Universitat
Autònoma
de Barcelona

A Semi-Supervised Statistical Framework and Generative Snakes for IVUS Analysis.

A dissertation submitted by **Oriol Pujol Vila** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informática**.

Bellaterra, July 12, 2004

Director: **Dra. Petia Radeva Ivanova**
Universitat Autònoma de Barcelona
Dept. Informàtica & Computer Vision Center

Centre de Visió
per Computador

This document was typeset by the author using LaTeX 2_ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN 84–922529–8–7

Printed by Ediciones Gráficas Rey, S.L.

a la Silvia

Agraïments

A l'hora d'escriure aquestes linies, em venen a la ment tota la gent que d'alguna forma o altra ha participat en el desenvolupament d'aquesta tesi. Alguns, aportant idees, altres, recolzant-les de forma directa o indirecta, i la majoria suportant-me. De fet, aquells qui mereixeu agraïment ja ho sabeu, però mai esta de més recordar alguns noms.

En primer lloc, he d'agraïr a Juan J. Villanueva, director del Centre de Visió per Computador, l'oportunitat que m'ha dispensat per poder realitzar aquesta tesi en matèria de visió, i la possibilitat que m'ha ofert per poder formar part d'aquesta comunitat científica, investigadora i docent.

Vull agrair a la Petia, directora d'aquesta tesi, les hores dedicades a que es pogués dur a terme, i el seu suport incondicional, així com la confiana i llibertat que m'ha donat per poder decidir com encaminar la tesi.

Vull extendre aquests agraïments a en Jordi Vitriá i a la Maria Vanrell, que m'ha recolzat i amb els quals he compartit moltes hores de xerrades d'inestimable valor i guia a l'ombra.

Als incondicionals que "patim" el formar part del grup d'imatge mèdica i que tantes hores i idees hem compartit. Especialment, vosaltres, Misael, David (Rotger), Debora i Fernando, que hem criticat tant com hem volgut.

A l'Ernest, el meu company de despatx que li ha tocat la desgracia de tenir-me a menys de dos metres cada dia.

Als meus companys de "promoció" : Robert, Cristina, Poal, Xevi, Juanma i David Guillamet. Ànims Robert, que només quedem dos per ser doctors.

Al Xavier Binefa, que m'ha ajudat quan li he demanat, i amb el qual hem tingut una molt bona relació. I amb els seus "acolits" i cracks en Ramon i el Botey. Podria dir moltes coses sobre vosaltres però probablement sobrin les paraules doncs ja sabeu per on va l'agraïment.

A la Gemma, he d'agrair-te moltes converses, per, sobretot, em vas salvar amb el *vals.*

Al Ramon Baldrich, Josep Lladòs, Antonio, Felipe, Enric i Ricardo, tots ells *cracks* amb els quals he conviscut molt amigablement en aquest període durant el qual ha durat la tesi.

A tres bons companys, l'Oriol Ramos, en David Masip i el Xavier Otazu.

A la Raquel i al Joan Masoliver, que m'ha salvat de més d'un problema tècnic i no han posat mai cap impediment per ajudar-me en temes que no tenen massa que veure amb la tesi.

A la M. José, M. Carmen, Ana Cèlia i Montse, per ajudar-me, sobretot, en ”problemes logístics”.

I a tots aquells que sense voler m’he descuidat i mereixieu estar aquí.

Finalment, vull agrair molt especialment a la meva familia el seu suport. Al meu germà Xavi, als meus pares Joaquim i Maria Dolors, així com a la padrina i a la iaia.

I la última linia va dedicada a agrair a la persona que ara mateix més li toca patir-me, que ha hagut d’armar-se de paciència i traquilitat per poder suportar a un insuportable Oriol; a la meva dona, la *Silvia*.

One of the most important topics in computer vision is pattern recognition and classification in images. Any classification process requires from a feature extraction process and a learning technique that categorizes each data sample. However, sometimes, it is not enough to have just a classification since we could need to introduce high-level knowledge constraints to obtain a meaningful classification. Deformable models are one of the possible tools to achieve this goal.

This PhD thesis describes several new techniques to be used in this scenario regarding deformable models and classification theory. The definition of deformable models guided using a external potential derived from a generative model is proposed. This approach is called *generative snakes.* To illustrate this process parametric snakes in a texture based context are used. The extension of the former work to geodesic deformable models is done by reformulating the geometric deformation process, leading to the *Stop and Go* formulation. A new tool for mixing labelled and unlabelled data for semi-supervised and particularization problems is developed and validated. This new technique allows supervised and unsupervised processes to compete for each data sample, defining the *supervised clustering hybrid competition scheme.*

These techniques are motivated by and applied to *medical image analysis*, in particular to *Intravascular Ultrasound (IVUS)* tissue segmentation and characterization. This work also studies the tissue characterization problem in IVUS images and defines a new framework for automatic plaque recognition.

Contents

List of Tables

List of Figures

Preface

I am really happy and proud to write these lines since it means I have fulfilled one of the requirements for being a true PhD. This work is a compilation of short and long stories, noticeable and unrecognized work, an everlasting compendium of processed thought in matters of computer vision, pattern analysis and machine intelligence, and utterly a product of five years of research and education.

A lot of things have changed around me in this time but, probably, the most notable ones are those that changed inside me. And maybe this is what matters, more than the contents of a research or the results of a technique. Is this what distinguish a PhD from a non-PhD?

However, this metamorphosis is induced. It is caused by an education, a self-education since it is a path to walk alone, and an environment-driven education for I have been also guided explicitly and concealedly. An education that has tried to bring balance to two of the most inner desires in a PhD candidate, the desire for theorizing and the desire for applicability. May this work tame both desires.

Chapter 1

Introduction

One of the most important topics in computer vision is pattern recognition and classification in images. Any classification process, in real life, is usually divided into two sub-processes: A first step, in which descriptors or measures are taken for each of the classes. And a second step, which is the learning/classification process itself. The first step is also known as *feature extraction*, and its goal is to reduce the complexity associated to real scenes by means of extracting measures to characterize the original data, in our case image regions of interest or objects. The second step consists of a learning/decission technique. If the classification is *supervised*, *a priori* knowledge of what the different classes are, is used to train a learning process. If no information of the classes desired is known, we talk about *unsupervised* classification/learning. Regardless of the degree of supervision, the decision step is the responsible to assign labels to each of the samples in the space created by the the data provided by the feature extraction process. However, sometimes, it is not enough to have a mere classification since we could need a compact representation of a shape or, simply, to include high-level knowledge to obtain a meaningful classification. In this matter is where deformable models stands as one of the possible tools to achieve this goal.

1.1 The Goal of this work

In the former scenario we emplace this PhD work. Our aim is to create a set of powerful tools to help in those steps. In particular we have designed techniques for the second and third step of the described process, namely: deformable models and classification theory. Figure 1.1 shows a little scheme of the theories and techniques derived in this work.

We have also attacked a real problem with all these tools, intravascular ultrasound image analysis and tissue characterization.

The techniques and theories developed in this work lies in two great groups:

1. **Deformable models:** The work in deformable models is divided in two groups. The first, is the definition of deformable models guided using a external potential derived from a generative model. We call this approach *generative snakes.*

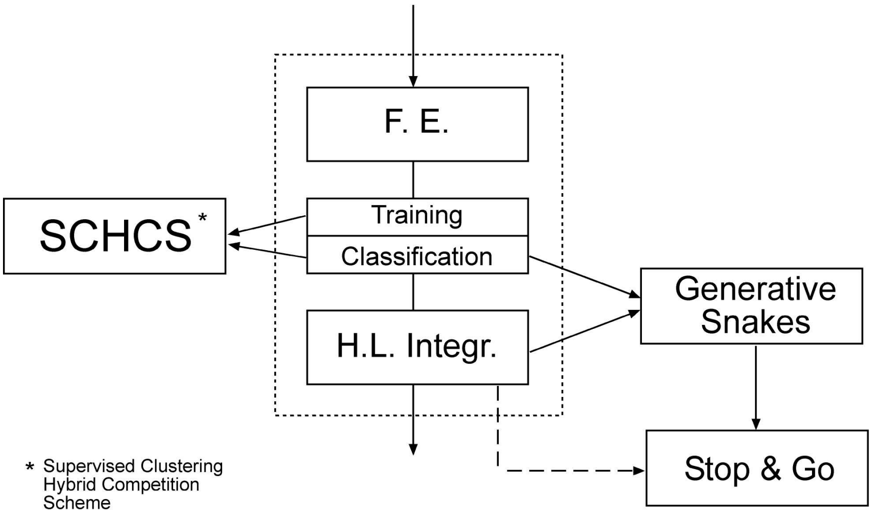


Figure 1.1: General overview of the main theories of the PhD work

To illustrate this process we use parametric snakes in a texture based context. The second, is the extension of the former work to geodesic deformable models. However, this extension is not trivial since we have to reformulate the whole deformation process. This new formulation for geodesic snakes is called *Stop and Go snakes*. With this new formulation we can use the former generative approach as well as allow certain control over the deformation process by decoupling the forces influencing the snake deformation *a la* parametric snakes.

2. **Classification Theory:** Classification theory is a vast region for researchers, since there are a lot of virgin niches and topics; one of this *en vogue* topic is semi-supervision techniques. Semi-supervision is a general denomination for the process of mixing labelled with unlabelled data. This mixture can be done in two particular fields: semi-supervised classification and semi-supervised clustering. The first one looks for help in classification in the unlabelled data. This is, how unlabelled data can improve the classification. The second one aims for a more meaningful clustering by aids of labelled data. Although there are approaches that clearly are designed to address each of the problems, the line separating both approaches can be sometimes really fuzzy. Our work in classification theory is twofold: first, define a semi-supervision technique and second, define what we call *particularization* problems, as well as a technique to solve them.

However, our work has been highly motivated by the application of those techniques in *medical image analysis*, since it has been one of our most important and active lines of work. In particular we have been toying with *Intravascular Ultrasound (IVUS)* images analysis, segmentation and characterization. Our concern with this kind of images regards the image interpretation of meaningful structures in the IVUS image, such as intima and adventitia layers and characterization of vulnerable plaque by identifying the kind of tissue present in the plaque region.

1.2 Outline / State-of-the-Art / Proposal

In this section we provide a general information background for the reader to set a general overview of each of the fields involved and how this work address each of the problems.

1.2.1 Deformable models

Active contours ([61], [51], [53], to mention just a few) are well known tools in computer vision for image segmentation [68], [71], [87], [85] and shape recovery. These techniques interpret low-level information (i.e. edge points) under general high-level assumptions/constraints to assure well-possedness of the segmentation problem. In particular, snakes are defined by internal and external constraints to deform a curve until it adapts to the object of interest. The internal constraints control continuity and smoothness of the snake, meanwhile the external ones are responsible for adjusting to the image features.

In general, there are two different approaches in current snakes formulations: the parametric (physics-based) and the geometric (geodesic) definition.

Parametric deformable models [61] use Newton mechanics laws to define the internal constraints of the model. In these physical terms, such constraints are given in terms of elasticity and stretching of the snake. By working with an explicit parametrization of the curve, the model restricts the search space of the segmentation solutions to single objects.

An alternative to physics-based snakes are the geodesic active contours [51]. Geodesic snakes are based on the theory of curve evolution and level sets methods [86]. In this geometric setting, the snake deforms in a Riemannian surface until its length, dependant on image features, is minimum. Its implicit level sets formulation [86] can naturally deal with topological changes during the snake evolution. This is a main advantage in those cases where the topology of the target object is not known *a priori*.

However, topologically invariant snakes are also important when object topology is known *a priori*, since the model introduces a hard constraint to restrict the search space. In our case, given that we are looking for an object represented as a connected region, without loss of generality we have applied parametric snakes to our first approach to define *generative snakes*. However, as can be seen later, this fact is not a restriction and geodesic snakes can be applied in the same framework.

Parametric as well as geometric active contour models depend in high degree on the image interpretation. Using heuristic external energy like image location with high/extreme image gradient or high response of edge detectors leads to different convergence problems: a need for close initialization, problems with stopping criterion, impossibility to converge to concave boundaries, etc. These problems have been addressed and partially solved in various papers with the use of multi-resolution methods, solenoidal external fields,etc.[69] [73] [55] but the most widespread method is the Gradient Vector Flow.[80] [81] This method regularizes the gradient of the boundaries leading to a smooth vector field when far from boundaries and keeping the gradient orientation near the boundaries. On the other hand, all these methods proposed until now are sensitive to the edge noise caused by a heuristic edge detection which can attract the active contour towards false contours. In this work, we claim that the Generalized Gradient Vector Flow (GGVF) is a general regularization method that can be applied to our approach in order to improve the classification of homogeneous regions.

The importance of snakes using external potential fields dependent on the feature space for segmentation has been noted by different authors reporting promising results.[82] [71] [72] Paragios et al.[71] proposed a framework using geodesic snakes and fusing the information from region and boundary; Zhu et al.[82] proposed a framework fusing split-merge techniques with active contours. Both approaches coincide in the way the feature information is inserted in the framework; they both minimize the conditional probability for a pixel being a boundary, and the active contour is led by this term seeking a trade-off among the different regions. Note the fact that the approaches mentioned [71] [82] use the probabilistic information about the features as a quotient of probabilities, the probability of the region of interest and the probability of the rest ($\alpha\mathbf{log}\frac{p_B(I(u))}{p_A(I(u))}$). One can observe that this force shrinks or expands the snake to zones with balanced probabilities. As the term just depends on the quotient of both probabilities and not on the magnitude of the same, the attraction to regions with low probabilities in both models could be a problem of relevance. The term probabilistic snake appears in the literature in different contexts, [68] however the Terzopoulos[79] and Szeliski[78] approaches apply to a different problem. They incorporate a prior model in terms of probability distributions, and seek the probability of an image given a model using a Bayesian framework and cast the internal energy in this frame by converting it to a prior distribution over expected shapes. Another work using statistical information and fusing it with deformable models is the approach proposed by Cootes[54], in which a statistical model of appearance is matched to an image.

