THÈSE de DOCTORAT de l'UNIVERSITÉ PARIS VI

Spécialité : Informatique

Équipes d'accueil :
Laboratoires d'Électronique Philips (LEP)
Laboratoire d'Informatique de Paris 6 (LIP6)

présentée par

Olivier GÉRARD

pour obtenir le grade de
Docteur de l'Université PARIS VI

Modélisation de séquences par technique adaptative : prévision de décharges de batteries et extraction de contours dans des images médicales

soutenue le 29 Juin 1999

devant la commission d'examen composée de :

Stéphane Canu	
Isabelle Bloch	Rapporteur
Christian Jutten	Rapporteur
Patrick Gallinari	
Florence d'Alché-Buc	
Jean-Noël Patillon	
Sherif Makram-Ebeid	

Avant-propos

Les travaux de recherche présentés dans ce document ont été effectués au sein des Laboratoires d'Electronique Philips (LEP), en collaboration avec le Laboratoire d'Informatique de l'université Paris 6 (LIP6) et avec le soutien financier de l'ANRT.
Au terme de ce travail, je souhaite exprimer ma reconnaissance envers toutes les personnes qui m'ont aidé tout au long de cette période.

Je souhaite tout d'abord remercier Jean-Bernard Theeten, pour son accueil alors qu'il était responsable de la division *Détection et photonique* et aussi Jean Pergrale, chef de la division *Systèmes d'imagerie professionnelle* pour le soutien et la confiance qu'il me témoigne.
Je tiens à remercier chaleureusement Patrick Gallinari et Florence d'Alché-Buc, du LIP6, pour avoir dirigé mes recherches pendant toute la durée de ce travail de thèse.
Je suis également très reconnaissant envers Jean-Noël Patillon pour avoir encadré une partie de ce travail. Son ouverture d'esprit, sa patience et son insatiable curiosité ont constitué pour moi la meilleure introduction possible au monde de la recherche industrielle.
Sherif Makram-Ebeid, expert de la division *Systèmes d'imagerie professionnelle* est responsable de mon travail en traitement d'images. Ses conseils, ses réflexions et son enthousiasme communicatif ont constitué une aide essentielle pour me diriger dans le vaste monde de l'analyse d'images.
Je remercie vivement Isabelle Bloch, professeur au département Image de l'ENST et Christian Jutten, professeur à l'INPG qui m'ont fait l'honneur de juger ce travail en tant que rapporteurs ; j'espère ne pas avoir abusé de leur gentillesse et de leur disponibilité. Mes remerciements vont également à Stéphane Canu, professeur à l'INSA de Rouen qui a accepté de présider le jury.

Je tiens également à remercier les personnes dont j'ai croisé le chemin au cours de la première partie de mes travaux au LEP : Jean-Pierre Nadal, Lars Hoff et surtout Jean-Philippe Laurent. D'autres personnes ont rendu cette première partie possible et je tiens à les en remercier : les experts Philips des batteries P.H.L. Notten, W. Kuijt et H.J. Bergsveld, ainsi que P. Heydron et S. Swift de Fluke inc. pour avoir fourni la base de données sur les batteries.

Le Dr Jean-Louis Hébert du CHU du Kremlin-Bicêtre, m'a beaucoup appris sur l'aspect clinique de l'angiocardiographie. Je le remercie également de m'avoir permis d'assister à de tels examens et d'avoir influencer certaines parties du manuscrit.

J'adresse de très chaleureux remerciements à tous les membres de la division *Systèmes d'imagerie professionnelle* pour la bonne ambiance générale, seule propice à un travail efficace : Olivier Simon pour sa bonne humeur communicative et ses conseils sur Matlab, Jean-Michel Létang pour sa rigueur intellectuelle, les conseils qu'il m'a prodigués et les nombreuses discussions que nous avons eues, Raoul Florent, Antoine Collet-Billon, Eric Denis, Sylvain Devillers, les collègues thésards Claire Levrier, Marion Benetière et Thomas Deschamps et tous les autres...

Un grand merci à toutes les personnes des services du LEP : secrétariat, bibliothèque, communication, reprographie qui ont beaucoup facilité ma tâche, et, bien entendu, à tous les membres du service informatique (avec une mention spéciale à Marc Germain) sans qui rien n'est possible.

Un clin d'œil à Oliver Vogel et à mes collègues du LIP6 : Hugo Zaragoza, Philippe Leray, Thierry Artières et tous les autres.

Et à Corinne...

