvements in order to deal with unlabeled nodes [85]. The remainder of the chapter is organized as follows: Section A.2.2 presents our regression approach to define an optimal data term for guide-wire tracking. Section A.2.3 reports experiments performed on two fluoroscopic sequences. Results show that our approach presents a promising alternative for guide-wire tracking in fluoroscopic sequences. Section A.2.4 concludes the paper and gives an outlook on future work.

A.2.2 Methods

A.2.2.1 Problem statement

The goal of tracking is to identify the relative motion of an object in a series of consecutive frames. In most tracking algorithms, we can distinguish two phases: first, the detection of the object of interest in the initial frame followed by the actual tracking in each new frame given previous positions. In this paper, we focus on the problem of tracking a guide-wire through a fluoroscopic sequence knowing its initial position. Let us denote $\mathcal{C}$ our guide-wire model and $\{I_t\}_{t\in\{0,\dots,T\}}$ the set of consecutive images in which we want to track the guide-wire. In fluoroscopic images, guide-wires appear as curvilinear structures which can be represented as B-spline curves. The advantage of such a representation is its low-dimensionality, its implicit smoothness and its local support of control points. Our guide-wire model $\mathcal{C}$ is defined as the following linear combination of control points:

$$\mathcal{C}(s) = \sum_{i=1}^{M} N_i(s) P_i \quad \text{where } s \in [0,1] \tag{A.14}$$

where N_i denote the basis functions and P_i the positions of M control points. By using this model, we want to estimate the optimal curve parameters, i.e. the best configuration of the control points, to match the visible structures in an image, and this, knowing its previous position. The tracking problem can be then formulated as a maximum a posteriori estimation:

$$\mathcal{C}_t^* = \mathbf{argmax}_{\mathcal{C}_t} P(I_t|\mathcal{C}_t) P(\mathcal{C}_t) \tag{A.15}$$

where $\mathcal{C}_t^*$ is the best curve estimate at instant t. $P(I_t|\mathcal{C}_t)$ is the likelihood of observing the data knowing the model and $P(\mathcal{C}_t)$ the prior or probability of the current curve configuration. Let us assume the likelihood to follow a Gaussian distribution and the prior a Gibbs' distribution, we can then reformulate Eq.A.15 as an energy minimization:

$$\mathcal{C}_t^* = \mathbf{argmin}_{\mathcal{C}_t} \left(E_{data}(I_t|\mathcal{C}_t) + E_{reg}(\mathcal{C}_t) \right) \tag{A.16}$$

$E_{reg}(\mathcal{C}_t)$ is a regularization term which constraints the space of possible model configurations. Assuming constant length of guide-wire segments in fluoroscopic sequences, we define the regularization term in order to penalize changes in length:

$$E_{reg}(\mathcal{C}_t) = \int_0^1 \left(1 - \frac{\|\mathcal{C}_t'(s)\|}{\|\mathcal{C}_0'(s)\|} \right)^2 ds \tag{A.17}$$

where $\mathcal{C}_t'$ and $\mathcal{C}_0'$ are the first derivatives at instant t and 0 respectively. Thank to the inherent smoothness of a B-spline representation, higher-order terms can be discarded. $E_{data}(I_t|\mathcal{C}_t)$ can be seen as a data term which drives the model according to the current image:

$$E_{data}(I_t|\mathcal{C}_t) = \int_0^1 \Phi(I_t(\mathcal{C}_t(s)))\, ds \tag{A.18}$$

A common choice for Φ is a function which enhances tubular structures similar to the ridgeness measure proposed by Frangi et al. [30]. Such measures can be tuned to emphasize only structures of the scale of the guide-wire and to remove outliers such as ribs or vertebrae. However, since the data term is only evaluated along the current position of the curve, the main drawback is a very low capture range and a lack of robustness in terms of outliers or partial occlusions.

Instead of relying on the feature image intensities along the curve profile, we propose to extract features from the unprocessed image orthogonally to the curve, namely *local mean orthogonal intensity profiles*. We can then model a data term by learning a function Ψ relating the space $\mathcal{M}$ spanned by these features and the tracking error. By using a single fluoroscopic image and a set of local displacements around the ground truth position of our guide-wire, we can sample the space $\mathcal{M}$ by extracting the local mean orthogonal intensity profiles associated to each displaced curve. Each of these "points" of $\mathcal{M}$ is then associated to a tracking error derived from the corresponding curve parameters, hereby

generating a set of data points. Finally Ψ is modeled by performing a regression on these points. The following section presents how to extract the mentioned features.

A.2.2.2 Local Mean Orthogonal Profiles

In a fluoroscopic image, a human being may recognize the guide-wire because of its curvilinear aspect and its darker intensities compared to its environment. For this reason, a common method would be to enhance this structure and to keep track of it along the sequence by using a data term based on the intensity profile along the curve. Unfortunately, in the case of larger displacements between two consecutive frames, it is hard to relocate the guide-wire in a heterogeneous region containing outliers without any information about the search direction. Indeed, such data terms suffer from an extremely narrow valley around the global extremum. To overcome this problem and benefit from an increased capture range, we propose features which describe the intensity profiles *orthogonally* to the curve. First, we subdivide our curve $\mathcal{C}_t$ into n segments $\left\{S_t^k\right\}_{k\in\{1,\dots,n\}}$. Each segment S_t^k is a spline we characterize by the following descriptor $\mathcal{J}_t^k$:

$$\mathcal{J}_t^k = \frac{1}{q}\sum_{j=1}^{q}\Lambda_t^{k,j}, \tag{A.19}$$

with q being the number of sample points along this segment. $\Lambda_t^{k,j}$ is an orthogonal intensity profile whose r^{th} element is defined as:

$$\Lambda_t^{k,j}(r) \;=\; I_t\left(S_t^k(u) + r\cdot\mathbf{n}(u)\right) \tag{A.20}$$

where $\mathbf{n}(u)$ is the normal vector at point $u = (j-1)/(q-1)$ and $r \in \{-R,\dots,R\}$. The dimensionality of this vector is $2R+1$ which corresponds to the length of the profile centered on the segment. Note that since only the profile's shape is of interest, each profile $\Lambda_t^{k,j}$ is normalized between 0 and 1. Taking the mean over the segment provides a feature vector which is more robust to noise and outliers. Each curve $\mathcal{C}_t$ is then described by the following set $\left\{\mathcal{J}_t^k\right\}_{k\in\{1,\dots,n\}}$.

A.2.2.3 Data points generation by motion learning

The goal of our approach is to learn a function Ψ relating the local mean orthogonal profiles and the tracking error:

$$\Psi : \mathcal{M} \to \mathbb{R}, \tag{A.21}$$

with good characteristics for tracking purposes, namely convexity and smoothness. Therefore, the space $\mathcal{M}$ spanned by these features needs to be sampled thoroughly as a function of the relative displacement. Since the guide-wire is a deformable structure, the intrisic dimensionality of our features according to free deformations would be high and thus, hard to sample. However, in a real fluoroscopic sequence, a guide-wire is not subject

to free deformations. Indeed, main displacements are due to breathing motions and additional small deformations. This means that in reality, our features do not describe the full space $\mathcal{M}$ but lie on a lower dimensional subspace. To reduce the complexity of our problem, we propose to learn the deformation probability distribution from a real sequence. Thus, random displacements can be automatically generated to build our training dataset.

Learning guide-wire Motions: During a sequence, each segment S_t^k of our curve $\mathcal{C}_t$ is subject to a series of consecutive displacements we denote $\left\{D_t^k\right\}_{t\in\{0,\dots,T-1\}}$. Each D_t^k is modeled by a vector containing the displacements of sample points of the segment between 2 consecutive frames. Its j^{th} element is defined as:

$$D_t^k(j) = S_{t+1}^k(u) - S_t^k(u) \tag{A.22}$$

These vectors are collected for all segments along the whole sequence and grouped in a training set $\mathcal{D} = \left\{D_t^k\right\}_{t\in\{0,\dots,T-1\}}^{k\in\{1,\dots,n\}}$. To learn the underlying probability distribution of these displacements, we propose to model it with a gaussian mixture model $\mathcal{G}$. The parameters of $\mathcal{G}$ can be estimated by using Expectation-Maximization. Once we have learned our gaussian mixture model, we can generate random segment displacements $\{D_i\}_{i\in\{1,\dots,Q\}}$ from this probability distribution.

Data points generation: As shown on Fig.A.6, by using a reference fluoroscopic image, e.g. the first frame of the sequence, we can generate local mean orthogonal profiles $\{\mathcal{J}_i\}_{i\in\{1,\dots,Q\}}$ by perturbating the segments of the ground truth curve with the randomly generated displacements $\{D_i\}_{i\in\{1,\dots,Q\}}$. The corresponding tracking error $\mathcal{E}_i$ associated to each $\mathcal{J}_i$ is computed as follows:

$$\mathcal{E}_i = \|D_i\|^2 \tag{A.23}$$

This procedure permits us to generate the set of pairs $\{(\mathcal{J}_i, \mathcal{E}_i)\}_{1,\dots,Q}$, on which the regression will be performed to learn our function Ψ.

A.2.2.4 Learning data term through support vector regression

From previously generated data points, the function Ψ can be learned through non-parametric support vector regression. Let us consider the problem of fitting a function on the set of Q data points $\{(\mathcal{J}_i, \mathcal{E}_i)\}_{i\in\{1,\dots,Q\}}$. Ψ is modeled as the following function:

$$\Psi(\mathcal{J}) = \langle w\ ,\ \mathcal{J}\rangle + b, \tag{A.24}$$

where w is a weighting vector of dimensionality $\mathbf{dim}(\mathcal{M})$ and b a bias. This can be written as a convex optimization problem [91]:

$$\begin{aligned} &\text{minimize}\ \ \frac{1}{2}\|w\|^2 \\ &\text{subject to:}\ \begin{cases} \mathcal{E}_i - \langle w\ ,\ \mathcal{J}_i\rangle - b \le \epsilon \\ \langle w\ ,\ \mathcal{J}_i\rangle + b - \mathcal{E}_i \le \epsilon \end{cases} \end{aligned} \tag{A.25}$$

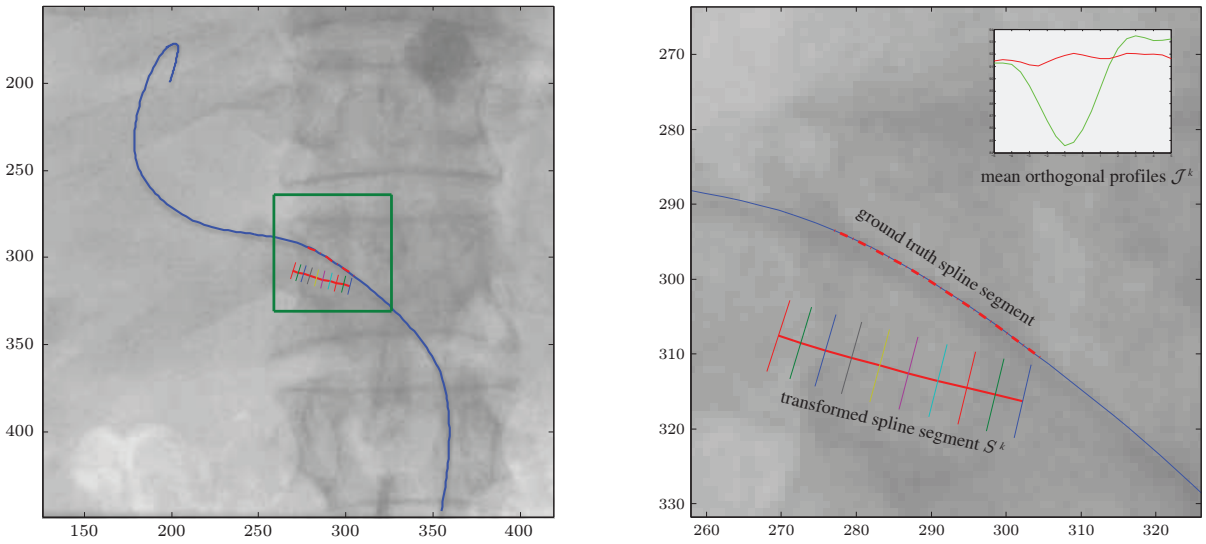

Figure A.6: Data term learning: by perturbing the ground-truth curve from a single frame with random displacements, we can build a training set of local mean orthogonal profiles with their associated tracking errors.