Our contribution: *Generative Snakes*

The objective of our work is to provide a statistical approach for region segmentation framework using generative based external potential fields in deformable models. These models are applied to regions of interest defined by certain values of the features. The main contributions of our work are twofold: a) we propose an improved active contour model that deforms on a likelihood map instead of heuristically constructed edge map in order to obtain regions with homogeneous feature description, that is, maximizing the likelihood of the set of pixels to represent the same object; b) on the other hand, we propose a new approach for supervised segmentation introducing general assumptions about the smoothness of object boundaries and solving the problem of hole appearance in the classified regions of pixels. Moreover, the concept of finding an exact threshold to classify pixels to different objects is relaxed.

In contrast to the explicit computation of probability of regions belonging to the same distribution during the region competition[82], in our case we construct a likelihood map assigning to each pixel of the image the likelihood of representing a target object. This fact allows us to analyze explicitly the potential field of the snake to modify it to assure snake convergence. In our preliminary work[75], we discuss the advantages of using explicit likelihood compared to implicit probability estimates in the snake formulation.

To concretize this approach we apply **texture modelling** to natural scenes and medical images segmentation, since they can be seen as to be mainly composed of textured objects. Regarding this matter we consider that the problem of texture analysis has played a prominent role in computer vision to solve problems of object segmentation and retrieval in numerous applications: medical image analysis, robotics, digital libraries, etc. Most approaches for texture analysis follow two stages: first during the modelling stage a feature space is defined to describe texture appearance. Different approaches have been presented in the bibliography: Markov Random Field-models [66], co-occurrence matrices [57], banks of filters [59], wavelets [65], fractal dimension [52], etc. The second stage represents an optimization procedure of classifying image pixels into different textures using both supervised and unsupervised classification processes. [63] [58] [74] [66] [59] [67] [62] Supervised texture segmentation is still a non-solved problem of great interest in medical imaging and retrieval from image databases. This statistical modelling supposes that the texture prototypes are the result of a generative model of the probability distributions of the random fields. Hence, a likelihood map can be constructed assigning to each pixel a likelihood value to represent a given texture. This approach has two drawbacks: the usual way of classification in this framework is done by applying a fixed threshold on the likelihood in order to make a decision about the membership in a textured class; and moreover, the algorithm does not assure that the regions classified, representing target object, are connected regions or regions without holes.

Furthermore, texture analysis suffers from an important problem: precise texture segmentation. This is due to the fact that texture descriptors need a substantial spatial support to extract local texture. As a result, texture boundaries differ significantly from both textures and it is difficult to recognize them, because of the low likelihood of belonging to learned texture regions. On the other hand, human beings are very good at recognizing boundaries between textured regions, when regions have significant size and smooth shape. This observation justifies our approach: to introduce as a third stage of the texture analysis, the organization of the high likelihood valued regions into the final solution. In order to achieve it we propose to incorporate high-level knowledge into the classification process of textured regions in the form of snakes. The unification of the classification task and the snake high-level interpretation led us to define *generative snakes*.

In practice this approach has been implemented in the following way: A set of texture features (i.e. co-occurrence matrix measures) is extracted from the desired texture pattern and is reduced by applying Linear Discriminant Analysis. In the reduced feature space a Gaussian Mixture Model is created for the desired texture patterns, and a likelihood map is constructed in terms of the likelihood of a mixture model. In this map, the location of rough changes (high gradients) of likelihood values represents candidates for textured boundaries as long as the shape of the region fulfills the smoothness constraint. To improve the convergence properties of our statistic snake, a regularization of the likelihood map is applied by means of the generalized gradient vector flow [80]. The snake deforms on the regularized likelihood map until it stops on the boundaries of regions with high likelihood of representing the object with the target texture.

On the other hand, in most snake applications of the segmentation problem, the metric is defined based on the image gradient in order to detect edges. The geodesic snake deformation is determined by two distinct terms in its evolution equation.

In this formulation, the first term is the normal component of the gradient of the metric and rules convergence to contours. The second one, dependant on the snake curvature, gives regularity to the snake and defines its motion at null gradient regions. That is, it also influences on the convergence scheme. The double role of the curvature term has some disadvantages: on one hand, because it is a second order term, it hinders the numeric scheme; on the other hand, it difficulties snake convergence to concave areas. The usual way to overcome poor convergence to non-convex shapes is to add a constant motion term, the balloon force [53], that pushes the snake into concave regions. In order to guarantee convergence into such regions, its magnitude there should be greater than the absolute value of the curvature. A major inconvenience is that the former requirement difficulties stopping the snake at the desired contour. Because balloon forces correspond to a minimization of the area enclosed by the snake, they can be embedded in a region based scheme.

Region-based methods are born to introduce region information in the geodesic formulation. They aim at finding a partition of the image such that the descriptors of each of the regions conform to a given "homogeneity" criterion. It follows that the force guiding the snake must be derived from the competition of the descriptors. In Ronfard et al. [88] the velocity function is proportional to the difference of simple statistical features. In Zhu et al. [82] and Paragios et al. [71], the autors define the region evolution as a quotient of probabilities corresponding to different regions. In Yezzi et al. [92] a dynamical approach is defined in which the evolution of the curve is described by the difference of mean gray levels inside and outside the evolving front at each iteration. In the same way, Besson et al. [85] propose a difference of simple statistics, variance and covariance matrix, inside and outside the curve, also recomputing that measures at each iteration. Chakraborty et al. [95] consider an evolution using a Fourier parametrization over the original image and a previously classified image regions. Most of the methods are based on simple non-supervised descriptors of image regions. This limits it applicability to segmentation of simple images. However, complex scenes such as natural and real images need more accurate descriptors for their segmentation. In this way, supervised feature extraction schemes are more suitable for the task [71], [89].

Our contribution: *Stop and Go formulation*

Considering the general problem of region-based segmentation, we propose a new geodesic snake formulation that assures a more efficient behavior. Given that convergence and regularity are the key issues of the snake formulation, we propose a new definition where the terms ruling these properties are decoupled. As a result, the curvature term does not interfere in the convergence process but restricts its role to the shape regularity in the last stages of the snake deformation. By removing the influence of the snake curvature from the convergence step, any *global* vector field properly defining the target contour curve as its set of equilibrium points ensures convergence. However, current external forces either restrict to a band [51] (non-global) around object contours or have saddle points [93] (target curve not properly defined) that prevent the snake from entering into concave regions. We propose using the decoupling strategy for the definition of a global vector field having the target curve as the set of equilibrium points. It can be seen that any vector field fulfilling the above requirement splits into an exterior attractor vector field (GO term) and an inner repulsive one (STOP term) which sum cancels on the curve of interest. This is the milestone in the definition of our *Stop and Go* snakes: defining separately the GO and the STOP term and glue them together by means of a characteristic function. Because we want to ensure snake convergence whatever curve concavity is, a balloon force will be our GO term. Since the curvature term has been removed from the convergence step, there is no restriction on the magnitude of this force, which prevents the snake from collapsing to a point. For the STOP vector field any standard external force restricted to the object interior suffices. Its choice hinges upon the particular segmenting problem.

A mask defining the object of interest would be the ideal tool to bound the scope of the curvature term and to perform any decoupling. To address segmentation of real images, we propose to use *likelihood maps* as an approximation of the object characteristic function. A likelihood map represents the likelihood value of each pixel of the image. Since it also characterizes the object of interest we introduce its use as a STOP term. In this manner the *Stop and Go* scheme presented in this paper is particulary suited for feature space based segmentation, such as textures [87] [71] [58], color [90] [82], motion [72], etc.

Our new formulation has several advantages over current snake schemes. On one hand, except for the very last refinement steps, the technique admits arbitrary large time increments in the iterative Euler scheme used in its implementation. On the other, by removing curvature influence from the convergence process, we can build a robust vector field to be used as an external force/attraction term that ensures convergence but, at the same time, snake stabilization. The use of likelihood maps also introduces a great advantage by allowing most generative schemes to operate in our snake framework.

1.2.2 Semi-supervised classification and Particularization

Semi-supervised classification and clustering line of work has been very active recently since several authors point out the beneficial effects that unlabelled data can have. For instance, McCallum and Nigam [118] declared that "by augmenting a small set [of labelled samples] with a large set of unlabelled data and combining the two pools with EM, we can improve our parameter estimates". This was not the first insightful citation of unlabelled data as a meaningful way to aid processes, since O'Neill [119] previously commented that "unclassified observations should certainly not be discarded". Those statements lead a new flow of work regarding the assessment of the validity of unlabelled data as possible improvers of a classifier performance. Up to this day, no clear arguments have been made if unlabelled data is beneficial or, on the other hand, detrimental for the classification process. In particular [120] devises the obvious conditions for the semi-supervised process to succeed.

Semi-supervised learning is born due to the necessity to find methodologies working with very few data points. This kind of work is emphasized by the amount of applications in which there are large pools of unlabelled data but it is very expensive to create a full labelled training set. The main idea in semi-supervised classification is to try to improve the training set by adding unlabelled data. In particular we have three main lines of work: the first one is to try to increase the pool of training data by means of supervised classification, using co-training approaches [121]. The second line particularizes the problem to transductive support vector machines [122]. And the last one describe a technique for incorporating unlabelled data into training using Expectation Maximization [123].

The Co-Training approach uses diversification in features or classifiers to add to the training set those data points in which the different classifiers agree. For instance, given a set of unlabelled data points, and two sets of different features over those data points we can add to the labelled set those unlabelled data that once classified, the classification label obtained is the same for both feature/classifier sets. However, this approach has a difficult time with disagreed samples. The Expectation Maximization technique shows how entangled are the supervised processes and unsupervised processes when dealing with semi-supervision of data.

Semi-supervised clustering uses class labels or pairwise constraints on some examples to aid the unsupervised process [124] [125]. To use pairwise constraints, a set of data pairs is labelled as *must-link* if both must be clusterized in the same cluster, or *cannot link* if the data points belong to different clusters. In particular if the labelled data represents all the classes, both semi-supervised clustering and semi-supervised classification can be used for categorization. However, in many domains, the knowledge of the classes is incomplete. This is the frame in which we must place semi-supervised clustering. This can lead to modification of the existing set of categories or to reflect irregularities of the data itself. In this category there are two main lines of work: search-based methods and similarity-based methods.

Search-based techniques rely on user labels or constraints to bias the search of the clusters to a meaningful partition. Several works show how to accomplish this goal. In [126] Demiriz et al. change the objective function to satisfy specific constraints. Wagstaff et al. [127] force some constraints to be met while the clustering process is being performed. And Sinkkonen et al. [128] use an auxiliary space created using side-information from conditional distributions to guide the clustering process. On the other hand, similarity-based methods use labelled data for training a similarity metric, that is used later for the clustering procedure. Several works focus on this line: Bilenko et al. [129] use string-edit distance trained using Expectation Maximization, Cohn et al. [130] use Jehnsen-Shannon divergence with gradient descent, Klein et al. [131] use Euclidian distance with shortest path algorithm, Hillel et al. [132] and Xing et al. [125] use Mahalanobis distance trained using convex optimization.

Our contribution: Our method lies in the "gray" area between semi-supervised clustering and classification since it can perform both at a time. In particular if we remove the decision step of the unsupervised learning machine we can perform semi-supervised clustering, or semi-supervised classification otherwise.

The method is based in expressing the supervised and the unsupervised processes using the same framework. In particular, we express both processes as a self-organizing problem derived from the minimization of certain functionals which describes the behavior of both classification processes.

Therefore, it is simple to understand how our method performs semi-supervision, since the labelled data allows the definition of a supervised classifier while the unlabelled data compete in our framework to be part of one of the classes or one of the clusters. Hence, if we want to perform semi-supervised classification we just seed the supervised part of our method with the labelled data and let the unlabelled data adapt. This leads to a fully supervised training set, that can be used later for classification.

However, the fusion of labelled and unlabelled data can be extrapolated to other problem domains. In particular, this structure can be argued to be used in a very overlooked problem, the *particularization* problem (this is the way we will refer to a generalization of several specific processes, namely *adaptation* or *situated learning* in natural language processing or *specification* in the work of Kumar et al. [139] based on region classification in images.)

What is a particularization problem?

Let us imagine a problem of handwritten character recognition, in which we know that all the characters in the document are written by the same author for instance, if we provide an OCR with a document from one writer; an architect drawing a technical plane with its own symbols; a medical application in which we know that the data we need to classify proceeds from a single patient; the problem of face recognition, with a significative data set representing what a face is, and a huge set of data representing what a face is not. In this problem we can consider that not all the non-face data is going to be present in the test set. In all these different application examples, there is a common fact, the knowledge of the fact that our test data set is a *particular subset* of the general training data set.

In human learning theory, a new underdeveloped proposal, *situated learning* considers that "every idea and human action is a generalization, adapted to the ongoing environment" [133]. They also state that "Training by abstraction is of little use; learning occurs by doing. Because current performance will be facilitated to the degree that the context more closely matches prior experience, the most effective training is to act in an apprenticeship relation to others in the performance situation"[134] [135]. Of course, those theories do not really apply to machine learning but to human-human learning transfer. However, this emphasizes the importance of learning in the exact context where the application is going to run. This motivates the research in the line of finding the context of the application given a wide training set. In particular, this approach is very common in *natural language processing*, in which a general language training corpus has to be changed for domain-specific tasks. This task is called *adaptation*. This is usually done by adding specific language corpus to the training set. In [136] a general English language corpus is adapted to medical-domain by mixing three specific medical corpus. Another general approach proposes to make a pre-partition of the space in clusters or trees and weigh each cluster at recognition time to achieve adaptation [137]. Other approaches are based on Bayesian learning and MAP approaches to infer adaptation [138]. On the image domain, the work of Kumar et al. [139] use an EM based refinement of a generative model to infer the particularities of the new data.

Therefore, what is a particularization problem? The particularization problem refers to all those problems in which we can assume the test set to be intrinsically highly correlated and that does not represent the overall training set. In all the aforementioned problems, the test set is highly correlated; all the characters written by the same person are a written in a very similar way with less variation than the all possibilities of writing the same characters. The intra-person variation is smaller than the variation of the overall training set representing the same class. In the medical application, tissues from a single patient are much less variable than the all possible examples of the same tissue. This correlation is the *a priori* knowledge that we want to exploit in this paper.