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

La détermination de modèles à partir d'observations et l'étude de leurs propriétés constituent les activités essentielles de toute démarche scientifique. Ces modèles, qu'ils soient qualifiés d'hypothèse, de paradigme, de relation statistique ou autres, peuvent être plus ou moins formels, mais ils partagent la caractéristique fondamentale d'essayer de chercher des régularités au sein des observations pour faire émerger des comportements cohérents et prévisibles.
L'intelligence artificielle et le connexionnisme participent de cette modélisation en essayant de la comprendre et de l'automatiser. Leur objectif est l'étude de l'apprentissage, c'est-à-dire de la manière de spécifier un modèle (de déterminer ses paramètres) à partir de données terrain. Le connexionnisme, qui s'appuie sur un formalisme statistique, explore les relations existant entre des données numériques afin de réaliser des tâches complexes comme la régression et la classification. Dans le premier cas, les caractéristiques sous-jacentes extraites du flot de données servent à prévoir l'état du système dans d'autres conditions alors que, dans le second cas, les caractéristiques extraites servent plus à regrouper entre elles les données « semblables » afin de dégager de l'information de plus haut niveau.

Nous nous plaçons dans le cadre de l'étude des systèmes dynamiques, les plus nombreux dans notre environnement direct. La dynamique implique que la valeur de sortie à chaque instant dépende non seulement de l'entrée au même instant mais également de valeurs d'entrées passées. En outre, dans le cas où certaines des entrées ne sont pas directement observables, les sorties sont souvent dénommées séries temporelles ou séquences. Le problème est alors, à nouveau, de constituer un modèle du phénomène temporel sous-jacent. Cette modélisation s'effectue en deux étapes successives. La première consiste à choisir un ensemble de modèles candidats, c'est-à-dire à proposer un type de modélisation. Ce choix est principalement guidé par les connaissances *a priori* sur le problème, mais il reste très délicat car il conditionne tout le bon déroulement de la modélisation. La seconde étape détermine le modèle le plus « adapté » à la tâche à traiter parmi les modèles candidats. C'est une étape d'optimisation des paramètres de modèle et de validation de l'adéquation entre le modèle et les données.

L'objectif de cette thèse est de proposer et de développer une approche générale et robuste pour traiter deux types de problématiques de traitement de séquences :

- la prévision du comportement d'un système dynamique évoluant selon le contexte ;

- la reconnaissance d'une séquence.

Les approches que nous proposons pour résoudre ces deux tâches se fondent toutes les deux sur une modélisation pertinente du phénomène observé et sur l'inférence statistique des paramètres de cette modélisation et de leurs variations. Pour chacune de ces tâches, nous avons développé des sytèmes hiérarchiques et hybrides, qui tirent parti des capacités d'approximateurs universels des réseaux de neurones et qui intégrent de la connaissance *a priori* sur le problème.

La première partie de cette thèse cherche à faire le point sur les méthodes stochastiques pour la modélisation de séquences et sur l'estimation des paramètres de ces modèles. Elle est composée de deux chapitres ; le **chapitre 2** passe en revue les techniques simples d'estimation de densité de probabilité alors que le **chapitre 3** se place dans le cadre plus riche des modèles à états, qu'ils soient issus de l'automatique avec les filtres de Kalman ou de la reconnaissance des formes avec les chaînes de Markov. Ces deux modèles sont mis en regard et il est montré qu'ils ne sont qu'expressions de la même modélisation de séquences temporelles.

La deuxième partie est une application des idées des chapitres précédents et propose une modélisation à états d'un phénomène physique complexe pour résoudre un problème industriel concret présenté dans le **chapitre 4** : la prévision de fin de décharge de batteries rechargeables alimentant un appareil portable. Les batteries rechargeables possèdent des caractéristiques très complexes, difficilement modélisables mathématiquement. Elles sont également très sensibles à l'environnement extérieur et aux conditions d'utilisation. En outre, l'implantation souhaitée sur un appareil portable réduit considérablement le champ des techniques utilisables. Le système proposé s'appuie sur la définition d'un système original, tel que décrit dans le **chapitre 5**. Après une étude des modèles concurrents envisageables, le modèle original proposé [Gérard99c] utilise donc deux réseaux de neurones organisés de façon hiérarchique et capables de s'adapter aux conditions extérieures. Le premier réseau est un simple modèle d'une courbe de décharge alors que le second est responsable de l'adaptation aux données contextuelles et estime les paramètres du premier réseau. Cette estimation est rendue complexe par la forme même d'un espace de poids synaptiques. Afin de simplifier cet espace, un algorithme d'apprentissage original est proposé [Gérard96]. Un système incrémental est en outre proposé ; il permet une adaptation en ligne aux variabilités comportementales des batteries [Gérard97]. Les résultats obtenus sont présentés dans le **chapitre 6** ; ils sont relativement bons, et en tout cas bien meilleurs que les spécifications demandées. L'incrémentalité permet d'obtenir des performances encore meilleures en s'adaptant à la batterie en cours d'utilisation. Ce chapitre contient également une brève description de l'implantation du système.