This aims at minimizing the norm of w to penalize the model complexity and the regression errors on the data points with a regression tolerance denoted by ϵ. Equation (A.25) corresponds to minimizing the following functional:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|w\|^2 + \mathrm{C}\sum_{i=1}^{Q}\left(\xi_i^+ + \xi_i^-\right) \\ &\text{subject to:} \begin{cases} \mathcal{E}_i - \langle w\ ,\ \mathcal{J}_i\rangle - b \leq \epsilon + \xi_i^+ \\ \langle w\ ,\ \mathcal{J}_i\rangle + b - \mathcal{E}_i \leq \epsilon + \xi_i^- \\ \xi_i^+, \xi_i^- \geq 0 \end{cases} \end{aligned} \tag{A.26}$$

where C weights the impact of the errors and thus the flexibility of the model. According to the Representer theorem, a solution w_{opt} of this minimization is always a linear combination of the training vectors in $\mathcal{M}$ with weights $\{\alpha_i\}_{i\in\{1,\ldots,Q\}}$:

$$w_{opt} = \sum_{i=1}^{Q} \alpha_i\ \mathcal{J}_i \tag{A.27}$$

which leads to the following model:

$$\Psi(\mathcal{J}) = \sum_{i=1}^{Q} \alpha_i\ \langle \mathcal{J}_i\ ,\ \mathcal{J}\rangle + b \tag{A.28}$$

Finally, the global data term computed on all segments can be written as:

$$E_{data}^{learn}(\mathcal{C}_t) = \frac{1}{n}\sum_{k=1}^{n} \Psi(\mathcal{J}_t^k) \tag{A.29}$$

A.2.3 Experiments and Results

In the following experiments, we show the successful application of our machine learning approach for the tracking of guide-wires in fluoroscopic images. The two sequences we used for our experiments were accquired during liver chemoembolizations. In this procedure, a guide-wire is inserted into the femoral artery and threaded into the aorta. The catheter is then advanced into the hepatic artery. Once the branches that feed the liver cancer are reached, the chemotherapy is infused. In both sequences, the catheter is already inserted in the artery and we aim at recovering from breathing motions.

Motion learning: A set of inter-frame segment displacements is computed from a reference sequence where the guide-wire positions were manually annotated. A gaussian mixture model is then fitted to this dataset by using EM algorithm. The analysis of Bayes' Information Criterion leads to the choice of two gaussian components.

Data term learning: A quadratic B-spline is fit to each hand-labeled point set by minimizing discontinuities in the second derivative [25]. Given the previously learned gaussian mixture model, a set of $Q = 3000$ random segment displacements is automatically generated. By perturbating the ground-truth curve from a single frame with these random displacements, we can build a training set of 3000 local mean orthogonal profiles with their associated tracking errors. Note that the choice of Q is a compromise between complexity and accurate modeling of the data term. During the experiments profiles with different radii are evaluated. Finally, the data term is learned by performing a support vector regression.

Tracking experiments: Experiments are conducted on two clinical sequences of 142 and 228 frames with a resolution of 512×512 pixels and respective pixel spacings of 0.432×0.432 mm and 0.308×0.308 mm. In order to evaluate the tracking results, guide-wires are manually annotated in each frame. The following distance measure has been used throughout all experiments to assess the quantitative tracking quality:

$$\chi = \frac{1}{2}\left(\frac{1}{|\mathcal{C}_t|}\sum_{x_i \in \mathcal{C}_t} \min_{y \in \mathcal{C}_{\mathrm{GT}}} d(x_i, y)^2 + \frac{1}{|\mathcal{C}_{\mathrm{GT}}|}\sum_{y_j \in \mathcal{C}_{\mathrm{GT}}} \min_{x \in \mathcal{C}_t} d(x, y_j)^2\right). \tag{A.30}$$

Here $\mathcal{C}_{\mathrm{GT}}$ is the manually annotated curve and $\mathcal{C}_t$ the tracking result of an individual frame.

Results: Tab.A.1 shows mean errors on whole sequences where the data term is trained on the first frame of one sequence, and tested in tracking in both sequences. Submillimeter yet subpixel tracking accuracy can be achieved with our learned data-term and this, for a frame rate of 1.5 frame/s on a 3 Ghz duo core. Moreover, cross-validation illustrates the robustness of our approach even if it has been trained on another sequence showing different contrasts, motions and background. Note that since the Seq.1 presents motions of higher amplitude, its mean error is slightly bigger than for the other sequence. The great

Tracking Results								
Trained on	Seq.1, Frame 1				Seq. 2, Frame 1			
Tested on	Seq.1		Seq.2		Seq.1		Seq.2	
Profile Radius	5 pixels	10 pixels	5 pixels	10 pixels	5 pixels	10 pixels	5 pixels	10 pixels
χ mean (mm^2)	0.7115	0.5249	0.1636	0.1622	0.6632	0.5815	0.1796	0.1700
χ std dev (mm^2)	0.4289	0.2715	0.1633	0.1185	0.6184	0.3366	0.1771	0.1645

Table A.1: Tracking experiments in real fluoroscopic sequences: training performed on initial frame and tested on the following frames.

advantage is the ability to model the convexity and smoothness of this term. Indeed, its convexity properties can be designed by replacing the tracking error function A.23. The choice of hyper-parameter C from equation (A.26) influences the flexibility of the regression and thus the smoothness of the resulting function.

A.2.4 Discussion and Conclusion

In this work, our contribution was to learn the relationship between features extracted from an unprocessed image and the tracking error in order to model a data term. Experiments conducted on two fluoroscopic sequences show that our approach is a promising alternative for deformable guide-wire tracking. Indeed, our method is robust to changes in contrast, background clutter and partial occlusions of the guide-wire during the sequence, and this, even if training was performed on another dataset. Since the feature space under free deformations is high-dimensional, we proposed to model the distribution of the reduced space of typical guide-wire motions with a gaussian mixture model. In turn, this permitted us to automatically generate random guide-wire deformations from this distribution for the sake of regression. Going further, the space of relative motions between consecutive frames could be constrained during tracking to expected guide-wire motions. In future work, we will explore the possibility of deriving an adapted regularization term from this motion distribution model.

APPENDIX

B

WAVELET ENERGY MAP, A ROBUST SUPPORT FOR MULTI-MODAL REGISTRATION OF MEDICAL IMAGES

Multi-modal registration is the task of aligning images from an object acquired with different imaging systems, sensors or parameters. The current gold standard for medical images is the maximization of mutual information by computing the joint intensity distribution. However intensities are highly sensitive to various kinds of noise and denoising is a very challenging task often involving a-priori knowledge and parameter tuning. In this chapter, we report our work published in [78] on a novel robust information support for multi-modal registration: the *wavelet energy map*, giving a measure of local energy for each pixel. This spatial feature is derived from local spectral components computed with a redundant wavelet transform. The multi-frequential aspect of our method is particularly adapted to robust registration of images showing ambiguities such as tissues, complex textures and multiple interfaces. We show the benefits of the wavelet energy map approach in comparison to the classical framework in 2D and 3D rigid registration experiments on synthetic and real data.

B.1 Introduction

Image registration is a crucial preprocessing step in all image analysis tasks in which information from various imaging sources needs to be combined. These sources of information can be acquisitions from different viewpoints of an object, at different times or with different sensors [114]. Establishing the correspondences between images acquired with different medical imaging modalities is a challenging task known as multi-modal registration. To identify the geometric transformation that maps the coordinate system of one modality to the other [115], objective functions that evaluate the quality of alignment known as similarity measures are optimized. The choice of the appropriate measure is not straightforward, because it implicitly models the relationship between the different images to register. Indeed, this measure quantifies how well images are registered according to the transformation parameters [84]. As modeling the physical relationship between different imaging modalities is very difficult, statistical measures have become more and

more popular.

Since its introduction by Viola and Wells [101] and Collignon et al [18], mutual information remains the state of the art of multi-modal registration of medical images. Several other entropy-based measures have also been introduced: for example, the normalized version of mutual information proposed by Studholme et al. [93] or the Kullback-Leibler distance introduced by Chung et al. [17]. Furthermore, a quantitative-qualitative measure of mutual information has been presented by Luan et al. [57] to take the saliency of each image voxel into account. All these different entropy definitions use the same support of information: the intensity distribution. But image intensities are very inclined to be corrupted by noise due to different phenomena that can occur during the acquisition procedure. Indeed, medical images such as magnetic resonance suffer very often from different types of noise due to interferences between electronic devices, which can dramatically influence registration results. Image denoising is however a very challenging task, because the type of noise has to be known or modeled to perform an efficient filtering.

To gain in robustness, Gan and Chung [33] introduced a novel spatial feature named maximum distance-gradient-magnitude (MDGM) for rigid registration of medical images. Each pixel is characterized by the most dominant local variation and its intensity value. Again, taking into account intensity values can lead to misregistration in presence of noise.

Because of its ability to extract features characterizing local frequency components, the Discrete Wavelet Transform (DWT), whose main application is data compression, has been recently introduced in the field of image registration. Le Moigne et al. [63] and Sharman et al. [88] proposed to perform the registration on a feature space formed by the dominant local variations. In a coarse to fine strategy, wavelet coefficients are selected with a magnitude above a certain threshold. This selection method is also used by Hongli et al. [44] on approximation coefficients computed with a slightly different wavelet transform scheme. Using a Complex Wavelet Transform (CWT), Oubel et al. [70] present a 2 steps registration framework, in which a first alignment is done on low frequency and refinement on high frequency coefficients from the first decomposition level. But relying on these high frequency components is not a safe strategy, especially in the case of high frequency noise. Li et al. [56] propose an energy feature based on the coefficients of the first decomposition level computed with a Discrete Frame Wavelet Transform. Each pixel being only characterized by the highest part of the frequency spectrum, again, these features are not reliable in the presence of noise.