In order to exploit this knowledge, let us explain how we can take advantage of the particularization and its implications. The assumptions we are making is that the particularized subcluster is a *non-representative subset* of the training data with *small dispersion* that can be modelled by a *simpler* distribution than the overall training set. These assumptions can be summarized saying that our test set is quite compact. We can address this problem in two ways: first, identify the underlying distribution of the test set. To achieve this goal we can fuse a learning process with a prediction oracle, to try to reinforce the prediction by incorporating all possible reliable data from the test data set into the training set. And the other one, to try to exploit the structure underlying in the test set while keeping a general decision rule to guide the process. This last option is the proposed methodology, and subject of this work, the *supervised clustering hybrid competition scheme.*

1.2.3 Intravascular ultrasound image analysis

Myocardial infarction, sudden cardiac death and unstable angina are a consequence of coronary thrombosis developed as a result of a ruptured vulnerable or an eroded atherosclerotic plaque. Plaque rupture or endothelial erosion with subsequent thrombosis formation are the most frequent cause of acute coronary syndromes. As studies have reported a high correlation among multiple plaque ruptures in acute coronary syndrome (ACS) patients. Other studies show that plaque ruptures occur not only in this case but in patients with stable angina or asymptotic ischemia too. Moreover, there are studies showing plaque rupture in patients with non-cardiac death. Hence, it is not clear why some plaque ruptures in coronary arteries of patients lead to severe consequences meanwhile other remain asymptomatic and heal. To understand the mechanisms of plaque destabilization and guide a pharmacological treatment, it is of high interest to image the fragile part of the atheromatous plaque and to differentiate between low-risk and high-risk plaques.

The composition and structure of the vessel change with age, hypertension, dia-

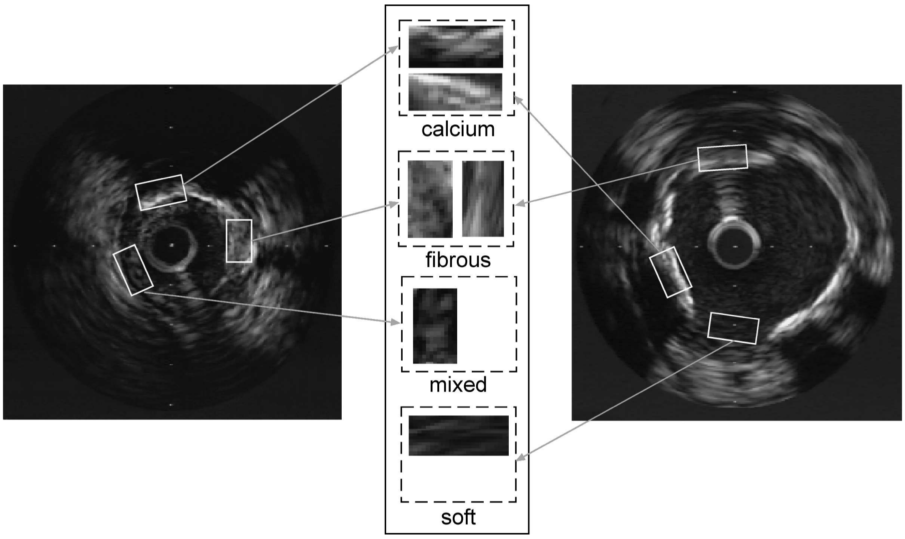


Figure 1.2: Typical IVUS images presenting different kind of tissues.

betes mellitus and many other factors. Until this moment, it is feasible to discriminate different morphological structures of the vessel as calcium deposits, fatty, fatty fibrous and fibrous materials. Although from several decades investigators recognized that noninvasive imaging of coronary calcium might be useful to identify patients with unsuspected coronary artery disease, until the advance of high-resolution techniques little success has been achieved. Today, it is well-known that coronary calcium is a result of a complex, regulated and active process similar to bone formation that is related and at the same time different from atherosclerosis. On the other hand, it is not completely clear what the vulnerable plaque is. The common researcher opinion is that a vulnerable plaque consists of: lipid core, fibrous cap, presence of inflammatory cells and is affected by the vessel remodelling and its 3D morphology. Still a complete morphological, mechanical and chemical information is necessary in order to characterize the vulnerable plaque in a robust way.

Coronary angiography has been so far the gold standard to assess the severity of obstructive luminal narrowing. Furthermore it serves as a decision tool to direct therapeutical procedures (as PTCA). By coronary angiography the lumen boundaries can be assessed, but no information is provided about plaque burden, plaque delineation and plaque components. The predictive power of occurrence of myocardial infarction is rather low since 70% of acute coronary occlusions are in areas that were previously angiographically normal, and only a minority occurs where there was severe stenosis. Other studies have affirmed, that the culprit lesion prior to a myocardial infarction has, in $48\% - 78\%$ of all cases, a stenosis smaller than 50%. The majority of ulcerated plaques are not big enough to be detected by angiography, but can be well assessed pathologically.

IVUS displays the morphology and histological properties of a cross-section of a vessel [1]. Figure 1.2 shows a good example of IVUS images. It is generally accepted that the different kind of plaque tissues distinguishable in IVUS images is threefold: *Calcium formation* is characterized by a very high echo-reflectivity and absorbtion of the emitted pulse from the transducer. This behavior produces a deep shadowing effect behind calcium plaques. *Fibrous plaque* has medium echo-reflectivity resembling that of the adventitia. This tissue has a good transmission coefficient allowing the pulse to travel through the tissue, and therefore, providing a wider range of visualization. *Soft plaque* or *Fibro-Fatty plaque* is the less echo-reflective of the three kind of tissues. It also has good transmission coefficient allowing to see what is behind this kind of plaque. Figure 1.2 shows different examples of the different described plaques.

Due to all the diagnostic possibilities provided by this technique, it is of vital importance for the physicians to address the difficult problem of tissue characterization and IVUS analysis.

Therefore, IVUS analysis is of clinical importance. To illustrate this fact, several researchers have focuses their efforts trying to solve the problem. There are three lines of research to describe the vessel morphology and detect plaques in IVUS images: by textural image analysis, radio-frequency analysis of IVUS data and elastograms.

Textural analysis is the most close to the physician "exercises" during IVUS analysis as a decision is taken on morphological analysis of image sequence. Visual textural analysis is a difficult, subjective and time-consuming process highly depending on the specialist. Therefore, there is an increasing interest of the medical community in developing automatic tissue characterization procedures of IVUS images. This is accentuated because the procedure for tissue classification by physicians implies the manual analysis of IVUS images frequently necessary to be done online during the therapeutical procedure.

The problem of automatic tissue characterization has been widely studied in different medical fields. The unreliability of gray level only methods to achieve good discrimination among the different kind of tissues forced us to use more complex measures, usually based on texture analysis. Texture analysis has played a prominent role in computer vision to solve tissue characterization problems in medical imaging [2] [3] [4] [5] [6] [7] [8] [9].

Several researching groups have reported different approximations to characterize the tissue of intravascular ultrasound image.

Vandenberg et. al., in [10], base their contribution on reducing the noise of the image, to have a clear representation of the tissues. The noise reduction is achieved by averaging sets of images when the least variance in diameter of the IVUS occurs. At the end, a fuzzy logic based expert is set to discriminate among the tissues.

Nailon et. al. devote several efforts to IVUS tissue characterization. In [11] they use classic Haralick texture statistics to discriminate among tissues. In [12] the author proposes the use of co-occurrence matrices texture analysis and fractal texture analysis to characterize intravascular tissue. Thirteen features plus fractal dimension derived from Brownian motion are used for this task. The conclusion shows that fractal dimension is unable to discriminate between calcium and fibrous plaque but helps in fibrous versus lipidic plaque. On the other hand, co-occurrence matrices are well suited for the overall classification. In [13], it is stated that the discriminative power of fractal dimension is poor when trying to separate fibrotic tissue, lipidic tissue and foam cells. The method used is based on fractal dimension estimation techniques (box-counting, brownian motion and frequency domain).

Spencer et. al. in [14], center their work on spectral analysis. Different features are compared: mean power, maximum power, Spectral Slope and 0Hz interception. The work concludes with the 0Hz spectral slope as the most discriminative feature. Dixon et. al. in [15], use co-occurrence matrices and discriminant analysis to evaluate the different kind of tissues in IVUS images. Ahmed et. al. [16] uses a radial transform and correlation for pattern matching. The features used are higher order statistics such as kurtosis, skewness, and up to four order cumulants. The results provided appear to have fairly good visual recognition rate.

The work of de Korte et. al. [17] opens a new proposal based on assessing the local strain of the atherosclerotic vessel wall to identify different plaque components. This line of work is based on estimating the radial strain by performing cross-correlation analysis on pairs of IVUS at a certain intra-coronary pressure. This very promising technique, is called *elastography.*

Probably, one of the most interesting work in this field is the one provided by Zhang et. al. [18]. This work is much more complex trying to evaluate the full morphology of the vessel. Detecting the plaque and adventitia borders and characterizing the different kind of tissues, the tissue discrimination is done using a combination of well-known techniques previously reported in the literature as: co-occurrence matrices and fractal dimension from brownian motion, and adding two more strategies to the amalgam of features: run-length measures and radial profile. The experiments assess the accuracy of the method quantitatively.

Most of the literature found in the tissue characterization matters use texture features, being co-occurrence matrices the most popular of all feature extractors. Further work has been done trying to use other kind of texture feature extractors and IVUS images, and although not specifically centered on tissue characterization, the usage of different texture features in plaque border assessment is reported, that can be easily extrapolated to tissue characterization. In [19], derivative of gaussian, wavelets, co-occurrence matrices, Gabor filters and cumulative moments are evaluated and used to classify blood from plaque. The work highlights the discriminative power of co-occurrence matrices, derivatives of gaussian and cumulative moments. Other works such as [20] provide some hints on how to achieve a fast framework based on local binary patterns and fast high-performance classifiers. This last line of investigation overcomes one of the most significant drawbacks of the texture based tissue characterization systems, the speed, as texture descriptors are inherently slow to be computed.

Whatever method we use in the tissue characterization task, we follow an underlying main methodology. First, we need to extract some features describing the tissue variations. This first step is critical since the features chosen have to be able to describe each kind of tissue in a unique way so that it can not be confused with another one. In this category of feature extraction we should consider the *co-occurrence matrix measures*, *statistical descriptors*, *local binary patterns*, etc. The second step is the classification of the extracted features. Depending on the complexity of the feature data some methods fit better than others. In most cases, high dimensional spaces are generated, so we should consider the use of dimensionality reduction methods such

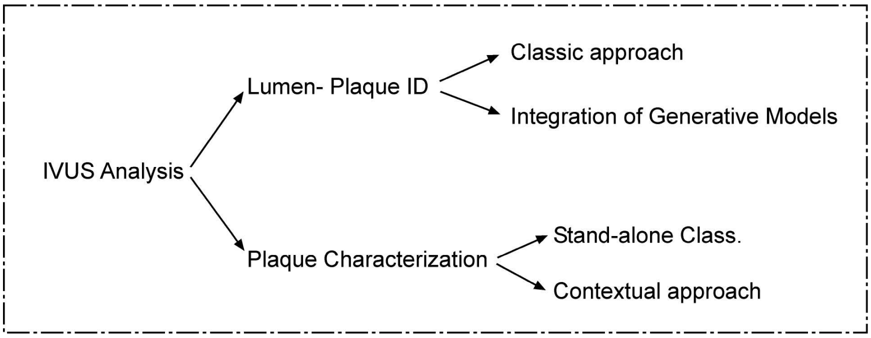


Figure 1.3: PhD working scheme in Medical Image analysis

as *Principal component analysis* or *Fisher linear discriminant analysis.* Either a dimensionality reduction process is needed or not, this step requires a classification procedure. For supervised classification we are using methods like *maximum likelihood*, *nearest neighbors*, *support vector machines*, and the center of our current analysis *adaptative boosting* [41] [42] [50]. In particular, adaptative boosting techniques allow to deal with high dimensional spaces by using an intelligent feature selection process while training the classifier.

Our contribution: In this work we analyze the problem of tissue characterization from a point of view of classical classification using an advanced classification technique: *adaboost.* However, we do not stop the work at this point, and we go further developing a full automatic intravascular ultrasound analysis framework capable of near-real time processing. The framework is able to segment the different layers (intima and adventitia borderlines)and tissues (calcium, fibrous and lipidic) of the IVUS image with high accuracy. Figure 1.3 shows a scheme of the work developed in intravascular ultrasound image analysis.

All the work sketched in the proposals is fully described in the core of this PhD work. The final layout of this work is divided in 6 chapters:

- **Chapter 1** contains this introduction.
- **Chapter 2** gives a general background of the feature spaces and classification techniques that have been used in this work.
- **Chapter 3** covers the basis of *parametric generative deformable models* and the foundations of the *Stop and Go* formulation are unravelled. At the end of the chapter generative snakes and the stop and go formulation are unified.
- **Chapter 4** is related to pattern recognition theory and describes particularization and semi-supervised clustering, introducing *SCHCS: Supervised Clustering Hybrid Competition Scheme.*

- **Chapter 5** provides the experiments and results validating the formerly described techniques.
- **Chapter 6** is devoted to illustrate the work in the medical imaging field. The first half of this chapter is concerned with adaboost and classical classification, and the second half is devoted to build a complete framework for IVUS analysis and tissue characterization.

Chapter 2

General Background: Feature Spaces and Classification Techniques

This chapter describes the set of feature spaces and classification techniques that will be used later in the chapters containing the experiments and IVUS framework definition (chapter 5 and 6 respectively). Since most of the techniques described are only used in chapter 6 for IVUS analysis, we exemplify each technique having in mind mostly ultrasound analysis. In this sense, some assertions are explicitly related with the ultrasound techniques.

2.1 Feature Spaces

The first issue when dealing with complex real problems, such as tissue characterization, is to create a representation of the data we are analyzing. The representation of the data is usually a more compact version of the original samples, for the problem to be analytically feasible. While centering in some aspects of the original samples, we restrict ourselves to that kind of features, and expect them to fully describe the problem; though this not always happens.