Dans la troisième partie, un autre problème est traité ; il s'agit de l'extraction d'un contour dans des images médicales et plus précisément, de la détection automatique des contours du ventricule gauche dans des image radiographiques. Ce contour est en effet nécessaire au médecin pour établir son diagnostic de certaines pathologies cardiaques. Le **chapitre 7** décrit l'examen de ventriculographie radiologique ainsi que son intérêt pour l'établissement du diagnostic. On se trouve donc confronté à un problème de détection automatique d'un objet dans des images numériques et le **chapitre 8** présente une étude bibliographique critique des méthodes envisageables en traitement d'image. Cette étude nous a permis d'extraire quelques caractéristiques désirables pour notre système. Le système proposé, objet du **chapitre 9**, utilise le plus possible de l'information *a priori* de haut niveau sur l'objet à détecter pour contraindre la recherche du contour le plus probable [Gérard98]. Cette recherche se base sur un modèle hybride chaîne de Markov - réseau de neurones, c'est-à-dire le deuxième type d'approche pour le traitement de séquences, le contour étant considéré comme l'enchaînement successif de points. Le système proposé comporte les étapes suivantes : classification des points de l'image dans des classes anatomiques ayant une signification pour le médecin, définition d'une première ébauche du contour recherché à l'aide des cartes de probabilités *a posteriori* estimées par le réseau de neurones, et enfin, extraction du contour proprement dite qui est réalisée par programmation dynamique à partir de la définition d'une chaîne de Markov cachée en 2D [Gérard99c]. Les conclusions sur ce travail ainsi que les perspectives d'amélioration et d'extension sont présentées dans le **chapitre 10**.

Une conclusion générale est proposée dans le **chapitre 11**.

Première partie

Modèles stochastiques de séquence et estimation

Chapitre 2

Estimation

Face à un problème réel, la première étape de la compréhension scientifique passe par une modélisation du phénomène observé. La première étape de cette modélisation est souvent d'ordre statistique ce qui nécessite l'estimation de densités de probabilités.

Les différentes méthodes permettant d'estimer les densités de probabilités peuvent être regroupées dans trois grandes classes en fonction de la connaissance possédée *a priori* sur la forme de la densité. La première technique est dite **paramétrique**. Elle fixe des hypothèses fortes sur la forme spécifique de la densité. Les valeurs des paramètres de cette forme sont estimées à partir des observations, le plus souvent en choisissant les valeurs de paramètres les plus vraisemblables. Cette technique très globale est simple mais elle souffre de la rigidité imposée par le choix préalable du type de fonction. La fonction choisie peut, en effet, s'avérer incapable de prendre en compte toute la complexité présente dans les données et donc mal les représenter.
La deuxième classe de méthodes, à l'opposé, ne fonde aucune hypothèse sur la forme de la densité. Elle regroupe des méthodes plus locales, dites **non-paramétriques**. La forme de la densité est entièrement dictée par les données, le plus souvent comme une somme de fonctions centrées sur les données. Cela donne à l'estimation une totale souplesse d'adaptation aux données. Le problème de ces techniques est alors le nombre de paramètres qui est directement proportionnel à la quantité de données.
La troisième classe de méthodes, dites **semi-paramétriques**, tente de prendre le meilleur des deux autres en construisant une forme de fonction très générale dont le nombre de paramètres ajustables varie indépendamment de la taille des données, mais tout en permettant de rendre le modèle plus ou moins flexible.

Ce chapitre présente ces trois classes de méthodes suivi d'une rapide introduction aux méthodes d'apprentissage Bayésien.

2.1 Estimation paramétrique

L'estimation paramétrique se base sur une hypothèse forte sur la forme de la densité de probabilité $p(x|\theta)$, avec x une variable aléatoire de $\mathbb{R}^p$ et θ les paramètres de cette densité. À partir des échantillons indépendants $\mathcal{X} = \{X_1, X_2, ..., X_N\}$ de la variable x, on peut calculer la densité de probabilité jointe de tous ces échantillons par

$$\mathcal{L}(\theta) = p(\mathcal{X}|\theta) = \prod_{i=1}^{N} p(X_i|\theta). \tag{2.1}$$