To take fully advantage of this kind of transforms, we combine the information contained in *all* sub-bands of the frequency spectrum. In this paper, we propose to perform the registration on a novel feature map we name *wavelet energy map* (WEM), whose computation is parameter-free and which is very robust to the noise present in the original images. The WEM measures the local signal energy at each pixel and is computed from local spectral components in its neighborhood. These spectral components are obtained with the redundant wavelet transform [22], which gives the best approximation of a space-frequency representation. For registration tasks, the energy probability distribution of the WEM is used as input for mutual information. The method does not require any additional a-priori information or parameter, but only a slightly increased initial computation. The multi-frequential aspect of this approach is especially adapted to registration

of medical images presenting ambiguities such as tissues with complex textures or multiple interfaces. We demonstrate its value on a wide range of experiments on synthetic and real images.

In the remaining of this chapter, section B.2 will define the wavelet energy map and the registration framework. Section B.3 will present experiments demonstrating three properties of the WEM:

1. ***correctness***: energy and intensity maps give the same global maximum for mutual information
2. ***robustness***: registration on local energies is robust to noise
3. ***efficiency***: mutual information computed on wavelet energy maps outperforms the classical approach in terms of robustness for an equivalent accuracy

B.2 Methods

B.2.1 Problem statement

The goal of multi-modal image registration is to identify the geometric transformation that maps the coordinate system of one modality to the other. Let us consider two 2D images defined on the domains Ω_1 and Ω_2 with intensity functions $I_1 : \Omega_1 \subset \mathbb{R}^2 \to \mathbb{R}$ and $I_2 : \Omega_2 \subset \mathbb{R}^2 \to \mathbb{R}$. The registration task can be defined as a maximization problem, in which we want to estimate the best transform T according to a chosen similarity measure S computed on the discrete overlap domain $\Omega = \Omega_1 \cap T(\Omega_2)$:

$$T = \mathbf{argmax}_T \; S_\Omega(I_1, T(I_2)) \tag{B.1}$$

Since intensities are highly sensitive to noise, we propose to evaluate the similarity in a more robust feature space. We introduce a novel spatial feature map named *wavelet energy map* (WEM) giving a measure of local energy around each pixel of the original images. In the following parts, we define the concept of local energy and its computation from local spectral components extracted with a redundant wavelet transform. We will discuss the 2D case for better readability, the extension to three dimensions being straightforward. In that case, T is the composition of a translation and a rotation.

B.2.2 Energy vs. Intensity

In signal processing, the energy of a signal $x(t)$ is defined as:

$$E = \int_t |x(t)|^2 \tag{B.2}$$

and in the discrete case, for an image with intensity function I:

$$E = \sum_{i,j} |I(i,j)|^2 \tag{B.3}$$

Interpreting a zero image as a flat surface, energy can be understood as the work capacity accumulated during its deformation to the final relief. In the previous equation, energy is expressed in terms of intensities. Because they do not provide any contextual information, intensity values are not a safe support of information. Their variation frequencies, in contrary, offer a safer support by involving spatial context. Parseval's theorem guarantees that in a Hilbert space (we assume intensity functions being elements of a Hilbert space, for example $L^2(\mathbb{R}^2)$) the energy of a signal x can also be determined from its frequency spectrum:

$$\int_t |x(t)|^2 \, dt = \int_f |X(f)|^2 \, df \tag{B.4}$$

with X being the Fourier transform of x. In the discrete case we obtain:

$$\sum_{i,j} |I(i,j)|^2 = \sum_f |F_I(f)|^2 \tag{B.5}$$

where F_I is the 2D Fourier transform of the image. The major drawback of the Fourier transform is its poor resolution in space domain. To define a *local* energy as a spatial feature, we need a frequency-space representation to know which spectral components exist at any given position in the image. A relatively new method introduced by Grossmann and Morlet [38] known as wavelet transform provides the best approximation of this space-frequency representation.

B.2.3 Extraction of local spectral components

B.2.3.1 The redundant wavelet transform

The traditional discrete wavelet transform (DWT) projects a signal onto an orthogonal wavelets basis. Its principle is to extract iteratively the information contained in each sub bands of the frequency spectrum [61]. In practice, the DWT is performed by passing the image through a cascade of orthogonal high pass (H) and low pass (L) filters to select each sub bands and analyze their content. The resulting decomposition coefficients are then down-sampled according to the Nyquist-Shannon sampling theorem as represented on Fig. B.1. The original image is decomposed in 4 components: HH corresponds to the application of high pass filters in x and y directions, HL to high pass in x and low pass in y direction, LH to the contrary and LL to low pass filters in both directions. The LL component is then redecomposed in 4 components and the process is repeated. HH, HL and LH components are called details and LL approximation coefficients.

Orthogonality is a crucial property ensuring the most exact conservation of the spectral information. DWT does not only provide orthogonality between each sub bands, but also between their components. Its major drawback is the down-sampling operation that results in a loss of position information. Hence, we use another transform known as redundant discrete wavelet transform (RDWT) that basically removes the down-sampling operation. The RDWT also known as "Algorithme à trous" produces an over complete representation of the image and is considered as a better approximation of the continuous wavelet transform [22]. It is implemented by using a bank of filters (refer to the filter cascade on Fig. B.2). In the one-dimensional case, the signal is filtered by a low l and a

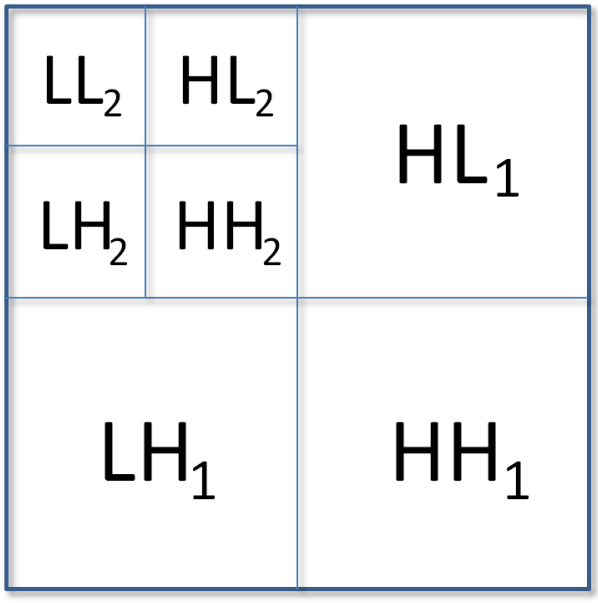

Figure B.1: Discrete Wavelet Transform of a 2D image.

high pass h as shown below:

$$v_{j+1}[n] = \sum_{k=1}^{p} v_j[k] l_j[n - 2^j k] \tag{B.6}$$

$$w_{j+1}[n] = \sum_{k=1}^{p} v_j[k] h_j[n - 2^j k] \tag{B.7}$$

with v_{j+1} being the approximation and w_{j+1} the detail component at the decomposition level $j + 1$ and p the size of the filter. This is analog to a classical filtering of the signal by iteratively inserting zeros, or in other words "holes" ("trous" in french) between all coefficients of the filters.

In the two-dimensional case, each row and column of the original image are treated like a one dimensional signal. By introducing redundant information, the RDWT is not orthogonal but projects the signal onto a frame. A frame can be a stable and redundant representation of signals if its basis verifies the Heisenberg-Weyl condition [24].

The whole frequency axis is then covered by this representation and it can be considered according to Daubechies [24] as a quasi-orthogonal expansion. Such a representation helps to characterize textures of an image and increases the robustness to additive noise [29]. This redundancy has the main advantage to permit a better localization of each spectral components in the image: indeed, to each pixel corresponds a set of coefficients characterizing the local spectrum.

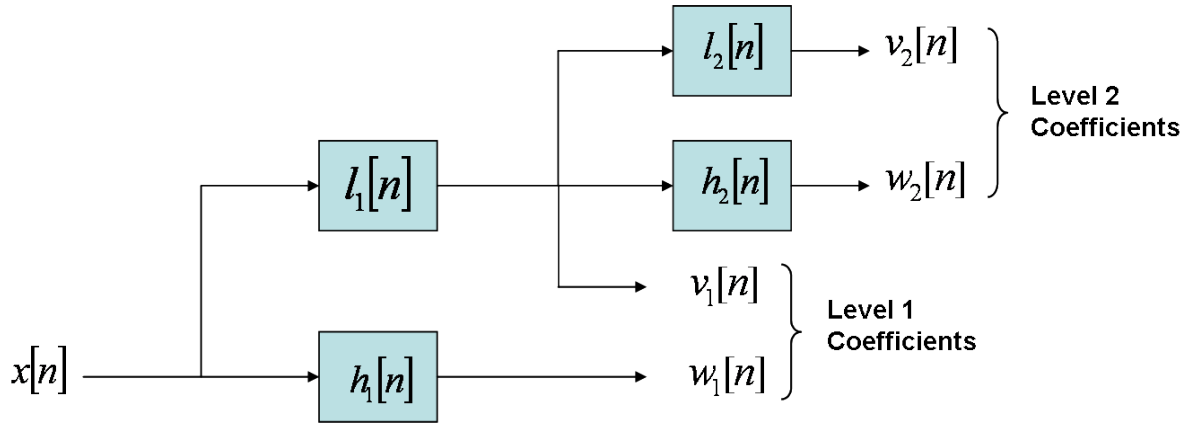

Figure B.2: Redundant Discrete Wavelet Transform: Filter bank for 1D signals.

B.2.3.2 Choice of the wavelet basis

Since the RDWT removes the down-sampling operation, the spatial sampling rate is fixed across all scales. This gives to this transform a translational invariance property in contrary to the traditional DWT. Unfortunately the RDWT is not rotational invariant. To reduce the impact of this rotational non invariance, we focus on two types of wavelets that have a compact support: *orthogonal* wavelets from the Daubechies family and the *biorthogonal* Cohen-Daubechies-Feauveau 9/7 wavelets.

Orthogonal wavelets:
Orthogonality and compact support are important properties to ensure the most exact conservation of information. If we can find a scaling function Φ associated with a multiresolution analysis and orthogonalize its basis, we can then find the associated orthogonal wavelets. But the orthogonalization of the basis associated to Φ has a main drawback: it renders a non-compacted support. Daubechies constructed the only known orthogonal wavelets offering a compact support and a specified number of vanishing moments. Vanishing moments are also a valuable property, in fact they limit the wavelet's ability to represent polynomial behaviour of a signal.

Biorthogonal wavelets:
To gain in flexibility in the construction of wavelets bases, the orthogonality condition can be relaxed. A basis does not have to be orthogonal to offer a stable representation of information. By using its dual basis, it is possible to reconstruct exactly the information. This allows the construction of more families of compactly supported wavelets that can also be symmetric. Symmetry is a nice property in many applications to construct filters with a linear phase of the transfer function. Cohen-Daubechies-Feauveau 9/7 wavelets that are also used in the JPEG2000 compression standard.