Plaque recognition is usually approached as a texture discrimination problem. This line of work is a classical extension of previous works on biological characterization, which also relies on texture features as has been mentioned in the former section. The co-occurrence matrix is the most favored and well known of the texture feature extraction methods due to its discriminative power in this particular problem but it is not the only one nor the fastest method available. In this section, we make a review of different texture methods that can be applied to the problem in particular, from the *co-occurrence matrix measures* method to the most recent texture feature extractor, *local binary patterns*.

To illustrate the texture feature extraction process we have selected a set of techniques basing our criterion of selection on the most widespread methods for tissue characterization and the most discriminative feature extractors reported in the literature [21].

Basically, the different methods of feature extraction emphasize on different fundamental properties of the texture such as scale, statistics or structure. In this way, under the non-elemental statistics property we can find two well-known techniques, co-occurrence methods [22] and higher order statistics represented by moments [23]. Under the label of scale property we should mention methods such as derivatives of gaussian [24], Gabor filters [25] or wavelet techniques [26]. Regarding structure related measures there are methods such as fractal dimension [27] and local binary patterns [28].

To introduce the texture feature extraction methods we divide them into two groups: The first group, that forms the *statistic related methods*, is comprised by co-occurrence matrix measures, accumulation local moments, fractal dimension and local binary patterns. All these methods are somehow related to statistics. Co-occurrence matrix measures are second order measures associated to the probability density function estimation provided by the co-occurrence matrix. Accumulation local moments are directly related to statistics. Fractal dimension is an approximation of the roughness of a texture. Local binary patterns provides a measure of the local inhomogeneity based on an "averaging" process. The second group, that forms the *analytic kernel-based extraction techniques*, comprises Gabor bank of filters, derivatives of gaussian filters and wavelet decomposition. The last three methods are derived from analytic functions and sampled to form a set of filters, each focused on the extraction of a certain feature.

2.1.1 Statistic related methods

Co-occurrence matrix approach

In 1963 Julesz [29] showed the importance of texture segregation using second order statistics. Since then, different tools have been used to exploit this issue. The Gray Level Co-occurrence Matrix is a well-known statistical tool for extracting second-order texture information from images [22]. In the co-occurrence method, the relative frequencies of gray level pairs of pixels at certain relative displacement are computed and sorted in a matrix, the *co-occurrence matrix* **P**. The co-occurrence matrix can be thought of as an estimate of the joint probability density function of gray-level pairs in an image. For G gray levels in the image, **P** will be of size $G \times G$. If G is large, the number of pixel pairs contributing to each element, $p_{i,j}$ in **P** is low, and the statistical significance poor. On the other hand, if the number of gray levels is low, much of the texture information may be lost in the image quantization. The element values in the matrix, when normalized, are bounded by $[0, 1]$, and the sum of all element values is equal to 1.

$$P(i, j, D, \theta) = P(I(l, m) = i \;\; and \;\; I(l + D\cos(\theta), m + D\sin(\theta)) = j)$$

where $I(l, m)$ is the image at pixel (l, m), D is the distance between pixels and θ is the angle. It has been proved by other researchers [30] [21] that the nearest neighbor pairs at distance D at orientations $\theta = \{0^0, 45^0, 90^0, 135^0\}$ are the minimum set needed to

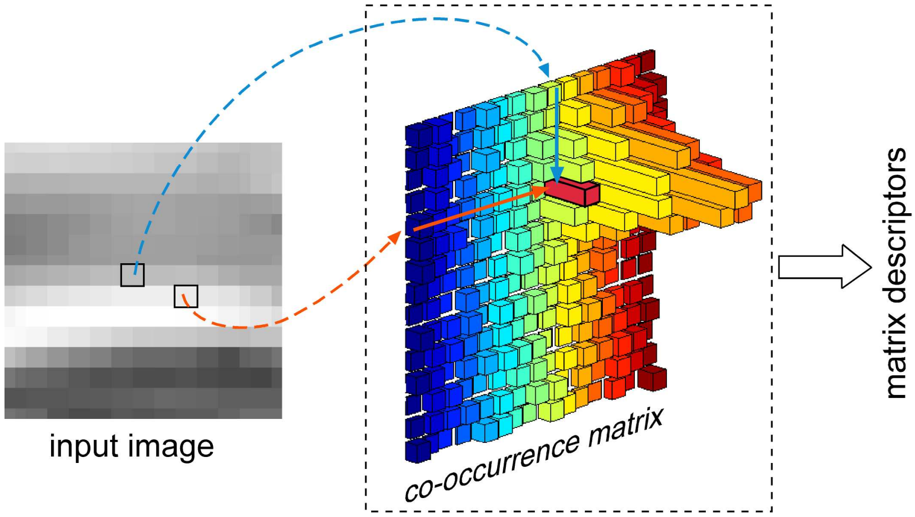


Figure 2.1: Co-occurrence matrix explanation diagram (see text)

describe the texture second-order statistic measures. Figure 2.1 illustrates the method providing a graphical explanation. The main idea is to create a "histogram" of the occurrences of having two pixels of certain gray levels at a determined distance with a fixed angle. Practically, we add one to the cell of the matrix pointed by the gray levels of two pixels (one pixel gray level gives the file and the other the column of the matrix) that fulfil the requirement of being at a certain predefined distance and angle.

Once the matrix is computed several characterizing measures are extracted. Many of these features are derived by weighting each of the matrix element values and then summing these weighted values to form the feature value. The weight applied to each element is based on a feature weighing function, so by varying this function, different texture information can be extracted from the matrix. We present here some of the most important measures that characterize the co-occurrence matrices: Energy, Entropy, Inverse Difference Moment (IDM), Shade, Inertia and Promenance [30]. Let us introduce some notation for the definition of the features:
$P(i,j)$ is the $(i,j)th$ element of a normalized co-occurrence matrix.

$$\begin{aligned}
P_x(i) &= \sum_j P(i,j) \\
P_y(j) &= \sum_i P(i,j) \\
\mu_x &= \sum_i i \sum_j P(i,j) = \sum_i iP_x(i) = E\{i\} \\
\mu_y &= \sum_j j \sum_i P(i,j) = \sum_j jP_y(j) = E\{j\}
\end{aligned}$$

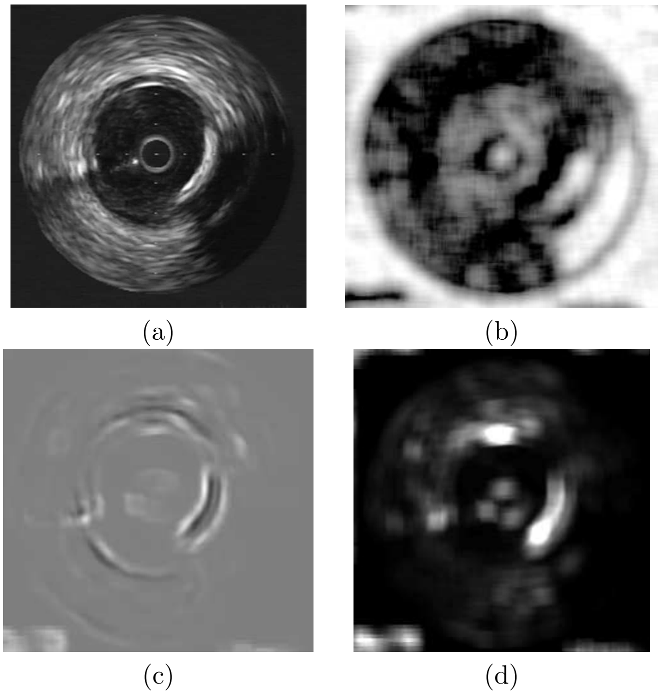

(a) (b) (c) (d)

Figure 2.2: Response of an IVUS image to different measures of the co-occurrence matrix. (a) Original image, (b) Measure shade response, (c) Inverse Different Moment, (d) Inertia.

With the above notation, the features can be written as follows:

$$\begin{aligned}
Energy &= \sum_{i,j} P(i,j)^2 \\
Entropy &= -\sum_{i,j} P(i,j) log P(i,j) \\
IDM &= \sum_{i,j} \frac{1}{1+(i-j)^2} P(i,j) \\
Shade &= \sum_{i,j} (i+j-\mu_x-\mu_y)^3 P(i,j) \\
Inertia &= \sum_{i,j} (i-j)^2 P(i,j) \\
Promenanance &= \sum_{i,j} (i+j-\mu_x-\mu_y)^4 P(i,j)
\end{aligned}$$

Hence, we create a *feature vector* for each of the pixels by assigning each feature measure to a component of the feature vector. Given that we have four different orientations and the six measures for each orientation, the feature vector is a 24-

dimensional vector for each pixel and for each distance. Since we have used two distances $D = 2$ and $D = 3$, the final vector is a 48-dimensional vector.

Figure 2.2 shows responses for different measures on the co-occurrence matrices. Although a straightforward interpretation of the feature extraction response is not easy, some deduction can be made by observing the figures. Figure 2.2(b) shows shade measure, as its name indicates it is related to the shadowed areas in the image, and thus, localizing the shadowing behind the calcium plaque. Figure 2.2(c) shows inverse different moment response, this measure seems to be related to the first derivative of the image, enhancing contours. Figure 2.2(d) depicts the output for the inertia measure, seems to have some relationship with local homogeneity of the image.

Accumulation Local Moments

Geometric moments have been used effectively for texture segmentation in many different application domains [23]. In addition, other kind of moments have been proposed, Zernique moments, Legendre moments, etc. By definition, any set of parameters obtained by projecting an image onto a 2D polynomial basis is called moments. Then, since different sets of polynomials up to the same order define the same subspace, any complete set of moments up to given order can be obtained from any other set of moments up to the same order. The computation of some of these sets of moments leads to very long processing times, so in this section a particular fast computed moment set has been chosen. This set of moments is known as the **accumulation local moments**. Two kind of accumulation local moments can be computed, direct accumulation and reverse accumulation. Since direct accumulation is more sensitive to round off errors and small perturbations in the input data [31], the reverse accumulation moments are recommendable.

The reverse accumulation moment of order $(k-1, l-1)$ of matrix $\mathbf{I}_{ab}$ is the value of $\mathbf{I}_{ab}[1,1]$ after bottom-up accumulating its column k times (i.e., after applying k times the assignment $\mathbf{I}_{ab}[a-i,j] \leftarrow \mathbf{I}_{ab}[a-i,j] + \mathbf{I}_{ab}[a-i+1,j]$, for $i = 0$ to $a-1$, and for $j = 1$ to b), and accumulating the resulting first row from right to left l times (i.e., after applying l times the assignment $\mathbf{I}_{ab}[1, b-j] \leftarrow \mathbf{I}_{ab}[1, b-j] + \mathbf{I}_{ab}[1, b-j+1]$, for $j = 1$ to $b-1$). The reverse accumulation moment matrix is defined so that $\mathbf{R}_{mn}[k.l]$ is the reverse accumulation moment of order $(k-1, l-1)$.

Consider the matrix in the following example:

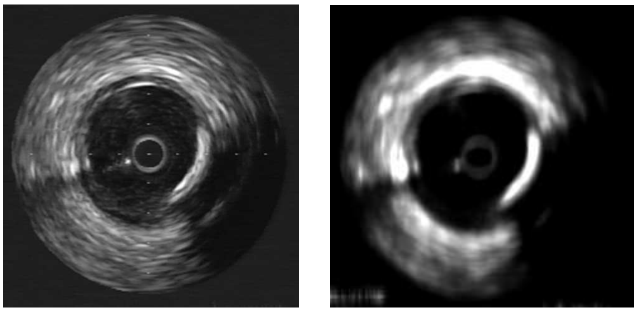
$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \\ 4 & 2 & 3 \end{pmatrix}$$

According to the definition, its reverse accumulation moment of order (1,2) requires two column accumulations,

$$\begin{pmatrix} 5 & 4 & 6 \\ 5 & 3 & 4 \\ 4 & 2 & 3 \end{pmatrix} \rightarrow \begin{pmatrix} 14 & 9 & 13 \\ 9 & 5 & 7 \\ 4 & 2 & 3 \end{pmatrix}$$

and three right to left accumulations of the first row:

$$\begin{pmatrix} 36 & 22 & 13 \end{pmatrix} \rightarrow \begin{pmatrix} 71 & 35 & 13 \end{pmatrix} \rightarrow \begin{pmatrix} 119 & 48 & 13 \end{pmatrix}$$



(a) (b)

Figure 2.3: Accumulation local moments response. (a) Original image. (b) Accumulation local moment of order (3,1).

Then it is said that the reverse accumulation moment of order (1,2) of the former matrix is 119.

The set of moments alone is not sufficient to obtain good texture features in certain images. Some iso-second order texture pairs which are pre-attentively discriminable by humans, would have the same average energy over finite regions. However, their distribution would be different for the different textures. One solution suggested by Caelli is to introduce a nonlinear transducer that maps moments to texture features [32]. Several functions have been proposed in the literature: logistic, sigmoidal, power function or absolute deviation of feature vectors from the mean [23]. The function we have chosen is the hyperbolic tangent function, which is logistic in shape. Using the accumulation moments image I_m, and a non linear operator $|tanh(\sigma(I_m - \bar{I_m})|$ an 'average' is performed throughout the region of interest. The parameter σ controls the shape of the logistic function. Therefore each textural feature will be the result of the application of the non-linear operator to the computed moments. If $n = k \cdot l$ moments are computed over the image, then the dimension of the feature vector will be n. Hence, a n-dimensional point is associated with each pixel of the image.

Figure 2.3 shows the response of moment (3,1) on an IVUS image. In this figure, the response seems to have a smoothing and enhancing effect, clearly resembling diffusion techniques.

Fractal Analysis

Another classic tool for texture description is the fractal analysis [13] [33], characterized by the fractal dimension. We talk roughly about fractal structures when a geometric shape can be subdivided in parts, each of which are approximately a reduced copy of the whole (this property is also referred as self-similarity). The introduction of fractals by Mandelbrot [27] allowed a characterization of complex structures that could not be described by a single measure using Euclidean geometry. This measure is the *fractal dimension*, which is related to the degree of irregularity of the surface texture.

The fractal structures can be divided into two subclasses: the deterministic fractals and the random fractals. Deterministic fractals are strictly self-similar, that is, they appear identical over a range of magnification scales. On the other hand, random fractals are statistical self-similar. The similarity between two scales of the fractal is ruled by a statistical relationship.