Cette quantité s'appelle la vraisemblance des paramètres θ par rapport aux données $\mathcal{X}$. Un principe général d'estimation consiste à maximiser cette **vraisemblance** et donc à choisir les valeurs θ qui maximisent $\mathcal{L}(\theta)$. Dans le cas général d'une distribution quelconque, le θ maximum doit être obtenu par des méthodes itératives (1). Cependant, dans le cas spécifique d'une distribution normale des données, c'est-à-dire distribuées selon :

$$p(x) = \frac{1}{\sqrt{(2\pi)^d\ |\Sigma|}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}, \tag{2.2}$$

une solution analytique peut être trouvée (par différenciation de (2.2)) et les paramètres de moyenne μ et de covariance Σ de cette distribution, qui vérifient :

$$\mu = E\left[x\right] = \arg\min_{m} \sum_{i=1}^{N} (X_i - m)^2 \tag{2.3}$$

$$\Sigma = E\left[(x-\mu)(x-\mu)^T\right], \tag{2.4}$$

sont estimés de façon standard par :

$$\tilde{\mu} = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{2.5}$$

$$\tilde{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \left(X_i - \tilde{\mu}\right)\left(X_i - \tilde{\mu}\right)^T . \tag{2.6}$$

La quantité

$$\Delta^2 = (x-\mu)^T\,\Sigma^{-1}(x-\mu) \tag{2.7}$$

qui apparaît dans l'exponentielle de (2.2) est la distance de *Mahalanobis* de x à μ [Duda73].
Cette méthode d'estimation sera utilisée dans le § 9.4.1, p. 141, pour construire un modèle moyen à partir d'un faible nombre de données terrain.

(1). pour des problèmes de calcul numérique.

2.2 Estimation non paramétrique

Les techniques d'estimation non-paramétriques ne présupposent pas de forme *a priori* pour les densités de probabilité génératrices des échantillons. Ces méthodes sont beaucoup plus locales et leur objectif est la constitution d'une loi, directement à partir des échantillons connus (pour un aperçu des techniques d'estimation non paramétrique utilisées en régression, voir l'annexe B de [Goutte97]). Tout comme les techniques des k plus proches voisins, les méthodes d'estimation non-paramétriques les plus répandues sont les méthodes d'**estimation par noyaux**.

En dimension d, l'estimation en un point x de la densité $\tilde{f}(x)$ est obtenue comme la somme d'une suite de noyaux centrés sur les N observations $\{X_1, X_2, \dots, X_N\}$ suivant :

$$\tilde{f}(x) = \frac{1}{N\,h^d} \sum_{i=1}^{N} K\left(\frac{x - X_i}{h}\right), \tag{2.8}$$

les noyaux $K(x)$ vérifiant

$$\int_{\mathbb{R}^d} K(x)\,dx = 1 \quad \text{et} \quad \forall x \in \mathbb{R},\ K(x) \geqslant 0 \tag{2.9}$$

pour garantir que la fonction $\tilde{f}$ soit bien une densité de probabilité.
La constante h représente la largeur du noyau et est également appelée constante de lissage [Saporta90, p. 119]. Si h est faible, $\tilde{f}$ sera très peu régulière, si h est grand $\tilde{f}$ sera très lisse. Le choix de la valeur de h est souvent fait empiriquement, même si l'on sait qu'elle doit être de l'ordre de $N^{-1/(d+4)}$ [Silverman85, p. 86]. Le plus souvent K est une densité symétrique unimodale, comme par exemple la densité normale centrée réduite :

$$K(x) = (2\pi)^{-d/2}\ \exp\left(-\frac{1}{2}\,x^T x\right). \tag{2.10}$$

D'autres choix de noyaux sont possibles, comme le **noyau d'Epanechnikov** qui est défini en dimension d par :

$$K_e(x) = \begin{cases} \frac{d+2}{2\,c_d}\ \left(1 - x^T x\right) & \text{si } x^T x < 1 \\ 0 & \text{sinon} \end{cases} \tag{2.11}$$

où c_d représente le volume de la sphère unité en dimension d ($c_1 = 2$, $c_2 = \pi$, $c_3 = 4\pi/3$, etc.). Ce noyau est beaucoup plus économe en ressource de calcul puisque les contributions des échantillons éloignés tendent rapidement vers 0 et qu'il ne nécessite pas de calcul d'exponentielle. Cette technique sera utilisée pour estimer les densités nécessaires à un calcul d'information mutuelle dans la partie 9.3.3, p. 133.

2.3 Estimation semi-paramétrique

La troisième classe de méthodes d'estimateur, dites **semi-paramétriques**, tente de construire une forme de fonction très générale dont le nombre de paramètres ajustables varie indépendamment de la taille des données. L'intérêt principal de ces méthodes est la grande souplesse du modèle présupposé.

2.3.1 L'algorithme EM *Expectation Maximization* : introduction

L'algorithme EM est une méthode très générale pour approcher une densité de probabilité.