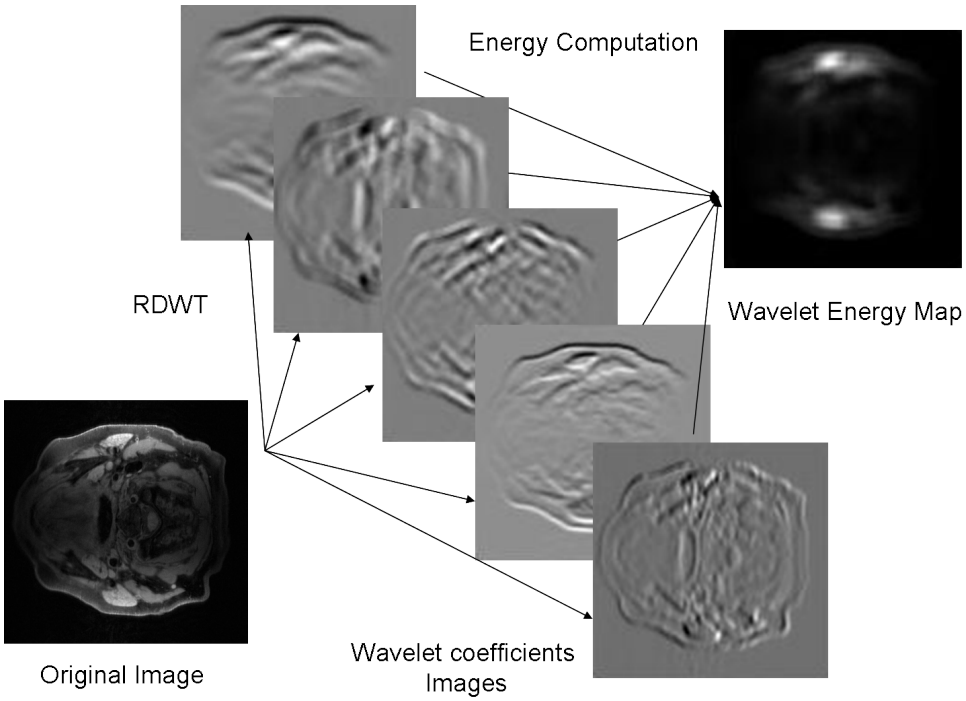

Figure B.3: Method overview: the wavelet energy map computation.

B.2.4 Local energy formulation

After decomposing an image I with a RDWT in m sub-bands with m being the maximum number of possible levels, we obtain for each pixel (i, j) a set of $3m+1$ coefficients (3 detail components per level and the approximation from the last level) providing information on local frequency components. Let $w(i, j)$ be the coefficient vector:

$$w(i,j) = (w_1(i,j), w_2(i,j), \ldots, w_{3m+1}(i,j)) \tag{B.8}$$

Using Parseval's theorem, we can define a local energy $\mathcal{W}(i, j)$ computed from the local spectral components:

$$\mathcal{W}(i,j) = \|w(i,j)\|^2 = \sum_{k=1}^{3m+1} |w_k(i,j)|^2 \tag{B.9}$$

We name *wavelet energy map* (WEM) the array containing all values $\mathcal{W}(i, j)$. Its computation is summarized by Fig. B.3.

B.2.5 Energy based registration framework

In terms of WEMs, equation B.1 becomes:

$$T = \mathbf{argmax}_T \; S_\Omega(\mathcal{W}_1, T(\mathcal{W}_2)) \tag{B.10}$$

with $\mathcal{W}_1$ and $\mathcal{W}_2$ being the WEMs computed from both images to align. Since different imaging systems emphasize different characteristics of an object, the resulting WEMs

will highlight different structures. As presented in the introduction, statistical similarity measures are the current standard in multi-modal registration. Thus, we propose to use the mutual information (MI) to evaluate the statistical relationship between both wavelet energy maps. First, we normalize them between 0 and 1:

$$\overline{\mathcal{W}}(i,j) = \frac{\mathcal{W}(i,j) - \min_{i,j}(\mathcal{W}(i,j))}{\max_{i,j}(\mathcal{W}(i,j))} \tag{B.11}$$

By dividing the domain $[0,1]$ in N bins $\mathcal{B}_1, \ldots, \mathcal{B}_N$, we can determine the probability of a pixel x to fall in the bin $\mathcal{B}_k$:

$$p(x \in \mathcal{B}_k) = \frac{\#\left\{(i,j),\ \overline{\mathcal{W}}(i,j) \in \left[\frac{k}{N}\ \frac{k+1}{N}\right[\right\}}{\#\Omega} \tag{B.12}$$

where $\#\cdot$ is the cardinality operator. It is then possible to compare the shared amount of information in the energy maps of both images by using the classical definition of MI:

$$\begin{aligned} &MI(\overline{\mathcal{W}}_1, T(\overline{\mathcal{W}}_2)) = \\ &H(\overline{\mathcal{W}}_1) + H(T(\overline{\mathcal{W}}_2)) - H(\overline{\mathcal{W}}_1, T(\overline{\mathcal{W}}_2)) \end{aligned} \tag{B.13}$$

with $H(\overline{\mathcal{W}})$ being Shannon's definition of information entropy:

$$H(\overline{\mathcal{W}}) = -\sum_{k=1}^{N} p(x \in \mathcal{B}_k) log_2(p(x \in \mathcal{B}_k)) \tag{B.14}$$

The joint entropy $H(\overline{\mathcal{W}}_1, T(\overline{\mathcal{W}}_2))$ is defined by using the probability of the pixel x to respectively fall into the bins $\mathcal{B}_k$ and $\mathcal{B}_l$ in the maps $\overline{\mathcal{W}}_1$ and $T(\overline{\mathcal{W}}_2)$.

B.3 Experiments and Results

First, experiments on synthetic datasets show that energy and intensity maps give the same global maximum for mutual information. Further tests reveal the robustness of our approach to gaussian noise. Finally, 2D and 3D experiments on real medical datasets illustrate the benefits of a WEM based registration framework. The different wavelets transforms are based on the Matlab implementation by Gabriel Peyré and the Rice wavelet toolbox. To compute a consistent WEM, three conditions have to be fulfilled:

1. Images must have the same pixel size,
2. the domains where the RDWT is computed must have a size which is a factor of 2,
3. both images have to be decomposed in the same number of levels.

Convergence to the right solution depends much more on the topography of the search space offered by the similarity measure than on the optimizer. Hence, we can choose a Downhill-Simplex optimizer, that does not require any gradient information, to solve our

measure maximization tasks. The quality of registration will be evaluated by using the *target registration error* (TRE). The TRE is computed by comparing the positions of a set of points $\{p_i,\ 1 \leq i \leq M\}$ after being mapped by the estimated transform T and by the ground truth transform G:

$$TRE = \frac{1}{M} \sum_{i=1}^{i=M} \|T(p_i) - G(p_i)\| \tag{B.15}$$

In the following, mutual information computed on wavelet energy map will be denoted by MEI (Mutual Energy Information), while the classical approach on intensity maps by MII (Mutual Intensity Information). Different MEI based on Haar, Daubechies 4 (D4) and Cohen-Daubechies-Feauveau 9/7 (CDF) wavelet bases have been evaluated. All experiments were performed with MATLAB 7.5.0 on a Intel Core 2 Duo CPU 2.40 GHz.

B.3.1 Correctness

The goal of these experiments is to show that registration performed with MEI leads to the same global maximum than by using MII. Therefore, we use synthetic images and plot both measures to compare their global maxima and smoothness. For a better understanding and visualization of the results, we analyze separately rotation and translation. The images contain ambiguities resulting in several local maxima to demonstrate the superiority of our approach in terms of smoothness of the search space.

Experiment 1:

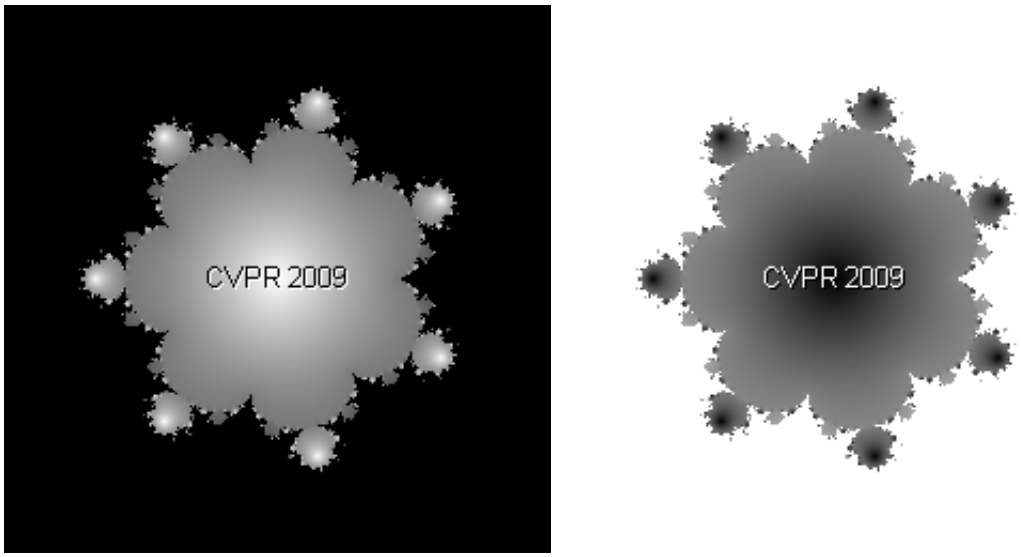

Figure B.4: Experiment 1: the Mandelbrot fractal image and its inverse used for visualizing the similarity measure as function of the rotation parameter.

We use a Mandelbrot fractal with equation $f(z) = z^8 + c$ which has interesting multi-frequential characteristics. Indeed, it shows fine structures at arbitrary small scales, as observed in medical images presenting tissues with complex textures. This fractal shows an orientation ambiguity: we can distinguish 7 different global rotation maxima. Thus a "*CVPR 2009*" detail is added to give an orientation to the whole image (see Fig. B.4). We compare MEI and MII similarity measures for this image and its inverse, both having a resolution of 256x256. Similarity values are plotted for a rotation angle varying between -90 and $+90$ degrees. As shown on Fig. B.5 (left), the global maxima perfectly correspond for both approaches. Besides, the WEM emphasizes the right solution by smoothing other local maxima contrary to the intensity map.

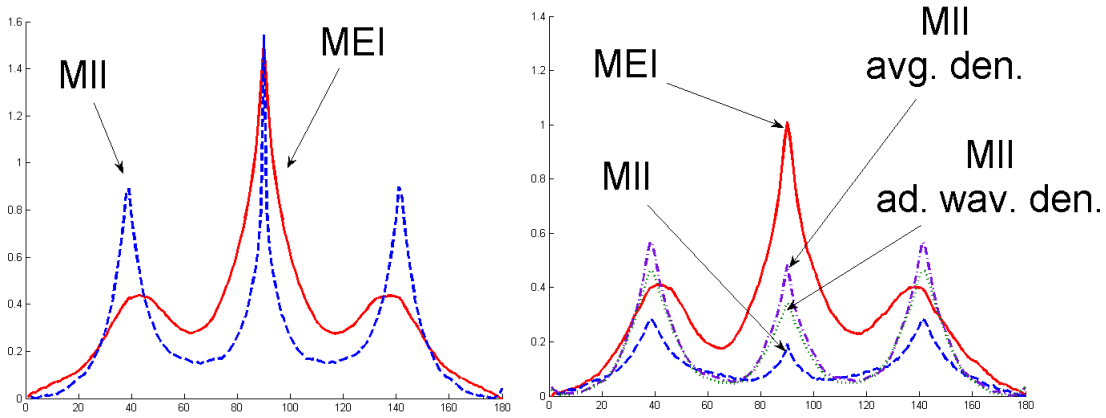

Figure B.5: Experiment 1 and 3 (left) and (right): plot of the similarity measures for rotation angles between -90 and $+90$ degrees. On this figure, D4 wavelet has been used to compute the WEM.