The fractal dimension represents the disorder of an object. The higher the dimension the more complex the object is. Contrary to the Euclidian dimension, the fractal dimension is not constrained to integer dimensions.

The concept of fractals can be easily extrapolated to image analysis if we consider the image as a 3D surface in which the height at each point is given by the gray value of the pixel.

Different approaches have been proposed to compute the fractal dimension of an object. Here we only consider three classical approaches: based on box-counting, Brownian motion and Fourier analysis.

Box-counting. The box-counting method is an approximation to the fractal dimension as it is conceptually related to self-similarity.

In the method the object to be evaluated is placed on a square mesh of various sizes, r. The number of mesh boxes, N, that contain any part of the fractal structure are counted.

It has been proved that in a self-similar structures there is a relationship between the reduction factor r and the number of divisions N into which the structure can be divided:

$$Nr^D = 1$$

where D is the self-similarity dimension. Therefore, the fractal dimension can be easily written as:

$$D = \frac{\log N}{\log 1/r}$$

This process is done at various scales by altering the square size r. Therefore, the box-counting dimension is the slope of the regression line that better approximates the data on the plot produced by $\log N \times \log 1/r$.

Fractal dimension from brownian motion. The fractal dimension is found by considering the absolute intensity difference of pixel pairs, $I(p_1) - I(p_2)$, at different scales. It can be shown that for a fractal Brownian surface the following relationship must be satisfied:

$$E(|I(p_1) - I(p_2)|)\alpha(\sqrt{(x_2 - x_1) + (y_2 - y_1)})^H$$

where E is the mean and H the Hurst coefficient. The fractal dimension is related to H in the following way $D = 3 - H$. In the same way than the former method for calculating the fractal dimension the mean difference of intensities is calculated for different scales (each scale given by the euclidian distance between two pixels), and the slope of the regression line between $\log E(|I(p_1) - I(p_2)|)$ and $\sqrt{(x2 - x1) + (y2 - y1)}$ gives the Hurst parameter.

Triangular prism surface area method. The triangular prism surface area (TPSA) algorithm considers an approximation of the 'area' of the fractal structure using triangular prisms. If a rectangular neighborhood is defined by its vertices A, B,

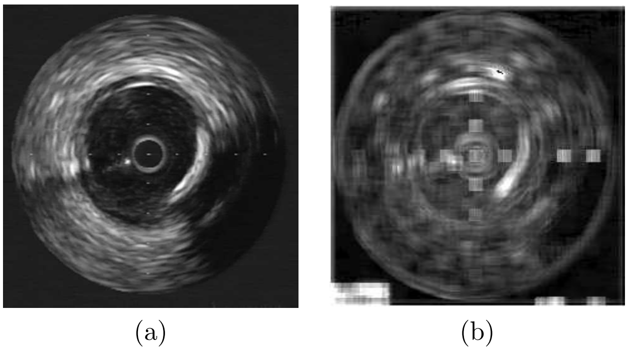


Figure 2.4: Fractal dimension from box-counting response. (a) Original image. (b) Fractal dimension response with neighborhoods of 10 × 10.

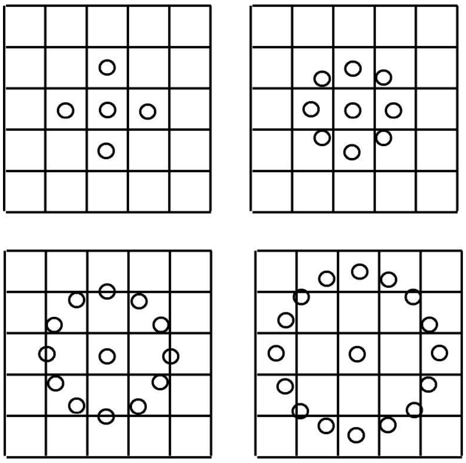


Figure 2.5: Typical neighbors (Top-Left) $P = 4$, $R = 1.0$ (Top-Right) $P = 8$, $R = 1.0$ (Bottom-Left) $P = 12$, $R = 1.5$ (Bottom-Right) $P = 16$, $R = 2.0$.

C and D, the area of this neighborhood is calculated by tessellating the surface with four triangles defined for each consecutive vertex and the center of the neighborhood.

The area of all triangles for every central pixel is summed up to the entire area for different scales. The double logarithmic Richardson-Mandelbrot plot should again yield a linear line whose slope is used to determine the TPSA dimension. Figure 2.4 shows the fractal dimension value of each pixel of an IVUS image considering the fractal dimension of a neighborhood around the pixel. The size of the neighborhood is 10 × 10. The response of this technique seems to take into account the border information of the structures in the image.

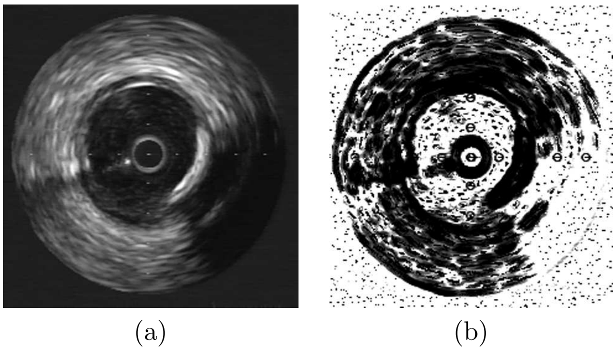


Figure 2.6: Local Binary Pattern response. (a) Original image. (b) Local Binary Pattern output with parameters $R = 3$,$P = 24$.

Local Binary Patterns

Local Binary Patterns [28] are a feature extraction operator used for detecting "uniform" local binary patterns at circular neighborhoods of any quantization of the angular space and at any spatial resolution. The operator is derived based on a circularly symmetric neighbor set of P members on a circle of radius R. It is denoted by $LBP_{P,R}^{riu2}$. Parameter P controls the quantization of the angular space, and R determines the spatial resolution of the operator. Figure 2.5 shows typical neighborhood sets. To achieve gray-scale invariance, the gray value of the center pixel (g_c) is subtracted from the gray values of the circularly symmetric neighborhood g_p ($p = 0, 1, ..., P - 1$) and assigned an 1 value if the difference is positive and 0 if negative.

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

By assigning a binomial factor 2^p for each value obtained, we transform the neighborhood into a single value. This value is the $LBP_{R,P}$:

$$LBP_{R,P} = \sum_{p=0}^{P} s(g_p - g_c) \cdot 2^p$$

To achieve rotation invariance the pattern set is rotated as many times as necessary to achieve a maximal number of the most significant bits, starting always from the same pixel. The last stage of the operator consists on keeping the information of "uniform" patterns while filtering the rest. This is achieved using a transition count function U. U is a function which counts the number of transitions 0/1, 1/0 while we move over the neighborhood:

$$\begin{aligned} U(LBP_{P,R}) = {} & |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \\ & \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \end{aligned}$$

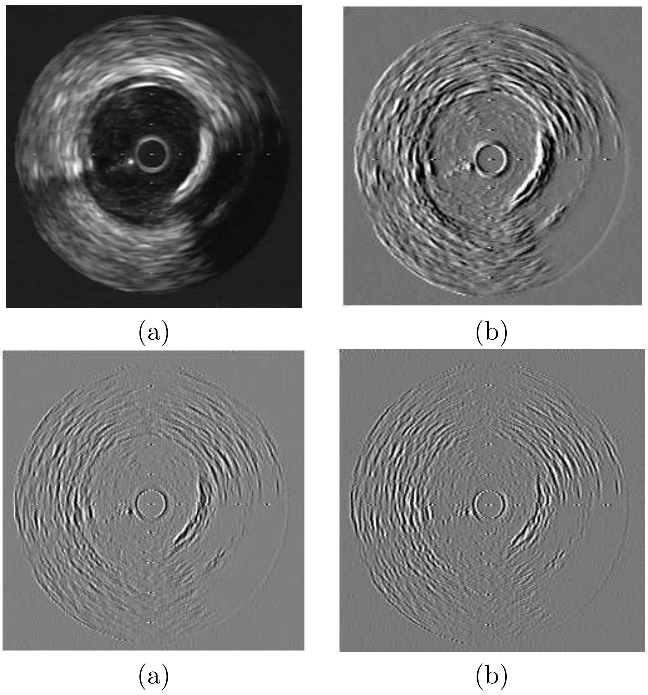


Figure 2.7: Derivative of Gaussian responses for $\sigma = 2$. (a) Original image. (b) First derivative of Gaussian response (c) Second derivative of Gaussian response (d) Third derivative of Gaussian response.

Therefore,

$$LBP_{P,R}^{riu2} = \begin{cases} LBP_{P,R}^{ri} & \text{if } U(LBP_{P,R}) \leq 2 \\ P+1 & \text{otherwise.} \end{cases}$$

Figure 2.6 shows an example of an IVUS image filtered using a Uniform Rotation Invariant Local Binary Pattern with values $P = 24$, $R = 3$. The feature extraction image displayed in the figure looks like a discrete response focussed on the structure shape and homogeneity.

2.1.2 Analytic kernel-based methods

Derivatives of Gaussian

In order to handle image structures at different scales in a consistent manner, a linear *scale-space representation* is proposed in [24], [34]. The basic idea is to embed the original signal into an one-parameter family of gradually smoothed signals, in which fine scale details are successively suppressed. It can be shown that the Gaussian kernel and its derivatives are one of the possible smoothing kernels for such scale-space. The Gaussian kernel is well-suited for defining a space-scale because of its linearity and spatial shift invariance, and the notion that structures at coarse scales should be related to structures at finer scales in a well-behaved manner (new structures are not created by the smoothing method). Scale-space representation is a special type of multi-scale representation that comprises a continuous scale parameter and preserves the same spatial sampling at all scales. Formally, the linear-space representation of a continuous signal is constructed as follows. Let $f : \Re^N \rightarrow \Re$ represent any given signal. Then, the scale-space representation $L : \Re^N \times R_+ \rightarrow \Re$ is defined by $L(\cdot;0) = f$ so that:

$$L(\cdot;t) = g(\cdot;t) * f$$

where $t \in \Re_+$ is the scale parameter, and $g : \Re^N x R_+ \{0\} \rightarrow \Re$ is the Gaussian kernel. In arbitrary dimensions, it is written as:

$$g(x;t) = \frac{1}{(2\pi t)^{N/2}} e^{-x^T x/(2t)} = \frac{1}{(2\pi t)^{N/2}} e^{-\sum_{i=1}^{N} x_i^2/(2t)} \qquad x \in Re^N, x_i \in \Re$$

The square root of the scale parameter, $\sigma = \sqrt(t)$, is the standard deviation of the kernel g, and is a natural measure of spatial scale in the smoothed signal at scale t. From this scale-space representation, multi-scale spatial derivatives can be defined by:

$$L_{x^n}(\cdot;t) = \partial_{x^n} L(\cdot;t) = g_{x^n}(\cdot;t) * f,$$

where g_{x^n} denotes a derivative of some order n.
The main idea behind the construction of this scale-space representation is that the fine scale information should be suppressed with increasing values of the scale parameter. Intuitively, when convolving a signal by a Gaussian kernel with standard deviation $\sigma = \sqrt{t}$, the effect of this operation is to suppress most of the structures in the signal with a characteristic length less than σ. Different directional derivatives can be used to extract different kind of structural features at different scales. It is shown in the literature [35] that a possible complete set of directional derivatives up to the third order are $\partial^n = [\partial_0, \partial_{90}, \partial_0^2, \partial_{60}^2, \partial_{120}^2, \partial_0^3, \partial_{45}^3, \partial_{90}^3, \partial_{135}^3]$. So our feature vector will consist on the directional derivatives, including the zero-derivative, for each of the n scales desired:

$$F = \{\{\partial^n, G^n\}, , n \in \Re\}$$

Figure 2.7 shows some of the responses for the DOG bank of filters for $\sigma = 2$. Figure 2.7(b), (c) and (d) display the first, second and third derivatives of gaussian, respectively.

Wavelets

Wavelets come to light as a tool to study non-stationary problems [36]. Wavelets perform a decomposition of a function as a sum of local bases with finite support and localized at different scales. Wavelets are characterized for being bounded functions with zero average. This implies that the shapes of these functions are waves restricted in time. Their time-frequency limitation yields a good location. So a wavelet ψ is a function of zero average:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0$$

which is dilated with a scale parameter s and translated by u:

$$\varphi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-u}{s})$$

The wavelet transform of f at scale s and position u is computed by correlating f with a wavelet atom:

$$W_f(u,s) = \int_{-\infty}^{+\infty} f(t)\frac{1}{\sqrt{(s)}}\psi^*(\frac{t-u}{s})dt \tag{2.1}$$

The continuous wavelet transform $W_f(u,s)$ is a two-dimensional representation of a one-dimensional signal f. This indicates the existence of some redundancy that can be reduced and even removed by sub-sampling the parameters of these transforms. Completely eliminating the redundancy is equivalent to building a basis of the signal space.

The decomposition of a signal gives a series of coefficients representing the signal in terms of the base from a mother wavelet, that is the projection of the signal on the space formed by the base functions.