Avant de décrire le cadre complet de l'algorithme EM (p. 17), ce paragraphe présente un exemple d'utilisation, ce qui constitue la meilleure introduction à cet algorithme. Le problème considéré est le suivant : les données sont générées par une densité de probabilité qui est un mélange de k densités normales disctinctes. Le processus de génération de données est le suivant : premièrement, le choix d'une des distributions est réalisé aléatoirement, deuxièmement, un point est généré selon la distribution choisie. Ce processus est itéré pour la constitution d'une base de données. Pour illustrer cette méthode, la cas particulier suivant sera utilisé : $k = 2$, les deux densités ont le même écart-type σ (elles ne diffèrent que par leur moyenne μ_1 et μ_2) et le choix de la distribution suit une loi uniforme. Ce processus de création de la donnée X_i s'écrit donc :

- sélection d'une Gaussienne $j \in \{1, 2\}$ avec $p(j = 1) = p(j = 2) = \frac{1}{2}$ (et donc d'une variable booléenne non observable $Z = (z_{i1}, z_{i2})$ avec $z_{ij} = \delta_{ij}$),
- tirage de $X_i = \mathcal{N}(\mu_j, \sigma)$.

L'objectif de EM est alors de rechercher le maximum de vraisemblance par estimation des variables cachées (z_{ij}) à partir de l'hypothèse courante sur les valeurs des paramètres $(\theta = (\mu_1, \mu_2))$, puis de recalculer les valeurs des paramètres les plus probables. Autrement dit, à partir d'une valeur initiale des paramètres $(\theta^0 = (\mu_1^0, \mu_2^0))$, l'algorithme EM itère les deux étapes suivantes, jusqu'à convergence des valeurs de paramètres :

- calcul de $E[z_{ij}]$, compte tenu de l'hypothèse courante sur les paramètres (c'est-à-dire « choix » d'une gaussienne),
- calcul de nouvelles valeurs de paramètres θ^{k+1} par maximum de vraisemblance, sous l'hypothèse que les variables cachées z_{ij} prennent les valeurs $E[z_{ij}]$ estimées à l'étape précédente (modification des paramètres sous l'hypothèse que la gaussienne qui a généré l'exemple est bien celle qui a été choisie).

Ces deux étapes sont dénommées *Expectation* (calcul d'espérance) et *Maximization* (calcul de maximum), respectivement.

Dans cet exemple, EM peut bien être vu comme une méthode d'estimation paramètrique, puisque l'estimation proprement dite des paramètres s'effectue à partir d'une connaissance très précise de la loi qui a servi à générer les exemples.

Cependant, l'utilisation des variables cachées (ici, Z) permet d'étendre considérablement les capacités de l'estimation paramétrique en rendant le modèle beaucoup plus riche et flexible. EM est donc généralement considéré comme une méthode d'estimation semi-paramétrique. Une présentation générale de l'algorithme EM est donnée dans le § 3.2, p. 17.

2.3.2 Les méthodes de gradient

Les méthodes de gradient sont utilisées dans le cadre supervisé de l'apprentissage pour constituer des systèmes non-linéaires. Les réseaux de neurones formels [LeCun87] sont les méthodes de gradient les plus souples et elles peuvent être efficacement utilisées dans un grand nombre de cas.

Les réseaux de neurones sont souvent considérés comme des estimateurs semi-paramétriques (voir [Bishop95]), dans le sens où le nombre de paramètres ajustables varie de façon presque indépendante (2) de la taille des données tout en permettant de rendre le modèle plus ou moins flexible. Par rapport à EM qui reste très proche de l'estimation paramétrique, les réseaux de neurones sont plutôt de type non-paramétrique dans le sens où ils ne posent pas d'*a priori* fort sur la densité à estimer. Une dénomination plus adéquate est donc de ranger les réseaux de neurones comme estimateur de densité dans la classe des estimateurs **flexibles** [Canu96].

En fait, de nombreux travaux ont même permis de déterminer que les réseaux de neurones sont des approximateurs universels, c'est-à-dire que face à un problème de régression, ils sont capables d'estimer tout type de fonctions continues (voir également § 5.2.3, p. 55).

La deuxième utilisation classique d'un réseau de neurones est la classification. Dans ce cas, il existe deux façons de concevoir l'objectif de l'apprentissage d'un réseau. La première, la plus simple, consiste à utiliser le réseau de neurones pour construire une fonction discrimante non-linéaire de telle sorte que les sorties du réseau déterminent directement la classe d'appartenance de tout vecteur d'observation présenté en entrée. La seconde approche, plus générale et plus puissante utilise le réseau de neurones pour construire un modèle des probabilités *a posteriori* d'appartenance aux différentes classes. Typiquement, on associe une classe à chacune des sorties du réseau dont l'activation va représenter la probabilité *a posteriori* $p(C_k|x)$, où C_k est la $k^{\text{ième}}$ classe et x le vecteur d'entrée. Un apprentissage utilisant sur un critère d'erreur quadratique ou entropique permet d'atteindre cette représentation [Richard91] et l'on dispose alors des propriétés suivantes [Bishop95] :

- la constitution de décisions « optimales » : contrairement au cas de la discrimination, les sorties n'ont pas besoin d'être contraintes à être proches de 0 ou de 1.
- la somme des sorties (des probabilités *a posteriori*) doit valoir 1. Des structures spéciales de réseaux peuvent être utilisées pour s'assurer de satisfaire cette exigence, mais elles ne sont généralement pas nécessaires, une simple « normalisation » étant le plus souvent suffisante.