Experiment 2:

In our second experiment, we use the following sum of cosinus to simulate multiple interfaces such as those observed in many medical images: $f(x) = a_1 cos(2\pi f_1 x) + a_2 cos(2\pi f_2 x) + a_3 cos(2\pi f_3 x)$. The resulting image shows ambiguities in both x and y directions. As before, we add the "*CVPR 2009*" detail giving a unique solution (see Fig. B.6). We compare MEI and MII for this image and its inverse, both having a resolution of 128x128. Similarity values are plotted for translation parameters varying between -20 and $+20$ pixels in both directions. As shown on Fig. B.7 the global maximum perfectly corresponds for both approaches. This experiment also reveals the abilities of the WEM to both emphasize the global maximum and offer a smoother search space, which are very valuable for optimization purposes.

B.3.2 Robustness to noise

The goal of these experiments is to argue for the superiority of MEI in terms of robustness in comparison to the classical approach, even when a denoising step has been performed

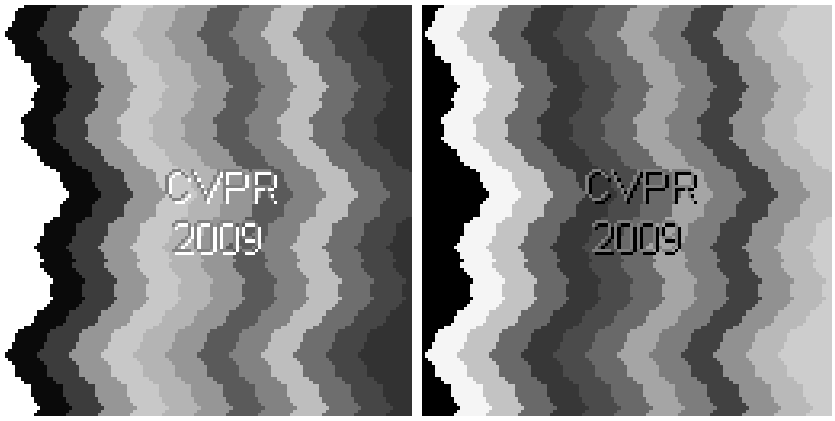

Figure B.6: Experiment 2: the multiple interfaces image and its inverse used for visualizing the similarity measure as function of the translation parameters.

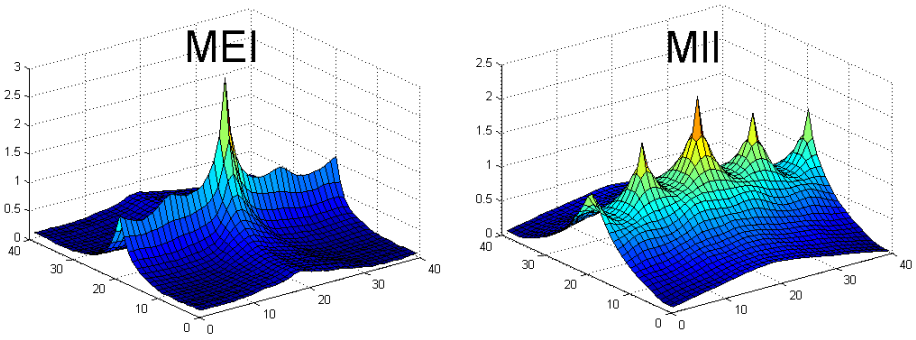

Figure B.7: Experiment 2: Measures plotted for variations in translation. On this figure, D4 wavelet was used to compute the WEM.

prior to the similarity computation. Two denoising methods are used: averaging and adaptive wavelet denoising [14]. The later method is parameter-free and based on soft-thresholding of the wavelet coefficients. It was chosen for fair comparison with wavelet energy maps. Denoising being a challenging task usually involving a priori knowledge on the type of noise, a "denoising-free" measure such as MEI is very valuable for robust registration. We will use for the following experiments a gaussian noise model, that affects independently all pixels of the images and thus highly corrupts the information content. We show in experiment 3 that MEI preserves its search space from distortions on the same synthetic images corrupted by gaussian noise. Experiment 4 shows its robustness to different levels of noise and is performed on an image without any ambiguity. A real registration framework is used for evaluation of the search space formed by the three transform parameters.

Experiment 3

The impact of gaussian noise on the smoothness of the search space of MEI, MII and MII

preceded by a denoising step is analyzed. We use the same figures and setup than in the previous section, and add to all images an additive gaussian noise with $\sigma = 20\%$ of the maximum intensity value.

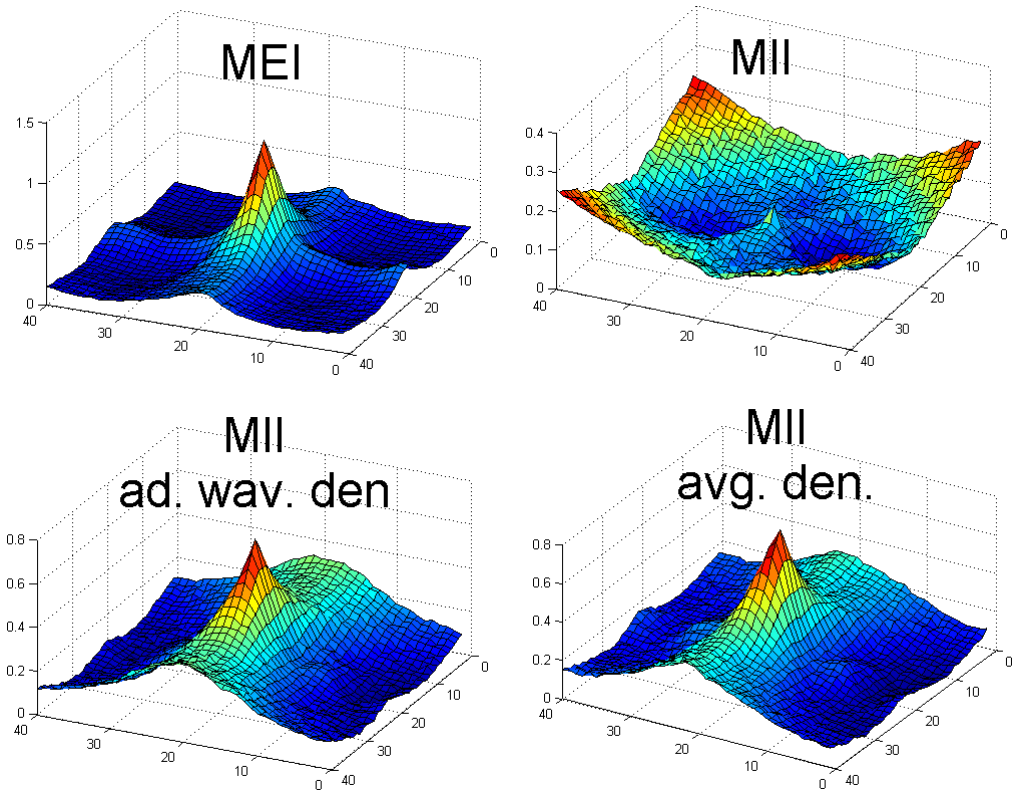

Figure B.8: Experiment 3: From left to right, top to bottom: MEI, MII without denoising, MII with adaptative wavelet denoising and MII with averaging denoising for variation in translation. In this figure, D4 wavelet was used to compute the WEM.

Figures B.5 (right) and B.8 show that our approach preserves the global maximum in presence of gaussian noise unlike the others. Even when a denoising step is applied prior to the computation of the similarity measure, MEI offers a smoother search space with a meaningful global maximum. The rotational case even reveals that MII looses the right solution.

Experiment 4

In this experiment, we evaluate the robustness to an increasing amount of gaussian noise. MEI is compared to MII and MII preceded by a denoising step on an image without any ambiguity. We use the portrait of Lena with a resolution of 128x128 pixels. Registration to its inverse image is performed with a Downhill Simplex optimizer by starting from 10 random initial positions within the range -3 to $+3$ pixels translation and -3 to $+3$ degrees rotation.

Fig. B.9 shows the mean target registration error in function of the percentage of noise. For the classical computation of mutual information, the TRE increases dramatically in

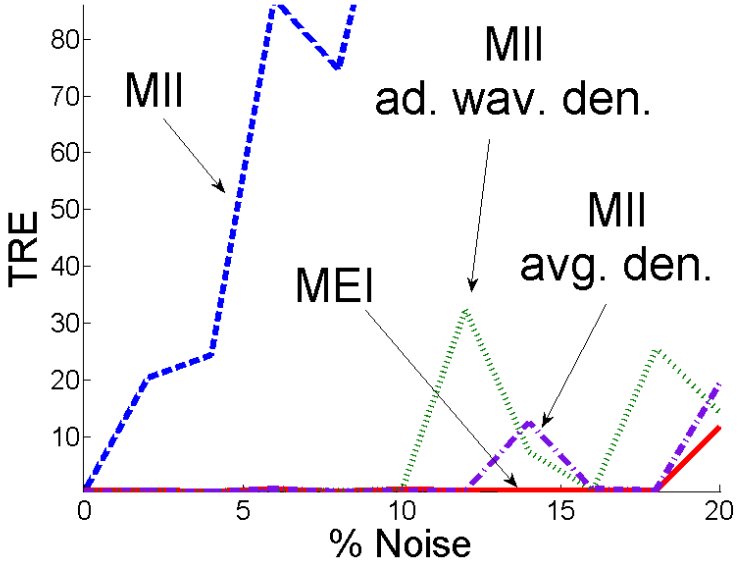

Figure B.9: 2D Registration noise experiment: performed on Lena image and its inverse to evaluate the robusness to noise of each method.

contrast to our approach that always presents the smallest error. Even when a denoising step is performed prior to the registration, results illustrate the benefits of MEI in terms of robustness.

B.3.3 Efficiency on medical images

In this part, we evaluate the efficiency of our approach in 2D and 3D multi-modal registration experiments. First, 2D experiments are conducted on magnetic resonance (MR) images from sequences acquired with different system parameters, namely T1, T2, PD and TOF acquisitions. Then, 3D experiments are performed on MR and SPECT volumes with an increasing amount of noise. MEI computed with Haar, D4 and CDF wavelet bases are compared to the classical MII. To evaluate the registration efficiency of each method, we distinguish between *success rate* and *accuracy*. A registration is considered as successful when the final target registration error is inferior to a given threshold t_e. Otherwise, the approach did not converge in the neighborhood of the right solution. The accuracy is evaluated as the mean target registration error computed on the cases where all methods have converged under t_e.

The chosen multi-modal datasets contain ambiguities which can lead classical approaches to misregistration. Experimental results illustrate the ability of our method to cope with such ambiguities by emphasizing the right global maximum and smoothing other local extrema.