The continuous wavelet transform has two major drawbacks: the first, stated formerly, is redundancy and the second, impossibility to calculate it unless a discrete version is used. A way to discretize the dilation parameter is $a = a_0^m, m \in Z, a_0 \neq 1$ constant. Thus, we get a series of wavelets ψ_m of width, a_0^m. Usually, we take $a_0 > 1$, although it is not important because m can be positive or negative. Often, a value of $a_0 = 2$ is taken. For $m = 0$ we make s to be the only integer multiples of a new constant s_0. This constant is chosen in such a way that the translations of the mother wavelet, $\psi(t-ns_0)$, are as close as possible in order to cover the whole real line. Then, the election of s level is as follows:

$$\psi_m, n(t) = a_0^{-m/2}\psi(\frac{t-ns_0a^m}{a_0^m}) = a_0^{-m/2}\psi(a_0^{-m}t - ns_0)$$

that covers the entire real axis as well as the translations $\psi(t-ns_0)$ does. Summarizing, the discrete wavelet transform consists of two discretizations in the transformation equation (2.1),

$$a = a_0^m, \qquad b = nb_0a_0^m, \qquad m,n \in Z, \qquad a_0 > 1, b_0 > 0$$

The multi-resolution analysis tries to build orthonormal bases for a dyadic grid, where $a_0 = 2, b_0 = 1$, which besides have a compact support region. Finally, we can imagine the coefficients $d_{m,n}$ of the discrete wavelet transform as the sampling of the convolution of signal $f(t)$ with different filters $\psi_m(-t)$, where $\psi_m(t) = a_0^{-m/2}\psi(a^{-m}t)$

$$y_m(t) = \int f(s)\psi_m(s-t)ds \qquad d_{m,n} = y_m(na_0^m)$$

Figure 2.8 shows the dual effect of shrinking of the mother wavelet as the frequency increases, and the translation value decreasing as the frequency increases. The mother wavelet keeps its shape but if high frequency analysis is desired the spatial support

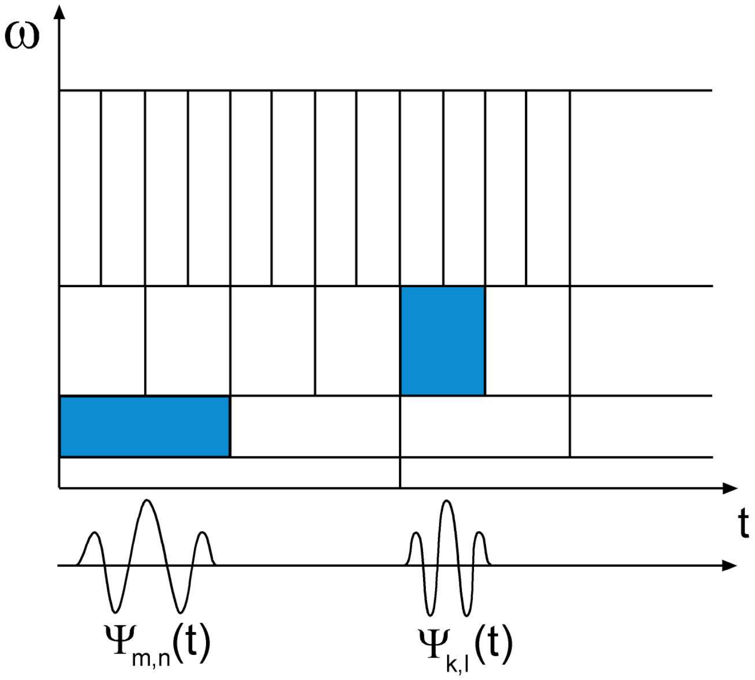


Figure 2.8: Scale-frequency domain of wavelets.

of the wavelet has to decrease. On the other hand, if the whole real line has to be covered by translations of the mother wavelet, as the spatial support of the wavelet decreases, the number of translations needed to cover the real line increases. Unlike Fourier transform, where translations of analysis are at the same distance for all the frequencies.

The choice of a representation of the wavelet transform leads us to define the concept of a *frame*. A frame is a complete set of functions, that, though able to span $L^2(\Re)$, it is not a base because it lacks the property of linear independence. Multi-resolution analysis (MRA) proposed in [26] is another representation in which the signal is decomposed in an approximation at a certain level L with L detail terms of higher resolutions. The representation is an orthonormal decomposition instead of a redundant frame and therefore the number of samples that defines a signal is the same that the number of coefficients of their transform. A multi-resolution analysis consists of a sequence of function subspaces of successive approximation. Let P_j be an operator defined as the orthonormal projection of functions of L^2 over the space V_j. The projection of a function f over V_j is a new function that can be expressed as a linear combination of the functions that form the orthonormal base of V_j. Coefficients of the combination of each base function is the scalar product of f with the base

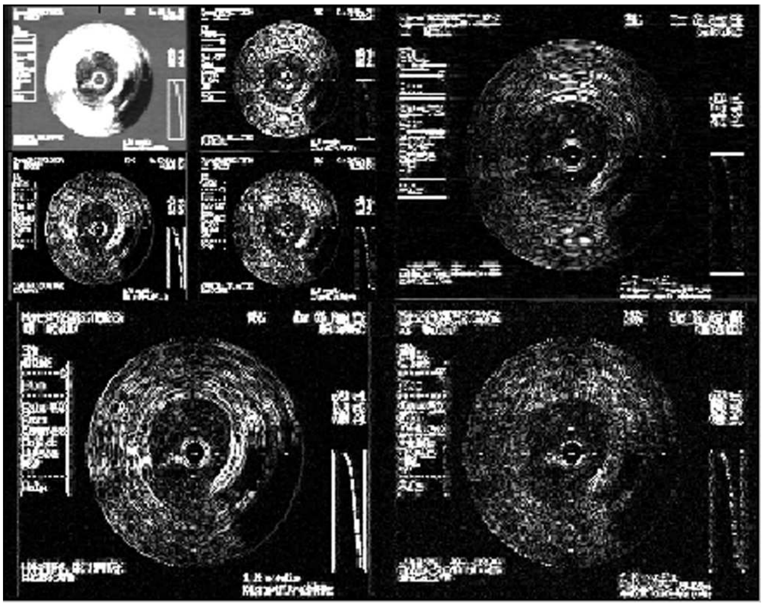


Figure 2.9: Wavelets multi-resolution decomposition.

functions:

$$P_j f = \sum_{n \in Z} \langle f, \phi_{j,n} \rangle \phi_{j,n}$$

where

$$\langle f, g \rangle = \int_{-\infty}^{+\infty} f(t) g(t) dt$$

Before, we have pointed out the nesting condition of the V_j spaces, $V_j \subset V_{j-1}$. Now, if $f \in V_{j-1}$ then $f \in V_j$ or f is orthonormal to all the V_j functions, that is, we divide V_{j-1} in two disjoint parts: V_j and other space W_j, such that if $f \in V_j, g \in W_j, f \perp g$; W_j is the orthonormal complement of V_j in V_{j-1}:

$$V_{j-1} = V_j \oplus W_j$$

where symbol $\oplus$ measures the addition of orthonormal spaces. Applying the former equation and the completeness condition, then

$$\ldots \oplus W_{j-2} \oplus W_{j-1} \oplus W_j \oplus W_{j+1} \oplus \ldots = \bigoplus_{j \in Z} W_j = L^2$$

So, we can write:

$$P_{j-1} f = P_j f + \sum_{n \in Z} \langle f, \psi_{j,n} \rangle \psi_{j,n}$$

From these equations some conclusions can be extracted. First, the projection of a signal f in a space V_j gives a new signal $P_j f$, an approximation of the initial signal.

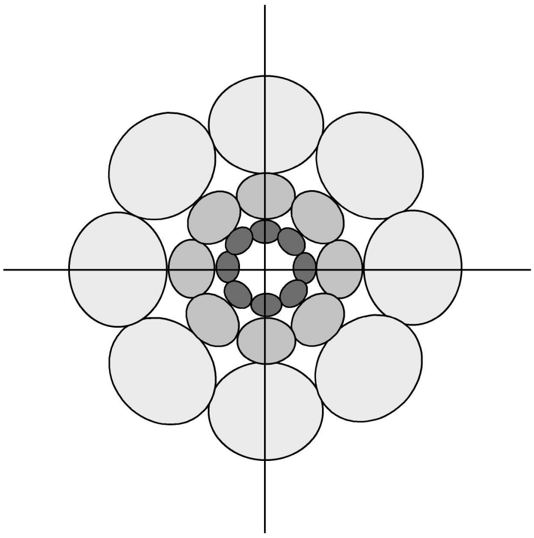


Figure 2.10: The filter set in the spatial-frequency domain.

Secondly, we have a hierarchy of spaces, then $P_{j-1}f$ will be a better *approximation* (more reliable) than $P_j f$. Since V_{j-1} can be divided in two subspaces V_j and W_j, if V_j is an approximation space then W_j, which is the complementary orthonormal space, it is the *detail* space. The less the j, the finer the details.

$$\begin{aligned} V_j &= V_{j+1} \oplus W_{j+1} = V_{j+2} \oplus W_{j+2} = \dots \\ &= V_L \oplus W_L \oplus W_{L-1} \oplus \dots \oplus W_{j+1} \end{aligned}$$

This can be viewed as a decomposition tree (see figure 2.9). At the top left side of the image the approximation can be seen, and surrounding it the successive details. The further the detail is located the finer the information provided. So, the details at the bottom and at the right side of the image have information about the finer details and the smallest structures of the image decomposed. Therefore, we have a feature vector composed by the different detail approaches and the approximation for each of the pixels.

Gabor's filter bank

Gabor filters represent another multi-resolution technique that relies on scale and direction of the contours [25] [37]. The Gabor filter consists of a two dimensional sinusoidal plane wave of a certain orientation and frequency that is modulated in amplitude by a two-dimensional Gaussian envelope. The spatial representation of the Gabor filter is as follows:

$$h(x,y) = exp\big\{ -\frac{1}{2}[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}]\big\} \cos(2\pi u_0 x + \phi)$$

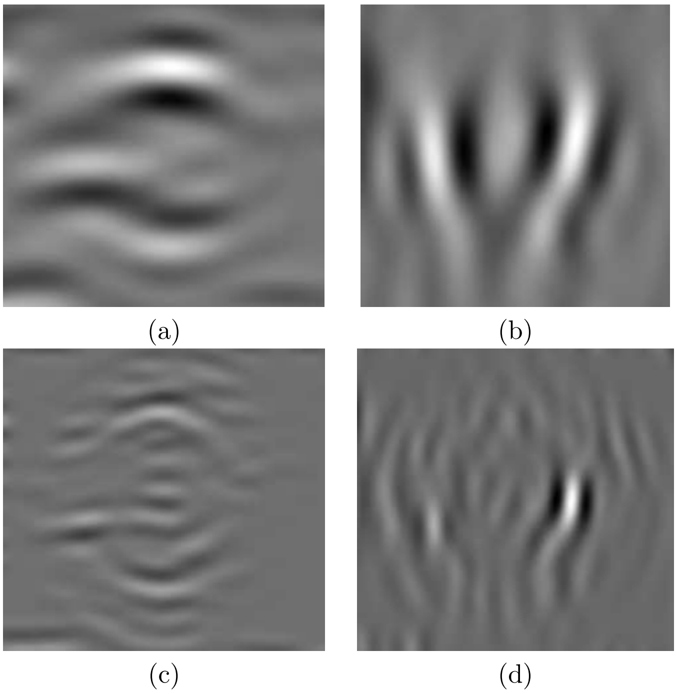


Figure 2.11: Gabor filter bank example responses. (a) Gabor vertical energy of a coarse filter response. (b) Gabor horizontal energy of a coarse filter response. (c) Gabor vertical energy of a detail filter response. (d) Gabor horizontal energy of a detail filter response.

where u_0 and ϕ are the frequency and phase of the sinusoidal plane wave along the x-axis and σ_x and σ_y are the space constants of the Gaussian envelope along the x- and y-axis respectively. Filters at different orientations can be created by rigid rotation of x-y coordinate system.

An interesting property of this kind of filters is its frequency and orientation-selection. This fact is better displayed in the frequency domain. Figure 2.10 shows the filter area in the frequency domain. We can observe that each of the filters has a certain domain defined by each of the leaves of the Gabor 'rose'. Thus, each filter responds to a certain orientation and at a certain detail level. Wider the range of orientations, smaller the space filter dimensions and smaller the details captured by the filter, as bandwidth in the frequency domain is inversely related to filter scope in the space domain. Therefore, Gabor filters provide a trade-off between localization or resolution in both the spatial and the spatial -frequential domains. As it has been mentioned, different filters emerge from rotating the x-y coordinate system. For practical approaches one can use four angles $\theta_0 = 0^o, 45^o, 90^o, 135^o$. For an image array of N pixels (with N power of 2), the following values of u_0 are suggested [25] [37]:

$$1\sqrt{2}, 2\sqrt{2}, 3\sqrt{2}, \ldots, \text{and } (N_c/4)\sqrt{2}$$

cycles per image-width. Therefore, the orientations and bandwidth of such filters vary with 45^o and 1 octave. These parameters are chosen due to the fact that there is physiologic evidences of frequency bandwidth of simple cells in visual cortex being of about 1 octave, and Gabor filters try to mimic part of the human perceptual system.

The Gabor function is an approximation to a wavelet. However, though admissible, it does not result in an orthogonal decomposition and, therefore, a transformation based on Gabor's filters is redundant. On the other hand, Gabor filtering is designed to be nearly orthogonal, reducing the amount of overlap between filters.

Figure 2.11 shows different responses for different filters of the spectrum. Figures 2.11(a) and 2.11(b) correspond to the inner filters with reduced frequency bandwidth displayed in figure 2.10. It can be seen that they only deliver coarse information of the structure and the borders are far from the original location. In the same way, figures 2.11(c) and 2.11(d) are filters located on a further ring, and therefore respond to details in the image.

It can be observed that the feature extraction process is a transformation of the original two-dimensional image domain to a feature space that probably will have different dimensions. In some cases, the feature space remains low, as in fractal dimension and local binary patterns, that with very few features try to describe the texture present in the image. However, several feature spaces require higher dimensions, as: accumulation local moments, co-occurrence matrix measures or derivatives of gaussian. Table 2.1 shows the dimensionality of the different spaces generated by the feature extraction process in our texture-based IVUS analysis.

Method	Space dimension
Co-occurrence matrix measures	48
Accumulation local moments	81
Fractal Dimension	1
Local Binary Patterns	3
Derivative of Gaussian	60
Wavelets	31
Gabor's filters	20

Table 2.1: Dimensionality of the feature space provided by the texture feature extraction process.

The next step after the feature extraction is the classification process. As a result of the disparity of the dimensionality of the feature spaces, we have to choose a classification scheme able to deal with high dimensionality feature data.

2.2 Classification process

Once completed the feature extraction process, we have a set of features disposed in feature vectors. Each feature vector is composed by all the feature measures computed at each pixel. Therefore, for each pixel we have an n-dimensional point in the feature space, where n is the number of features. This set of data is the input to the classification process. The classification process is divided in two main categories:

supervised and unsupervised learning. While supervised learning is based on a set of examples of each class that trains the classification process, the unsupervised learning is based on the geometry position of the data in the feature space and its possibility to be grouped in clusters.