(2). c'est-à-dire, au moins en relatif, et en tout cas pas de façon linéaire.

– il faut tenir compte de différences dans les probabilités *a priori* entre les données présentes dans la base d'apprentissage et celles de la situation d'utilisation courante : une simple règle de trois, sans modification du réseau, suffit à compenser les différences de situation.

Afin de construire un bon modèle de ces probabilités, il faut bien garder en mémoire que le résultat cherché est un peu différent de la discrimination : ce n'est pas un taux de bonne classification mais plutôt les cœfficients d'un mélange de distributions.

Les capacités en régression seront par la suite utilisées pour approximer des courbes de décharges de batteries reachargeables (deuxième partie) et des approximations de probabilités *a posteriori* obtenues par réseaux de neurones seront intégrées dans un modèle hybride HMM – réseau de neurones pour extraire un contour dans des images médicales (troisième partie).

2.4 L'apprentissage Bayésien

Un modèle Bayésien est donné par la vraisemblance des données $p(\{X\}\,|\,\theta)$, auxquelles on associe une densité de probabilité *a priori* sur les paramètres $p(\theta)$ [Duda73] ; c'est-à-dire que les paramètres θ sont considérés comme des véritables variables aléatoires. Tout l'apprentissage Bayésien repose sur l'obtention de la densité *a posteriori* $p(\theta|\{X\})$ qui se décompose par le théorème de Bayes (3) en :

$$p\left(\theta|\{X\}\right) = \frac{p\left(\{X\}\,|\,\theta\right)\, p(\theta)}{p(\{X\})} \tag{2.12}$$

avec la constante de normalisation $p(\{X\})$ donnée par

$$p(\{X\}) = \int_{\Theta} p(\{X\}|\theta)p(\theta)d\theta. \tag{2.13}$$

La connaissance de cette densité *a posteriori* permet alors le calcul d'estimateurs que l'on peut classer en deux types :

– les estimateurs par intégrales qui possèdent la forme générale :

$$I(f) \triangleq E[f(\theta)|\{X\}] = \int_{\Theta} f(\theta)p(\theta|\{X\})d\theta \tag{2.14}$$

et qui englobent notamment la moyenne *a posteriori* ($f(\theta) = \theta$) et les covariances *a posteriori* ($f(\theta) = \|\theta - E[\theta]\|^2$)

– les estimateurs de type *maximum a posteriori* (MAP) définis par

$$\theta_{\mathrm{MAP}} = \arg\max_{\theta} p(\theta|\{X\}). \tag{2.15}$$

(3). D'après le révérend Thomas Bayes, 1702-1761, qui a décrit un cas particulier de ce théorème dans un article publié deux ans après sa mort. Ce théorème a été redémontré de manière indépendante et dans un cadre plus général par P.-S. Laplace en 1814 (voir [Robert92, p. 16-22]).

2.4.1 Les méthodes de Monte Carlo

Les méthodes de Monte Carlo sont des approximations numériques de l'inférence Bayésienne puisque les différentes intégrales (la constante de normalisation $p(\{X\})$ (2.13), ainsi que les estimateurs) ne sont que très rarement calculables analytiquement. Elles tentent donc d'apporter une solution aux deux problèmes suivants :

- le tirage d'échantillons $\left\{\theta^{(k)}\right\}_1^N$ de la densité $p(\theta|x)$;
- l'estimation des espérances des estimateurs.

L'algorithme EM (voir section 3.2) tente de le faire d'une manière détermistique alors que les méthodes de Monte Carlo sont plutôt d'ordre stochastique.

L'idée générale de ces méthodes est de représenter les densités de probabilités par des ensembles d'individus. Si on a $\left\{\theta^{(k)}\right\}_1^N$ N échantillons distribués selon $p(\theta|x)$, alors une approximation Monte Carlo de cette loi est donnée par

$$\hat{p}_N(\theta|x) = \frac{1}{N}\sum_{k=1}^{N}\delta_{\theta^{(k)}}(\theta) \tag{2.16}$$

où δ est la fonction impulsion de Dirac. L'idée générale est que la « concentration » des échantillons dans une région donnée de Θ sera supposée suffisamment représentative de la probabilité de cette même région. La loi forte des grands nombres permet alors d'approcher $I(f)$ par

$$\int_\Theta f\left(\theta, p(\theta|\{X\})\right) d\theta = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} f\left(\theta^{(k)}\right) \;. \tag{2.17}$$

Le problème est alors celui du tirage efficace d'échantillons i.i.d. selon la densité *a posteriori* $p(\theta|x)$.