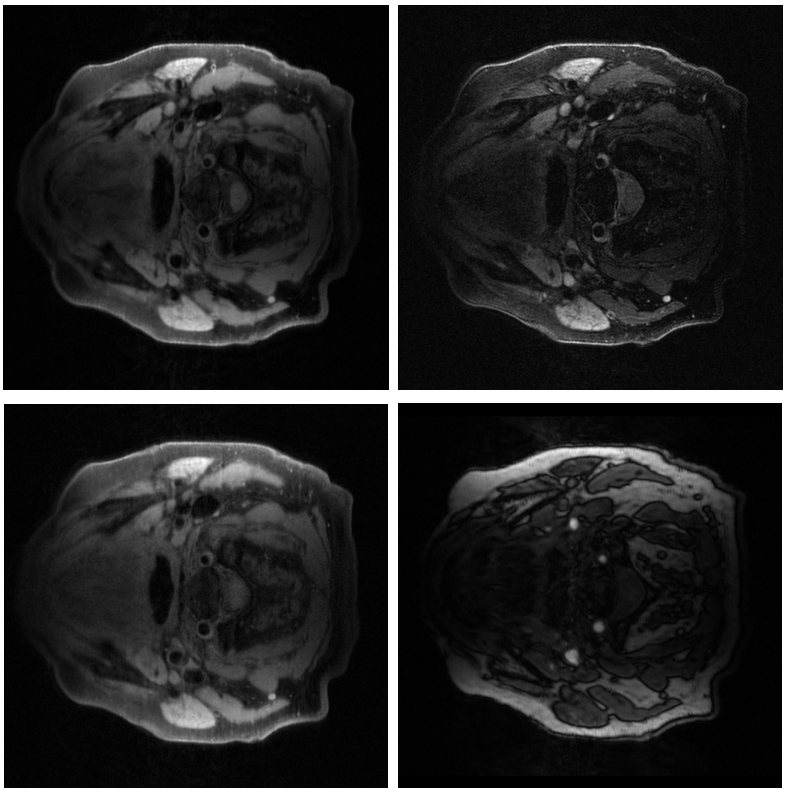

Figure B.10: Experiments on different MR channels: T1, T2, proton density (PD) spin echo sequences and Time of Flight (TOF) MR Angiography gradient echo sequence of the neck of the same patient (from left to right, and top to bottom)

B.3.3.1 2D registration experiments: Real Magnetic Resonance datasets

Between June 2005 and November 2006, volunteers were recruited at a partner hospital at the Neurology department of Klinikum Rechts der Isar in Munich, Germany to perform a study on internal carotid artery stenosis. All patients underwent the same MR imaging protocol: they were imaged using a 1.5T Magnetom Symphony Quantum Gradient scanner from Siemens Healthcare Sector. Four different sequences were acquired: T1, T2, PD and TOF images (refer to Fig. B.10). The MR scans were centered on the carotid bifurcation and patients were positioned on a vacuum pillow. Surface coils receiving the electro-cardiogram gated MR signal for imaging were fixed on the neck to ensure a perfect alignment between the four sequences. These datasets are considered as ground truth for all our experiments.

While T1, T2, PD sequences provide different information related to tissue characteristics, TOF gives dynamic information related to the blood flow in arteries. The circular shape of the neck makes the registration task ambiguous in 2D. Indeed, to recover the right rotation parameter, registration can only rely on corresponding tissues or interfaces appearing in each modalities.

In the following experimental setup, 2D registration tests are conducted on all possible combinations of T1, T2, PD and TOF images taken from 8 patients. They have a reso-

lution of of 128 x 128 with a pixel size of 1.25mm x 1.25mm. Knowing the ground truth position for each dataset, we give a random initial perturbation within the TRE range of 20mm. The threshold t_e is set to 10mm which corresponds to 50% of the initial TRE range. Results presented in tables B.1,B.2 reveal the benefits of our approach on real medical datasets: MEI shows the best success rates for an equivalent accuracy. In hard tasks such as registrations to TOF images, both success rate and accuracy are better. Even though MII is an accurate measure, it shows more local extrema than MEI when images present ambiguities. These local extrema trap the optimizer, leading thereby to more misregistration errors. In contrast, by smoothing these local extrema, our approach offers a better success rate.

Success rate in %				
	Haar MEI	D4 MEI	CDF MEI	MII
T1/TOF	87.50%	87.50%	87.50%	84.38%
T1/T2	100%	100%	100%	98.96%
T1/PD	100%	100%	100%	98.96%
T2/TOF	86.46%	87.50%	87.50%	85.42%
T2/PD	100%	100%	100%	98.96%
PD/TOF	87.50%	87.50%	87.50%	82.29%

Table B.1: 2D registration experiments: **success rate** on T1, T2, PD and TOF images.

Target Registration Error in mm					
		Haar MEI	D4 MEI	CDF MEI	MII
T1/TOF	*mean*	2.84	3.18	2.80	3.75
	std dev	1.69	1.96	0.92	2.16
T1/T2	*mean*	0.91	0.90	0.99	0.64
	std dev	0.64	0.49	0.59	0.55
T1/PD	*mean*	0.95	1.04	1.05	0.83
	std dev	0.55	0.55	0.52	0.58
T2/TOF	*mean*	2.99	3.25	3.19	3.26
	std dev	1.78	2.03	2.02	2.32
T2/PD	*mean*	1.14	1.02	1.25	0.60
	std dev	0.61	0.56	0.73	0.44
PD/TOF	*mean*	2.68	3.03	2.79	3.21
	std dev	1.82	1.91	1.61	2.32

Table B.2: 2D registration experiments: **final TRE** on T1, T2, PD and TOF images.

B.3.4 3D registration experiments: T1 Magnetic Resonance and SPECT-Tc volume

Single photon emission computed tomography (SPECT) is a nuclear medicine tomographic imaging technique used to provide information related to the blood flow. In

3D, SPECT volumes present a blurry cloud aspect with smooth intensity variations that do not correspond to any structure visible in the MR volume. This makes the recovery of transformation parameters for such a task challenging.

In the following experimental setup, 3D registration tests are conducted on MR and SPECT volumes of 4 patients taken from the Whole Brain Atlas online database [46]. They have an in-plane resolution of 128 x 128 with a voxel size of 1.67mm x 1.67mm x 1mm. When the size in z is not a power of 2, a zero-padding is performed at the boundaries of the volume before the RDWT. Knowing the ground truth for each dataset, an initial perturbation is applied within a range of 14mm of initial TRE. By using 20 initializations for each patient, and this for an increasing amount of noise, we can investigate the ability of each measure to converge towards the right solution and thereby assess their robustness. The threshold t_e is set to 7mm which corresponds to 50% of the initial TRE range. Results presented in Fig. B.11 show that MEI offers better success rate for a better accuracy. With an increasing amount of noise, even though an averaging denoising step is performed prior to registration, our approach leads to better results. Since noise is localized in a small part of the frequency spectrum, its impact is minimized by the computation of the WEM. As in the 2D experiments, our method copes better with the ambiguities caused by the aspect of the SPECT volumes.

B.4 Conclusion

In this chapter, we proposed to perform the registration on a feature map called *wavelet energy map* (WEM) instead of using the original image. We empirically showed that mutual information performed on the WEM leads to the same solution than the classical approach on intensity maps. Moreover, its multi-frequential aspect permits to emphasize the global maximum in ambiguous cases containing multiple local extrema, offering thereby a smoother search space, even in presence of noise. 2D and 3D registration experiments on real medical datasets illustrated the efficiency of our approach in comparison to the classical framework which is more sensitive to noise and image ambiguities. In future work, we plan to address the rotational non-invariance issue of the redundant wavelet transform, for instance by using filters computed on more orientations.

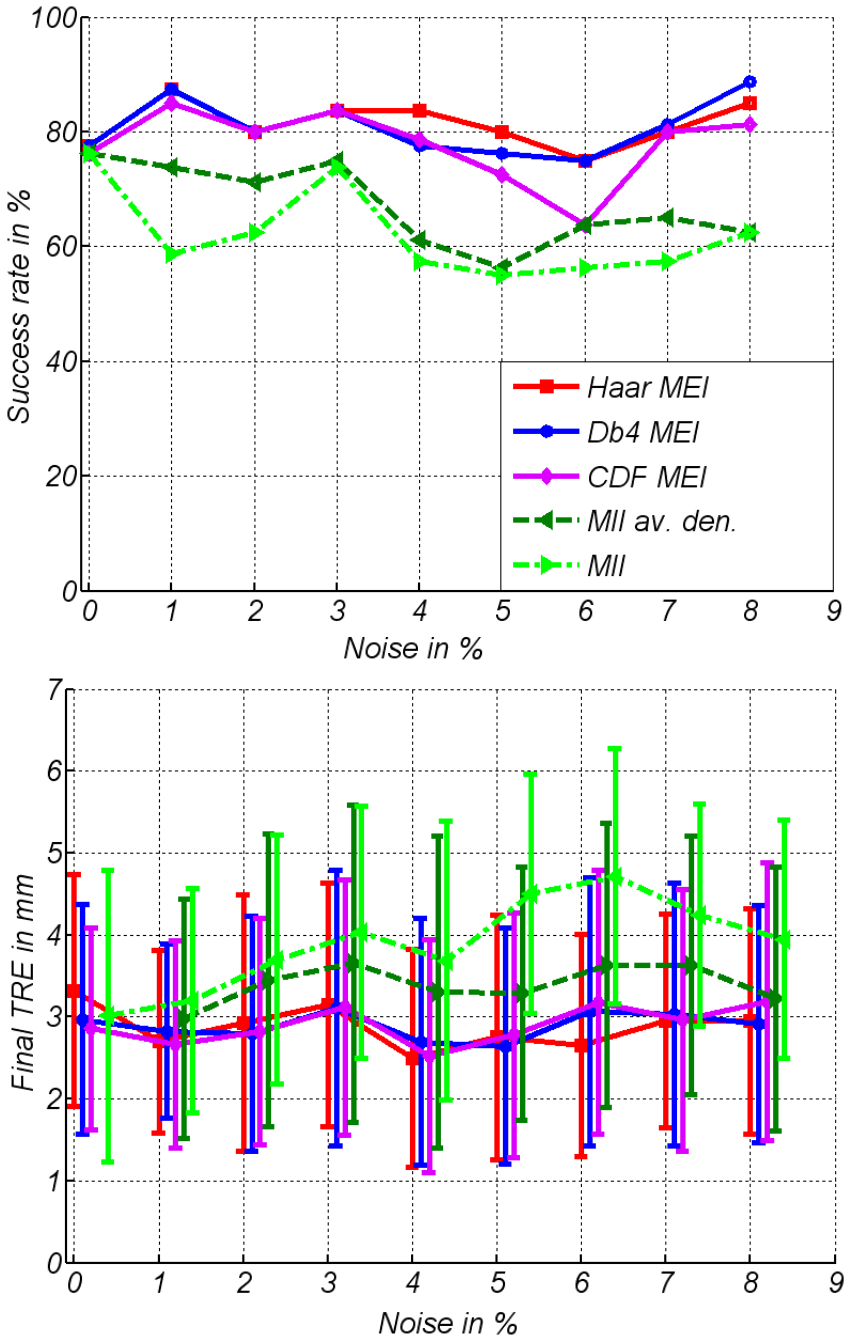

Figure B.11: 3D registration: Plot of **success rate** and **final TRE** according to an increasing amount of noise for MR-SPECT volumes.