In this section, we are mainly concerned in supervised learning and classification, since we know exactly what classes we are seeking. Supervised classification techniques are usually divided in parametric and non-parametric. Parametric techniques rely on knowledge of the probability density function of each class. On the contrary, non-parametric classification, does not need the probability density function and is based on the geometrical arrangement of the points in the input space. We begin describing a non-parametric technique, *k-nearest neighbors*, that will serve later as a ground truth to verify the discriminability of the different feature spaces. Since non-parametric techniques have high computational cost, we make some assumptions that lead to describe *maximum likelihood* classification techniques. However, the last techniques are very sensitive to the input space dimension. It has been shown in the former section that some feature spaces cast the two-dimensional image data to high dimensional spaces. In order to deal with high dimensional data, a dimensionality reduction is needed. The dimensionality reduction techniques are useful to create a meaningful set of data because the feature space is usually large in comparison to the number of samples retrieved. The most known technique for dimensionality reduction is *principal component analysis* [38]. However, PCA is susceptible to errors depending on the arrangement of the data points in the training space, due to the fact that it does not consider the different distributions of data clusters. In order to solve the deficiency of PCA in discrimination matters, *Fisher linear discriminant analysis* is introduced [38] [40]. In order to try to improve the classification rate of simple classifiers, combination of classifiers is proposed. One of the most important classification assembling process is *boosting.* The last part of this section is devoted to a particular class of boosting techniques, *Adaptative Boosting (AdaBoost)* [41] [42].

2.2.1 k-Nearest Neighbors

Voting k-Nearest Neighbors classification procedure is a very popular classification scheme which does not rely on any assumption concerning the structure of the underlying density function.

As any non-parametric technique, the resulting classification error is the smallest achievable error given a set of data. This is true due to the fact that this technique implicitly estimates the density function of the data, and therefore, the classifier becomes the *Bayes classifier* if the density estimates converge to the true densities when an infinite number of samples are used [38].

In order to classify a test sample X, the k nearest neighbors to the test sample are selected from the overall training data, and the number of neighbors from each class ω_i among the k selected samples is counted. The test sample is then classified to the class represented by a majority of the k nearest neighbors. That is:

$$\mathbf{k}_j = \max\{\mathbf{k}_1 \cdots \mathbf{k}_L\} \rightarrow X \in \omega_j$$

$$\mathbf{k}_1 + \cdots + \mathbf{k}_L = k$$

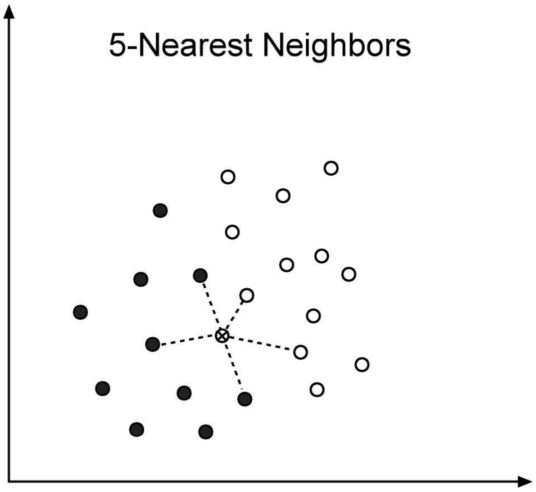


Figure 2.12: 5-Nearest Neighbors example.

where $\mathbf{k}_j$ is the number of neighbors from class $\omega_j, (j = 1, \cdots, L)$ among the selected neighbors. Usually, the same metric is used to measure the distance to samples of each class.

Figure 2.12 shows an example of a 5-nearest neighbors process. Sample X will be classified as member of the light gray class since there are 3 nearest neighbors of the black class while there are only 2 members of the white class.

2.2.2 Maximum Likelihood

The maximum likelihood classifier is one of the most popular methods of classification [39]. The goal is to assign the most likely class w_j, from a set of N classes $w_1, \ldots, w_N$, to each feature vector. The most likely class w_j from a given feature vector $\mathbf{x}$ is the one with maximum posterior probability of belonging to the class $P(w_j|\mathbf{x})$. Using the Bayes' theorem, we have:

$$P(w_j|\mathbf{x}) = \frac{P(\mathbf{x}|w_j)P(w_j)}{P(\mathbf{x})}$$

On the left side of the equation, there is the *a posteriori* probability of a feature vector $\mathbf{x}$ to belong to the class w_j. On the right side, the *a priori* probability $P(\mathbf{x}|w_j)$ that expresses the probability of the feature vector $\mathbf{x}$ being generated by the probability density function of w_j. $P(\mathbf{x})$ and $P(w_j)$ are the *a priori* probability of appearance of feature vector $\mathbf{x}$ and the probability of appearance of each class w_j respectively.

This model relies on the knowledge of the probability density function underlying each of the classes, as well as the probability of occurrence of the data and the classes. In order to reduce the complexity of such estimations, some assumptions are made. The first assumption generally made is the equiprobability of appearance for each of the feature vector as well as for each of the classes. This assumption reduces the

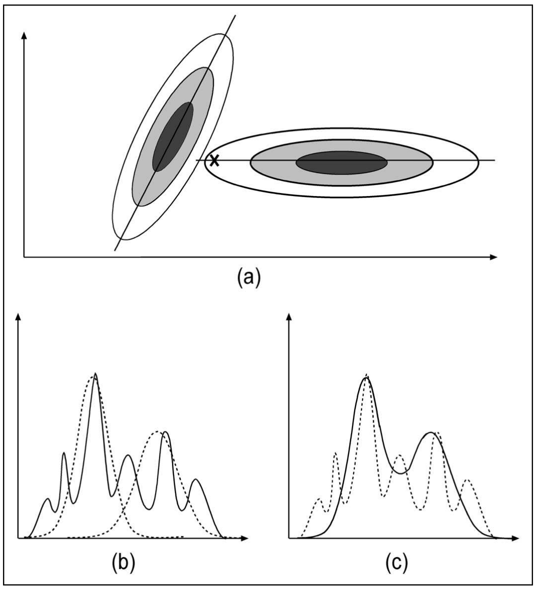


Figure 2.13: (a) Graphic example of the ML classification assuming an underlying density model. (b) Unknown probability density function estimation by means of a 2 gaussian mixture model. (c) Resulting approximation of the unknown density function.

Bayes' theorem to estimate the probability density function for each class:

$$P(w_j|\mathbf{x}) = P(\mathbf{x}|w_j)$$

Multiple methods can be used to estimate the *a priori* probability. Two of the most widespread methods are the assumption of a certain behavior and the mixture models.

A very common hypothesis is to identify the underlying probability density function with a multivariate normal distribution. In that case the likelihood value is:

$$P(\mathbf{x}|w_j) = \frac{1}{\sqrt{\Sigma_j}(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu_j)\Sigma_j^{-1}(\mathbf{x}-\mu_j)^T}$$

where Σ_j and μ_j are the covariance matrix and the mean value for class j, respectively. In the case where the determinants of the covariance matrix for each of the classes are equal each other, the likelihood value becomes the same as the Mahalanobis distances. Figure 2.13(a) shows an example of the effect of this kind of classifier on a sample "X". Although the sample seems to be nearer the left hand distribution in terms of Euclidean distance, it is assigned to the class on the right hand since the probability of generating the sample is higher than its counterpart.

The other approach is to estimate the model of the probability density function. In the mixture model approach, we assume that the probability density function can be modelled using an ensemble of simple known distributions. If the base distribution is the gaussian function it is called Gaussian Mixture Model. The interest in this method consists of the estimation of complex density function using low-level statistics.

The mixture model is composed of a sum of fundamental distributions, following the next expression:

$$p_i(x|\Theta) = \sum_{k=1}^{C} p_k(x|\theta_k)P_k \tag{2.2}$$

where C is the number of mixture components, P_k is the *a priori* probability of the component k, and θ_k represents the unknown mixture parameters. In our case, we have chosen **gaussian mixture models** $\theta_k = \{P_k, \mu_k, \sigma_k\}$ for each set of texture data we want to model. Figure 2.13(b) and figure 2.13(c) show an approximation of a probability density function with a mixture of two gaussian and the resulting approximation. Figure 2.13(b) shows the function to be estimated as a continuous line and the gaussian functions used for the approximation as a dotted line. Figure 2.13(c) shows the resulting approximated function as a continuous line and the function to be estimated as a dotted line as a reference. One can observe that with a determined mixture of gaussian distributions, an unknown probability density function can be well approximated. The main problem of this kind of approaches resides in its computational cost and the unknown number of base functions needed, as well as the value of their governing parameters. In order to estimate the parameters of each base distribution, general maximization methods are used, such as Expectation Maximization (EM) algorithm [39].

However, this kind of techniques is not very suitable as the number of dimensions is large and the training data samples size is small. Therefore, a process of dimensionality reduction is needed to achieve a set of meaningful data. Principal component analysis and Fisher linear discriminant analysis are the most popular dimensionality reduction techniques used in the literature.

2.2.3 Feature data dimensionality reduction

Principal Component Analysis

This method is also known as *Karhunen-Loeve* method [38]. Component Analysis seeks directions or axes in the feature space that provide an improved, lower-dimensional representation of the full data space. The method chooses a dimensionality reducing linear projection that maximizes the scatter of all projected samples. Let us consider a set of M samples $\{x_1, x_2, \ldots, x_M\}$ in an n-dimensional space. We also consider a linear transformation that maps the original space in a lower dimensional space (of dimension m, $m < n$). The new feature vectors y are defined in the following way:

$$y_k = W^T x_k \ , \quad k = 1, \ldots, M$$

where W is a matrix with orthonormal columns. The total scatter matrix S_T is defined as:

$$S_T = \sum_{k=1}^{M} (x_k - \mu)(x_k - \mu)^T$$

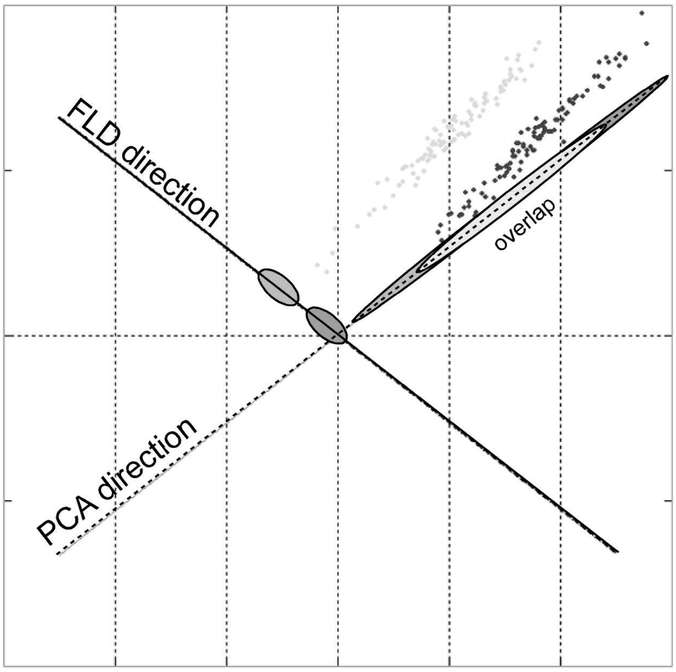


Figure 2.14: Example of the resulting direction using PCA and FLD.

where M is the number of samples, and μ is the mean vector of all samples. Applying the linear transformation W^T, the scatter of the transformed feature vectors is $W^T S_T W$. PCA is defined as to maximize the determinant of the scatter of the transformed feature vectors:

$$W_{opt} = \text{argmax}|W^T S_T W| = [w_1 w_2 \dots w_m]$$

where $\{w_i | i = 1, 2, \dots, m\}$ is the set of n-dimensional eigenvectors of S_T corresponding to the m largest eigenvalues.

Therefore, PCA seeks the directions of maximum scatter of the input data, which correspond to the eigenvectors of the covariance matrix having the largest eigenvalues. The n-dimensional mean vector μ and the $n \times n$ covariance matrix Σ are computed for the full data set.

In summary, the eigenvectors and eigenvalues are computed and sorted in decreasing order. The k eigenvectors having the largest eigenvalues are chosen. With those vectors a $n \times m$ matrix W_{opt} is built. This transformation matrix defines an m dimensional subspace. Therefore, the representation of the data onto this m-dimensional space is:

$$y = A^t(x - \mu)$$

Principal component analysis is a general method to find the directions of maximum scatter of the set of samples. This fact however does not ensure that such directions will be optimal for classification. In fact, it is well-known that some specific distributions of the samples of the classes result in projection directions that deteriorate the discriminability of the data. This effect is shown in figure 2.14 in which the loss of information when projecting to the PCA direction clearly hinders the discrimination process. Note that both projections of the clusters on the PCA subspace overlap.

Fisher Linear Discriminant Analysis

A classical approach to find a linear transformation that discriminates the clusters in an optimal way is discriminant analysis. **Fisher Linear Discriminant Analysis** [38] [40] seeks a transformation matrix W such that the ratio of the between-class scatter and the within-class scatter is maximized. Let the between-class scatter S_B be defined as follows:

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{2.3}$$

where μ_i is the mean value of class X_i, μ is the mean value of the whole data, c is the number of classes and N_i is the number of samples in class X_i. Let the within-class scatter be:

$$S_W = \sum_{i=1}^{c} \sum_{x_{k,i} \in X_i} (x_{k,i} - \mu_i)(x_{k,i} - \mu_i)^T \tag{2.4}$$

where μ_i is the mean value of class X_i, c is the number of classes and N_i is the number of samples in class X_i. If S_W is not singular, the optimal projection matrix W_{opt} is chosen as the matrix which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples:

$$W_{opt} = argmax_W \frac{|W^T S_B W|}{|W^T S_W W|} = [\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_m}] \tag{2.5}$$

where $\mathbf{w_i}, i = 1 \ldots m$ is the set of S_W-generalized eigenvectors of S_B corresponding to the m largest generalized eigenvalues.

Opposed to PCA behavior, FLD emphasizes the direction in which both classes can be better discriminated. FLD uses more information about the problem as the number of classes and the samples in each of the classes must be known *a priori.* In figure 2.14 the projections on the FLD subspace are well separated.