L'échantillonneur de Gibbs

L'échantillonneur de Gibbs [(4)] consiste à simuler à tour de rôle les lois conditionnelles sur chacun des paramètres (éléments du vecteur θ). Si l'on définit $\theta = (\theta_1, ..., \theta_q)$, l'itération k de l'algorithme consiste à simuler les paramètres $\theta^{(k)}$ selon les lois conditionnelles de la façon suivante :

$$\begin{array}{lcl} \theta_1^{(k)} & \sim & p\left(\theta_1 \mid \{X\}, \theta_2^{\boldsymbol{k-1}}, ..., \theta_q^{\boldsymbol{k-1}}\right) \\ \theta_2^{(k)} & \sim & p\left(\theta_1 \mid \{X\}, \theta_1^{\boldsymbol{k}}, \theta_3^{\boldsymbol{k-1}}, ..., \theta_q^{\boldsymbol{k-1}}\right) \\ & ... & \\ \theta_q^{(k)} & \sim & p\left(\theta_1 \mid \{X\}, \theta_1^{\boldsymbol{k}}, ..., \theta_{q-1}^{\boldsymbol{k}}\right) \end{array} \tag{2.18}$$

On peut noter que cet algorithme n'est qu'un cas particulier de l'algorithme de Metropolis-Hastings décrit dans le paragraphe suivant.

(4). Ce nom n'est qu'un accident historique et un terme plus exact serait « échantillonneur par substitutions successives » [Andrieu97, p. 16].

La méthode de Metropolis-Hastings

L'algorithme de Metropolis-Hastings permet de simuler des échantillons distribués selon une densité de probabilité cible $\pi(x)$, éventuellement multidimensionnelle et dont on ne peut pas calculer la constante de normalisation. Il se base sur une densité $q(x,.)$ dont on sait tirer des échantillons. Cette densité est qualifiée « d'instrumentale » ou « candidate » [Andrieu97]. Le principe de l'algorithme est de sélectionner parmi les échantillons candidats fournis par $q(x,.)$ ceux qui sont représentatifs de $\pi(x)$. Une itération de cet algorithme consiste donc à simuler un nouveau candidat x' et à ne le garder qu'avec une certaine probabilité acceptation $s(x,x')$. Plus précisément :

- simuler $x' \sim q(x,.)$;
- $s(x,x') = \min\left\{1,\ \frac{\pi(x')}{\pi(x)}\frac{q(x,x')}{q(x',x)}\right\}$;
- accepter x' avec la probabilité $s(x,x')$.

Pour une justification de cet algorithme, se reporter à [Andrieu97]. Cette référence décrit également les choix classiques de $q(x,.)$ (marche aléatoire, Metropolis-Hastings indépendant, Metropolis-Hastings « une variable à la fois »). On peut d'ailleurs noter que l'échantillonneur de Gibbs est un cas particulier de cette dernière version de l'algorithme de Metropolis-Hastings.

Dans le cas qui nous intéresse on ne cherche pas une densité $\pi(x)$ mais plutôt la densité conditionnelle $\pi\left(\theta|\{X\}\right) \propto \pi\left(\{X\}|\theta\right)\pi(\theta)$ (par le théorème de Bayes). Dans ce cas, la probabilité d'acceptation $s(x,x')$ de (4) devient :

$$s(\theta,\theta') = \min\left\{1,\ \frac{\pi(\{X\}|\theta')\pi(\theta')}{\pi(\{X\}|\theta)\pi(\theta)}\frac{q(\theta,\theta')}{q(\theta',\theta)}\right\}. \tag{2.19}$$

Les méthodes de Monte Carlo par chaînes de Markov (MCMC : *Markov Chain Monte Carlo*) ne sont que des cas particuliers (système à états) de ce dernier cas de l'algorithme de Metropolis-Hastings.

On peut également construire des algorithmes mixtes Gibbs - Metropolis-Hastings quand on ne sait pas simuler selon les probabilités conditionnelles du premier (2.18).

Ces techniques peuvent être utilisées pour relaxer les champs de Markov, comme décrit dans en annexe, page 184.