LIST OF FIGURES

LIST OF TABLES

REFERENCES

[1] Ahmadi, S.-A., Baust, M., Karamalis, A., Plate, A., Boetzel, K., Klein, T., and Navab, N. Midbrain segmentation in transcranial 3d ultrasound for diagnosis. In *Proc. MICCAI* (2011), pp. 362–369.

[2] Ali, S., Veltri, R., Epstein, J. I., Christudass, C., and Madabhushi, A. Adaptive energy selective active contour with shape priors for nuclear segmentation and gleason grading of prostate cancer. In *Proc. of MICCAI Conf.* (2011).

[3] André, B. *Smart Atlas for Endomicroscopy Diagnosis Support: A Clinical Application of Content-Based Image Retrieval.* Ph.d. thesis, Ecole Nationale Supérieure des Mines de Paris, October 2011.

[4] André, B., Vercauteren, T., Buchner, A. M., Wallace, M. B., and Ayache, N. A smart atlas for endomicroscopy using automated video retrieval. *Medical Image Analysis 15*, 4 (August 2011), 460–476.

[5] André, B., Vercauteren, T., Buchner, A. M., Wallace, M. B., and Ayache, N. Learning semantic and visual similarity for endomicroscopy video retrieval. *IEEE Transactions on Medical Imaging 31*, 6 (June 2012), 1276–1288.

[6] Arthur, D., and Vassilvitskii, S. K-means++: The advantages of careful seeding. *SODA* (2007).

[7] Barbu, A., Athitsos, V., Georgescu, B., Boehm, S., Durlak, P., and Comaniciu, D. Hierarchical learning of curves application to guidewire localization in fluoroscopy. In *Proc. of CVPR Conf.* (2007).

[8] Berg, D., Seppi, K., Behnke, S., Liepelt, I., Schweitzer, K., Stockner, H., Wollenweber, F., Gaenslen, A., Mahlknecht, P., Spiegel, J., Godau, J., Huber, H., Srulijes, K., Kiechl, S., Bentele, M., Gasperi, A., Schubert, T., Hiry, T., Probst, M., Schneider, V., Klenk, J., Sawires, M., Willeit, J., Maetzler, W., Fassbender, K., Gasser, T., and Poewe, W. Enlarged substantia nigra hyperechogenicity and risk for Parkinson disease: a 37-month 3-center study of 1847 older persons. *Arch. Neurol. 68* (2011), 932–937.

[9] Bishop, C. M. *Pattern Recognition and Machine Learning.* Springer-Verlag, 2006.

[10] Bosch, A., Zisserman, A., and Muñoz, X. Image Classification using Random Forests and Ferns. In *IEEE International Conference on Computer Vision* (2007).

[11] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[12] Breiman, L., Friedman, J., Stone, C., and Olshen, R. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[13] Bronstein, M., Bronstein, A., Paragios, N., and Michel, F. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *Proc. of CVPR Conf.* (2010).

[14] Chang, S. G., Yu, B., and Vetterli, M. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. on Image Processing 9*, 9 (2000), 1532–1546.

[15] Chen, L., Seidel, G., and Mertins, A. Multiple Feature Extraction for Early Parkinson Risk Assessment Based on Transcranial Sonography Image. In *IEEE Int. Conf. on Image Processing* (2010).

[16] Christoyianni, I., Koutras, A., Dermatas, E., and Kokkinakis, G. Computer aided diagnosis of breast cancer in digitized mammograms. *Comp. Med. Imag. and Graph. 26* (2002), 309–319.

[17] Chung, A., Wells, W., Norbash, A., and Grimson, E. Multi-modal image registration by minimising kullback-leibler distance. In *Proc. of MICCAI Conf.* (2002).

[18] Collignon, A., Vandermeulen, D., Suetens, P., and Marchal, G. 3d multi-modality medical image registration using feature space clustering. In *Proc. of Computer Vision, Virtual Reality and Robotics in Medicine Conf.* (1995).

[19] Cortes, C., and Vapnik, V. Support-vector networks. *Machine learning 20*, 3 (1995), 273–297.

[20] Criminisi, A., Shotton, J., and Konukoglu, E. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Tech. rep., Microsoft Research Cambridge, UK, 2011.

[21] Criminisi, A., Shotton, J., Robertson, D., and Konukoglu, E. Regression Forests for Efficient Anatomy Detection and Localization in CT Studies. In *Jiang T., Navab N., Pluim J., Viergever M. (Eds.) MICCAI 2010 workshop in Medical Computer Vision* (2010).

[22] Cui, S., and Wang, Y. Redundant wavelet transform in video signal processing. *Proc. of Image Processing, Computer Vision, and Pattern Recognition Conf.* (2006), 191–196.

[23] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. *CVPR* (2005).

[24] Daubechies, I., Grossman, A., and Meyer, Y. Painless non orthogonal expansions. *J. Math. Phys. 27* (1986), 1271–1283.

[25] Dierckx, P. Curve and surface fitting with splines. *Oxford University Press, Inc., New York, NY, USA* (1993).

[26] Dollar, P., Welinder, P., and Perona, P. Cascaded pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition* (2010).

[27] El-Baz, A., Nitzken, M., Elnakib, A., Khalifa, F., Gimel'farb, G., Falk, R., and El-Ghar, M. A. 3d shape analysis for early diagnosis of malignant lung nodules. In *Proc. of MICCAI Conf.* (2011).

[28] Engel, K., and Toennis, K. D. Segmentation of the Midbrain in Transcranial Sonographies using a TwoComponent Deformable Model. *Annals of the British Machine Vision Association and Society for Pattern Recognition (BMVA) 4* (2009), 1–12.

[29] Fowler, J. E. The redundant discrete wavelet transform and additive noise. *IEEE Signal Processing Letters 12*, 9 (2005), 629–632.

[30] Frangi, A. F., Niessen, W. J., Vincken, K. L., and Viergever, M. A. Multiscale vessel enhancement filtering. In *Proc. of MICCAI Conf.* (1998).

[31] Freund, Y., Dasgupta, S., Kabra, M., and Verma, N. Learning the structure of manifolds using random projections. In *Advances in Neural Information Processing Systems* (2007).

[32] Gaenslen, A., Unmuth, B., Godau, J., Liepelt, I., Di Santo, A., Schweitzer, K. J., Gasser, T., Machulla, H. J., Reimold, M., Marek, K., and Berg, D. The specificity and sensitivity of transcranial ultrasound in the differential diagnosis of Parkinson's disease: a prospective blinded study. *Lancet Neurol 7* (May 2008), 417–424.

[33] Gan, R., and Chung, A. Multi-dimensional mutual information based robust image registration using maximum distance-gradient-magnitude. *Proc. of IPMI Conf.* (2005), 210–221.

[34] Geremia, E., Clatz, O., Menze, B. H., Konukoglu, E., Criminisi, A., and Ayache, N. Spatial decision forests for ms lesion segmentation in multi-channel magnetic resonance images. *NeuroImage 57(2)* (July 2011), 378–90.

[35] Geremia, E., Menze, B. H., Clatz, O., Konukoglu, E., Criminisi, A., and Ayache, N. Spatial decision forests for ms lesion segmentation in multi-channel mr images. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2010).

[36] Geurts, P., Ernst, D., and Wehenkel, L. Extremely randomized trees. *Machine Learning 63*, 1 (2006), 3–42.

[37] Glocker, B., Pauly, O., Konukoglu, E., and Criminisi, A. Joint classification-regression forests for spatially structured multi-object segmentation. In *12th European Conference on Computer Vision (ECCV)* (Firenze, Italy, October 2012).

[38] Grossman, A., and Morlet, J. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. 15* (1984), 723–736.

[39] Hammer, P. L., Hansen, P., and Simeone, B. Roof duality, complementation and persistency in quadratic 0 1 optimization. *Mathematical Programming 28* (1984), 121–155.

[40] Heibel, T. H., Glocker, B., Groher, M., Paragios, N., Komodakis, N., and Navab, N. Discrete tracking of parametrized curves. In *Proc. of CVPR Conf.* (2009).

[41] Ho, T. K. Random decision forests. In *Proc. of Document Analysis and Recognition Conf.* (1995).

[42] Ho, T. K. The random subspace method for constructing decision forests. *PAMI* (1998).

[43] Hofmann, M., Steinke, F., Scheel, V., Charpiat, G., Farquhar, J., Aschoff, P., Brady, M., Schölkopf, B., and Pichler, B. J. MRI-Based Attenuation Correction for PET/MRI: A Novel Approach Combining Pattern Recognition and Atlas Registration. *Journal of Nuclear Medicine* (2008).

[44] Hongli, S., and Bo, H. Image registration using a new scheme of wavelet decomposition. *IEEE Proc. of Instrumentation and Measurement Technology Conf.* (2008), 235–239.

[45] Ishikawa, H. Higher-order clique reduction in binary graph cut. In *Proc. of CVPR Conf.* (2009).

[46] Johnson, K. A., and Becker, J. A. The whole brain atlas. *http://www.med.harvard.edu/AANLIB/home.html*.

[47] Judenhofer, M. S., and et al. Simultaneous pet-mri: a new approach for functional and morphological imaging. *Nature Medicine*, 14 (2008), 459–465.

[48] Jurie, F., and Triggs, B. Creating efficient codebooks for visual recognition. *ICCV* (2005).

[49] Kalpathy-Cramer, J., and Hersh, W. Multimodal medical image retrieval: image categorization to improve search precision. *Int. Conf. on Multimedia Information Retrieval* (2010).

[50] Kier, C., Cyrus, C., Seidel, G., Hofmann, U. G., and Aach, T. Segmenting the substantia nigra in ultrasound images for early diagnosis of Parkinson's disease. *Int. J. of Computer Assisted Radiology and Surgery 2*, S1 (June 2007), S83–S85.

[51] Kinahan, P., Hasegawa, B., and Beyer, T. X-ray-based attenuation correction for positron emission tomography/computed tomography scanners. *Semin. Nuc. Med.* (2003).

[52] Kops, E. R., and Herzog, H. Template-based attenuation correction of PET in hybrid MR-PET. *Journal of Nuclear Medicine* (2008).

[53] Lee, D., Hofmann, M., Steinke, F., Altun, Y., Cahill, N. D., and Schölkopf, B. Learning similarity measure for multi-modal 3d image registration. In *Proc. of CVPR Conf.* (2009).

[54] Lepetit, V., and Fua, P. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 9 (2006), 1465–1479.

[55] Leventon, M. E., and Grimson, E. Multi-modal volume registration using joint intensity distributions. In *Proc. of MICCAI Conf.* (1998).

[56] Li, S., Peng, J., Kwok, J. T., and Zhang, J. Multimodal registration using the discrete wavelet frame transform. *Proc. of ICPR Conf.* (2006), 877–880.