In real problems, it can occur that it is not possible to find an optimal classifier. A solution is presented by assembling different classifiers.

2.2.4 Adaboost classification process

The Adaboost process is a supervised learning and classification tool, since we know exactly the classes we are seeking. Adaboost is created as a method for combining simple classifiers to obtain a very accurate decision. Roughly, it is an iterative assembling process in which each classifier is devoted to find a good division of the sub-set of points formed by the samples that are more difficult classified up to that point.

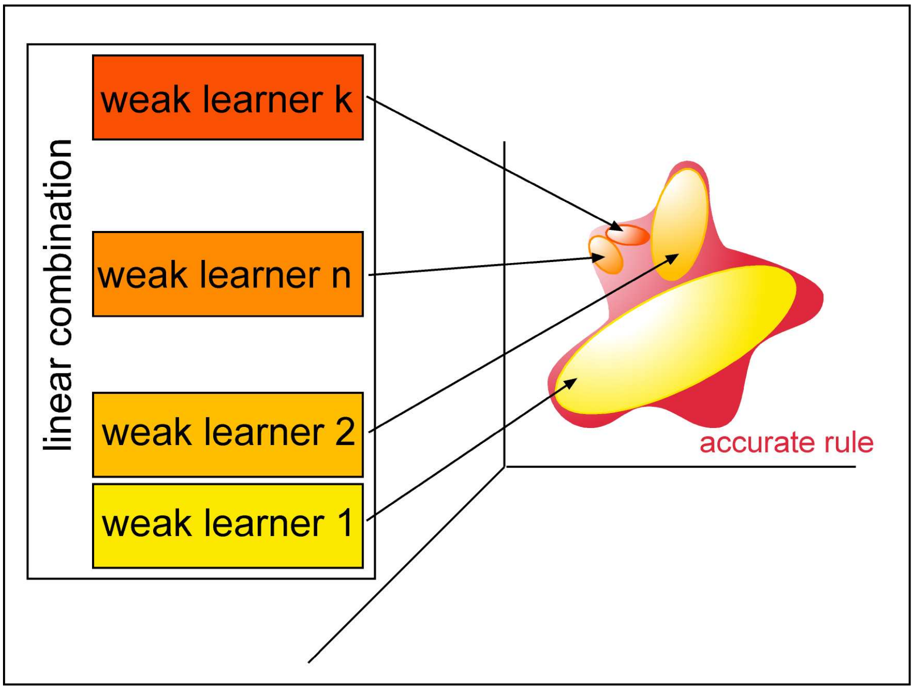


Figure 2.15: Diagram of the AdaBoost rule behavior.

In particular Adaboost is a shortening for *Adaptative Boosting (AdaBoost)*, and is widely recognized as one of the most accurate processes for high accuracy classification.

AdaBoost procedure

Adaptative Boosting (AdaBoost) is an arcing method that allows the designer to keep adding ”weak” classifiers until some desired low training error has been achieved [41] [42] [50]. At each step of the proces, a weight is assigned to each of the feature points. These weights measure how accurate the feature point is being classified at that stage. If it is accurately classified, then its probability of being used in subsequent learners is reduced, or emphasized otherwise. This way, AdaBoost focuses on difficult training points at each stage.

The classification result is a linear combination of the ”weak” classifiers. The weight of each classifier is proportional to the amount of data that classifies in a correct way.

Figure 2.15 shows the evolution of the AdaBoost rule. The first learner tries to deal with the great amount of data of the real rule. However, as it is a weak learner, it probably could not represent the whole rule. Therefore, the AdaBoost process emphasizes the difficult set of data (data that have not been correctly classified) using weights that modify the probability density function of the appearance of each sample data point. Therefore, the next classifier will focus on those samples that are not correctly classified. As the number of classifiers increases, the scope of each new added weak classifier decreases.

The modification of the probability of appearance of each point in the process can be troublesome, since we need to find classifiers that allow weighing the samples points. Another possibility is to resample the data set according to the weights of each feature data. The new set of feature points is used as inputs of the new classifier to be added to the process. Although, this last method is more general it is unadvisable to use it, since after several iterations, the training set can be trimmed to very little data points. Therefore, it hinders the classification process.

As an additional feature, AdaBoost is capable of performing a feature selection process while training. In order to perform both tasks, feature selection and classification process, a weak learning algorithm is designed to select the single features which best separate the different classes. That is, one classifier is trained for each feature, determining the optimal classification function (so that the minimum number of feature points is misclassified). And then, the most accurate classifier-feature pair is stored at that stage of the process. If feature selection is not desired, the weak classifier focuses on all the features at a time.

The general algorithm is described as follows:

- Determine a supervised set of feature points $\{x_i, c_i\}$ where $c_i = \{-1, 1\}$ is the class associated to each of the features classes.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $c_i = \{-1, 1\}$ respectively, where m and l are the number of feature points for each class.
- For $t = 1..T$:
  - Normalize weights

    $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,i}}$$

    so that w_t is a probability distribution.
  - For each feature, j train a classifier, h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i |h_j(x_i) - c_i|$.
  - Choose the classifier, h_t with the lowest error ϵ_t.
  - Update the weights:

    $$w_{t+1,i} = w_{t,i} \beta_t^{e_i}$$

    where $e_i = 1$ for each well-classified feature and $e_i = 0$ otherwise. $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$. Calculate parameter $\alpha_t = -log(\beta_t)$.
- The final "strong" classifier is:

  $$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the strong classifier is the ensemble of a series of simple classifiers ("weak"). Parameter α_t is the weighting factor of each of the classifiers. The loop ends when the classification error of a "weak" classifier is over 0.5, the estimated error for the whole "strong" classifier is lower than a given error rate or if we achieve the desired number of "weaks". The final classification is the result of the weighted classifications of the "weaks". The process is designed so that if $h(x) > 0$, then pixel x belongs to one of the classes.

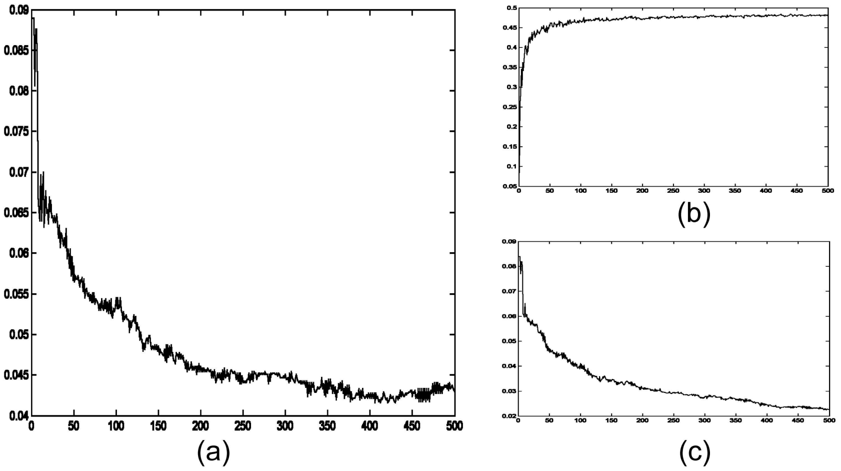


Figure 2.16: Error rates associated to the AdaBoost process. (a) Test error rate. (b) "Weak" single classification error (c) Strong classification error on the training data.

Behavior of the Adaboost procedure

Analyzing the Adaboost process, we can figure out the error rate behavior when adding new "weak" classifiers. As we have described in the former section, the probability of each sample to be used in a "weak" classifier raises if it has been misclassified up to that moment by the "strong" classifier. So if we want to add a classifier $h_{t+1}(x)$, we take the misclassified points according $h(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ and raise its probability in $t+1$. As the set with higher probability is composed by the difficult data points, the "weak" classifier will easily fail in assigning the correct label to each sample. This fact, tells us that the error rate will increase the more classifiers we add. This is true for the transient time. To further understand the behavior of the stationary time, we now describe the behavior of the "strong" classifier error rate.

One of the conditions that stops the process is the fact that the "weak" classifier must perform better than the random guess. That is, we always want the error rate of the "weak" classifier to be under 0.5. If this condition is granted at each step, it means that although the "weak" is focusing on the most difficult data set, it still manages to find a usable solution. This translates in the fact that some misclassified points will now be correctly assigned to the true label. This, of course, is decreasing the error rate of the compound of "weaks". This is true up to the point that if no other stop condition is met, the error rate tends asymptotically to zero.

Resuming the "weak" classification error rate in the stationary stage, it is expected that the classifier will be able to classify correctly at least half plus one of the samples. If this happens, the error will be better than random guess, though tending to 0.5. Otherwise, the "weak" can not be used to train the "strong" classifier and the process will end.

Figure 2.16 shows the evolution of the error rates for the training and the test feature points. Figure 2.16(a) shows the test error rate. One can observe, that the overall error has a decreasing tendency as more "weak" classifiers are added to the process. Figure 2.16(b) shows the error evolution of each of the "weak" classifiers. The figure illustrates how the error increases as more "weak" classifiers are added. Figure 2.16(c) shows the error rate of the system response on the training data. As it is expected, the error rate decreases to very low values. This, however does not ensure a test classification error of such accuracy.

One question arises at this point. What will happen with the test error rate? The answer is not so simple. While we expect the test error rate to decrease in the same way as the training error rate does, one can not guarantee this behavior. However, we realize that if the training set is meaningful, in the sense that it correctly represents the problem, the test error rate should decrease according with the "strong" error rate. But we also must take into account that we have a finite amount of samples, and therefore, the "weak" classifiers could try to distinguish among not representative and conflictive points due to the sampling. This fact can lead to overtraining stages, in which, though the training is correctly classified, as it is not a good representation of the reality, the test samples are misclassified.

The role of the "weak" classifier

The weak classifier has a very important role in the procedure. Different approaches can be used, however it is relatively interesting to center our attention in low time-consuming classifiers.

The first and the most straight forward approach to a "weak" is the perceptron. The perceptron is constituted by a weighed sum of the inputs and an adaptative threshold function. This scheme is easy to embed in the AdaBoost process since it relies on the weights to make the classification.

Another approach to be taken in consideration is to model the feature points as Gaussian distributions. This allows us to define a simple scheme by simply calculating the weighed mean and weighed covariance of the classes at each step of the process:

$$\mu_{i,t}^{j} = \sum w_{i,t} x_i \qquad \Sigma_{i,t}^{j} = \sum w_{i,t}(x_i - \mu_{i,t}^{j})^2$$

for each x_i^j point in class C_j. $W_{i,j}$ are the weights for each data point.

If feature selection is desired, this scheme is highly constrained to the N features of the N-dimensional feature space. If N is not enough large, the procedure could not improve its performance. Therefore we propose another classifier for relatively low dimensional spaces (2 magnitude orders). Because the selection of a single feature for each of the classifiers is quite a hard constraint, we can look for the most significant pair of features which discriminates better the different classes.

For each pair of features of our space one can use linear discriminant analysis to find the transformation which leads to the most discriminant axis. Therefore, the pair of features with the lowest error can be chosen. We can describe this "weak" classifier as follows.

$$h(x) = \begin{cases} 1 & \text{if } p_j W_j^t x < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

where p_j and θ_j are the parity and threshold parameters and W_j is defined as follows:

$$W_j = \Sigma_j^{-1}(\mu_{-1,j} - \mu_{1,j})$$

which is the canonical variate. W_j is the principal axis of the solution of the linear discriminant analysis system which maximizes

$$J(W) = \frac{W^t S_B W}{W^t S_W W}$$

, where S_B is the between-class scatter $S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^t$ and S_W is the within-class scatter$S_W = \sum_{i=1}^{C} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^t$, where μ is the mean value of the whole data, c is the number of classes and N_i is the number of samples in class i.

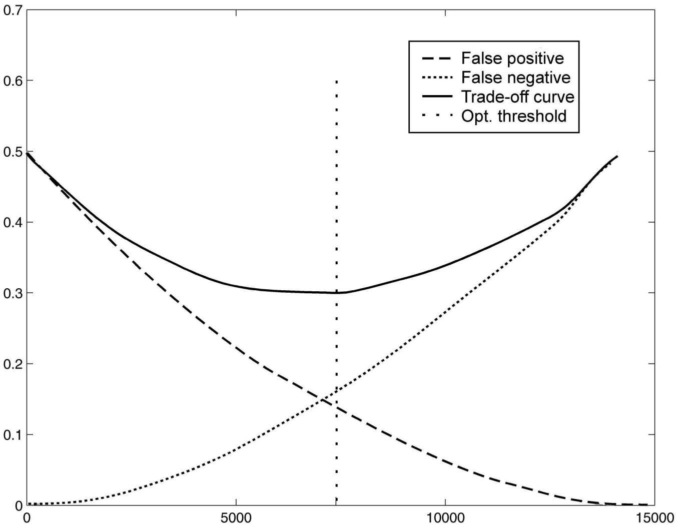


Figure 2.17: Optimal threshold search using ROC curves

Another approach to a ”weak” classifier relies on the use of the ROC curves. The ROC curves show the amount of false positives and false negatives for each possible parameter of the classifier. In particular, if we use a threshold value, it shows the curves for each threshold value. At this point, the optimal threshold value is the minimum of the sum of both curves. This is the optimal trade-off between the misclassification in both classes. This process can be done for each feature using a feature selection plus classification process. This is the approach we have used in this article.

Figure 2.17 shows an example of the optimal threshold search using ROC curves. False positives and false negatives are displayed as two curves. The search for the optimal threshold involves a trade-off between both curves (the continuous line displays the trade-off function). In particular, we look for the minimum amount of false classified data. Therefore, the solution is the minimum of the continuous curve.

In general, the weak classifier information to store at each iteration consists of the features selected f_j, if the feature extraction process is desired, and the parameters of the classifier Θ_j. Those parameters are a threshold θ_j, a parity p_j. Although the threshold separates the two classes it is not enough to identify which class is in either side of the threshold. Therefore, a parameter p_j (parity) is needed to indicate the direction of the inequality sign when classifying:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

Both, the feature extraction and the classification processes, are the central parts of any classification system. We will use the framework explained in the former sections in order to classify exhaustively the plaque in the IVUS image.