[57] Luan, H., Qi, F., and Shen, D. Multi-modal image registration by quantitative-qualitative measure of mutual information (q-mi). *Proc. of CVBIA Conf.* (2005), 378–387.

[58] Ma, J. Dixon Techniques for Water and Fat Imaging. *J Mag. Res. Im.* (2008).

[59] Madabhushi, A., Feldman, M. D., Metaxas, D. N., Tomaszeweski, J., and Chute, D. Automated detection of prostatic adenocarcinoma from high-resolution ex vivo mri. *IEEE Transactions on Medical Imaging* (2005).

[60] Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. Supervised dictionary learning. *NIPS* (2008).

[61] Mallat, S. G. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 11*, 7 (1989), 674–693.

[62] Martinez-Möller, A., Souvatzoglou, M., Delso, G., Bundschuh, R. A., Chefd'hotel, C., Ziegler, S. I., Navab, N., Schwaiger, M., and Nekolla, S. G. Tissue Classification as a Potential Approach for Attenuation Correction in Whole-Body PET/MRI: Evaluation with PET/CT Data. *Journal of Nuclear Medicine 50*, 4 (2009), 520–526.

[63] Moigne, J. L., Campbell, W. J., and Cromp, R. F. An automated parallel image registration technique based on the correlation of wavelet features. *IEEE Trans. On Geoscience and Remote Sensing 40*, 8 (2002), 1849–1864.

[64] Moosmann, F., Triggs, B., and Jurie, F. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS* (2006).

[65] Müller, H., Kalpathy-Cramer, J., Eggel, I., Bedrick, S., Jr., C. E. K., and Hersh, W. Overview of the clef 2010 medical image retrieval track. *ImageCLEF 2010* (2010).

[66] Nguyen, M. H., and de la Torre, F. Metric learning for image alignment. *International Journal on Computer Vision* (2009).

[67] Nistér, D., and Stewénius, H. Scalable recognition with a vocabulary tree. *CVPR* (2006).

[68] Ojala, T., Pietikäinen, M., and Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition 29*, 1 (1996), 51–59.

[69] Oliver, A., Llado, X., Freixenet, J., and Marti, J. False positive reduction in mammographic mass detection using local binary patterns. In *Proc. of MICCAI Conf.* (2007).

[70] Oubel, E., Frangi, A. F., and Hero, A. O. Complex wavelets for registration of tagged mri sequences. *Proc. of ISBI* (2006), 622–625.

[71] Özuysal, M., Calonder, M., Lepetit, V., and Fua, P. Fast Keypoint Recognition using Random Ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 3 (2010), 448–461.

[72] Pauly, O., Ahmadi, S.-A., Plate, A., Boetzel, K., and Navab, N. Detection of substantia nigra echogenicities in 3d transcranial ultrasound for early diagnosis of parkinson disease. In *in Proc. International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)* (2012).

[73] Pauly, O., Glocker, B., Criminisi, A., Mateus, D., Martinez-Moeller, A., Nekolla, S., and Navab, N. Fast multiple organs detection and localization in whole-body mr dixon sequences. In *MICCAI* (2011).

[74] Pauly, O., Heibel, H., and Navab, N. A machine learning approach for deformable guide-wire tracking in fluoroscopic sequences. In *MICCAI* (2010).

[75] Pauly, O., Mateus, D., and Navab, N. Building implicit dictionaries based on extreme random clustering for modality recognition. In *MICCAI Workshop on Medical Content-Based Retrieval for Clinical Decision Support* (2011).

[76] Pauly, O., Mateus, D., and Navab, N. Stars: A new ensemble partitioning approach. In *ICCV Workshop ITINCVPR* (2011).

[77] Pauly, O., Padoy, N., Poppert, H., Esposito, L., Eckstein, H.-H., and Navab, N. Towards Application-specific Multi-modal Similarity Measures: a Regression Approach. In *MICCAI Workshop on Probabilistic Models in Medical Image Analysis (PMMIA)* (2009).

[78] Pauly, O., Padoy, N., Poppert, H., Esposito, L., and Navab, N. Wavelet Energy Map: A Robust Support for Multi-modal Registration of Medical Images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).

[79] Perbet, F., Stenger, B., and Maki, A. Random forest clustering and application to video segmentation. In *BMVC* (2009).

[80] Perronnin, F., Dance, C., Csurka, G., and Bressan, M. Adapted vocabularies for generic visual categorization. *ECCV* (2006).

[81] Plate, A., Ahmadi, S.-A., Pauly, O., Klein, T., Navab, N., and Bötzel, K. 3D Sonographic Examination of the Midbrain for Computer-Aided Diagnosis of Movement Disorders. *Ultrasound in Medicine & Biology (UMB)* (2012).

[82] Quinlan, J. R. *C4.5: Programs for Machine Learning*. 1993.

[83] Rangayyan, R., Ayres, F., and Desautels, J. A review of computer-aided diagnosis of breast cancer: Toward the detection of subtle signs. *Journal of the Franklin Institute 3–4*, 334 (2007), 312–348.

[84] Roche, A., Malandain, G., Ayache, N., and Prima, S. Towards a better comprehension of similarity measures used in medical image registration. In *Proc. of MICCAI Conf.* (1999).

[85] Rother, C., Kolmogorov, V., Lempitsky, V., and Szummer, M. Optimizing binary mrfs via extended roof duality. In *Proc. of CVPR Conf.* (2007).

[86] Safi, A., Baust, M., Pauly, O., Castaneda, V., Lasser, T., Mateus, D., Navab, N., Hein, R., and Ziai, M. Computer-aided diagnosis of pigmented skin dermoscopic images. In *MICCAI Workshop on Medical Content-Based Retrieval for Clinical Decision Support* (2011).

[87] Sahiner, B., Chan, H., Wei, D., Petrick, N., Helvie, M., Adler, D., and Goodsit, M. Image feature selection by a genetic algorithm: Application to classification of mass and normal breast tissue. *Med. Phys. 23* (1996), 1671–1684.

[88] Sharman, R. A fast and accurate way to register medical images using wavelet modulus maxima. *Pattern Recognition Letters 21*, 6 (2000), 447–462.

[89] Shotton, J., Johnson, M., and Cipolla, R. Semantic texton forests for image categorization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (2008).

[90] Sivic, J., and Zisserman, A. Video google: A text retrieval approach to object matching in videos. In *ICCV* (2003).

[91] Smola, A. J., and Schölkopf, B. A tutorial on support vector regression. *Statistics and Computing 14* (2004), 199–222.

[92] Strehl, A., and Ghosh, J. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res. 3* (Mar. 2003), 583–617.

[93] Studholme, C., Hill, D. L. G., and Hawkes, D. J. Automated 3-d registration of mr and ct images of the head. *Medical Image Analysis 1*, 2 (1996), 163–175.

[94] Taki, A., Pauly, O., Setarehdan, S. K., Unal, G., and Navab, N. Ivus-based histology of atherosclerotic plaques: improving longitudinal resolution. In *SPIE Medical Imaging* (2010).

[95] Taki, A., Roodaki, A., Pauly, O., Setarehdan, S. K., Unal, G., and Navab, N. A new method for characterization of coronary plaque composition via ivus images. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (2009).

[96] Tanner, C., and et. al. Classification improvement by segmentation refinement: Application to contrast-enhanced mr-mammography. In *Proc. of MICCAI Conf.* (2004).

[97] Tao, Y., Lu, L., Dewan, M., Chen, A. Y., Corso, J., Xuan, J., Salganicoff, M., , and Krishnan, A. Multi-level ground glass nodule detection and segmentation in ct lung images. In *Proc. of MICCAI Conf.* (2009).

[98] Tiwari, P., Kurhanewicz, J., Rosen, M., and Madabhushi, A. Semi supervised multi kernel (sesmik) graph embedding: Identifying aggressive prostate cancer via magnetic resonance imaging and spectroscopy. In *Proc. of MICCAI Conf.* (2010).

[99] van de Loo, S., Walter, U., Behnke, S., Hagenah, J., Lorenz, M., Sitzer, M., Hilker, R., and Berg, D. Reproducibility and diagnostic accuracy of substantia nigra sonography for the diagnosis of parkinson's disease. *Journal of Neurology, Neurosurgery & Psychiatry 81*, 10 (2010), 1087–1092.

[100] Viola, P., and Jones, M. J. Robust real-time face detection. *International Journal on Computer Vision 57*, 2 (2004), 137–154.

[101] Viola, P., and Wells, W. Alignment by maximization of mutual information. *International Journal of Computer Vision* (1997).

[102] Vlaar, A., de Nijs, T., van Kroonenburgh, M., Mess, W., Winogrodzka, A., Tromp, S., and Weber, W. The predictive value of transcranial duplex sonography for the clinical diagnosis in undiagnosed Parkinsonian syndromes: comparison with SPECT scans. *BioMed Central Neurology 8* (2008), 42.

[103] Walter, U., Dressler, D., Probst, T., Wolters, A., Abu-Mugheisib, M., Wittstock, M., and Benecke, R. Transcranial brain sonography findings in discriminating between parkinsonism and idiopathic Parkinson disease. *Arch. Neurol. 64(11)* (2008), 1635–1640.

[104] Wang, P., Chen, T., Zhu, Y., Zhang, W., Zhou, S. K., and Comaniciu, D. Robust guidewire tracking in fluoroscopy. In *Proc. of CVPR Conf.* (2009).

[105] Winn, J., Criminisi, A., and Minka, T. Object categorization by learned universal visual dictionary. *ICCV* (2005).

[106] Yang, L., Jin, R., Sukthankar, R., and Jurie, F. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Proc. of CVPR Conf.* (2008).

[107] Zhang, X., McKay, C. R., and Sonka, M. Tissue characterization in intravascular ultrasound images. *IEEE Transactions on Medical Imaging* (1998).

[108] Zheng, Y., Baloch, S., Englander, S., Schnall, M. D., and Shen, D. Segmentation and classification of breast tumor using dynamic contrast-enhanced mr images. In *Proc. of MICCAI Conf.* (2007).

[109] Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M., and Comaniciu, D. Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. *IEEE Transactions on Medical Imaging 27*, 11 (2008), 1668–1681.

[110] Zheng, Y., Bogdan, and Comaniciu, D. Marginal Space Learning for Efficient Detection of 2D/3D Anatomical Structures in Medical Images. In *Information Processing in Medical Imaging* (2009).

[111] Zhou, J., Chang, S., Metaxas, D., Zhao, B., Ginsberg, M., and Schwartz, L. Automatic detection and segmentation of ground glass opacity nodules. In *Proc. of MICCAI Conf.* (2006).

[112] Zhou, S. K., Zhou, J., and Comaniciu, D. A boosting regression approach to medical anatomy detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (2007).

[113] Zhou, S. K., Zhou, J., and Comaniciu, D. A boosting regression approach to medical anatomy detection. In *Proc. of CVPR Conf.* (2007).

[114] Zitova, B., and Flusser, J. Image registration methods: a survey. *Image and Vision Computing 21*, 11 (2003), 977–1000.

[115] Zollei, L., Fisher, J., and Wells, W. A unified statistical and information theoretic framework for multi-modal image registration. In *Proc. of IPMI Conf.* (2003).