Chapitre 3

Analyse de séquences par modèles à états

On considère maintenant un problème plus général que celui présenté dans le chapitre précédent. Soient X et Y deux processus aléatoires ; le processus X n'est pas observable et le problème général, modélisant de nombreux problèmes réels, est alors celui de l'estimation de la réalisation de X à partir de la réalisation observée de Y ; dans le vocabulaire du traitement du signal, on parle de restauration de X. En d'autres termes, la modélisation utilisée est dite « à état », puisqu'elle repose sur les valeurs de processus X non-observables.

L'autre partie du problème consiste en l'estimation des paramètres du modèle choisi pour représenter le phénomène.

Les problèmes posés et les méthodes utilisées pour modéliser cette situation sont décrits dans le cas le plus général, avant une particularisation à deux modèles à états, l'un à valeur d'état continue, très utilisé dans le domaine de l'automatique : le filtre de Kalman, et l'autre surtout employé en reconnaissance de formes : les chaînes de Markov cachées. Une discussion montre ensuite que ces deux types de modèles ne sont que deux expressions de la même modélisation et qu'ils sont tous deux très utiles pour l'analyse de séquences temporelles.

3.1 Le cadre de l'estimation

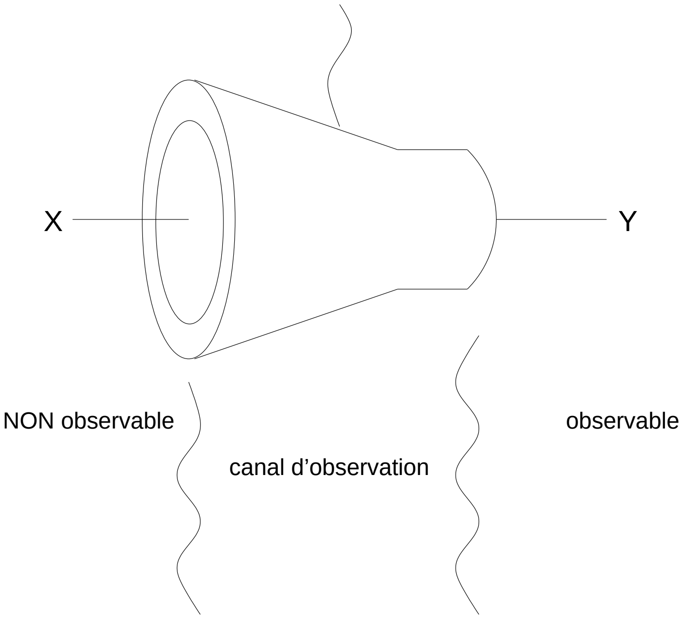

FIG. 3.1: *Schéma de l'observation indirecte d'un phénomène physique.*

Pour beaucoup de problèmes, on n'a souvent accès qu'à une partie des variables. X constitue les variables cachées et on suppose que l'on peut facilement mesurer des observations Y_k d'un espace $\mathcal{Y} \subset \mathbb{R}^p$, pour $k = 1, ..., T$. Dans le vocabulaire du traitement du signal, le processus Y est considéré comme une version « bruitée » du processus X. La notion de « bruit » est cependant à considérer dans un sens très général qui peut englober aussi bien des caractéristiques du canal d'observation que de véritables bruits (acquisition, transmission, ...), (voir la figure 3.1).

$f(X, Y|\theta)$ est la famille de fonctions au sein de laquelle on cherche la génératrice de l'ensemble des exemples complets. De la même façon, $g(Y|\theta)$ est la famille de fonctions parmi laquelle on recherche la génératrice de notre échantillon. L'estimation de paramètres consiste alors à estimer des paramètres θ à valeurs dans un ensemble $\Theta \subset \mathbb{R}^q$.

Il existe de nombreuses méthodes d'estimation qui s'attaquent à ce problème, selon les hypothèses faites sur la loi de (X, Y) et le type d'estimation utilisée (déterministe, stochastique,...). La première famille de méthodes regroupe des techniques qui font jouer le même rôle aux variables observées et non-observées, que l'on qualifie alors de variables manquantes. Parmi ces méthodes, l'estimation déterministe effectuée par l'algorithme EM sera décrite dans le paragraphe suivant. Il existe également des méthodes mixtes déterministes-stochastiques (ICE [Pieczynski94]) et enfin totalement stochastiques (SEM [Celeux95] et les méthodes de Monte Carlo), mais elles sortent du cadre de ce document. La seconde famille est de type « hiérarchique » dans le sens où elle ne fait pas du tout jouer le même rôle à X et à Y. Y est plutôt considéré comme une version « bruitée » mais observable de X. Le paragraphe 3.3 introduit cette famille dont deux méthodes sont décrites et comparées : le filtre de Kalman (qui suppose une dépendance linéaire entre X et Y) et les modèles de Markov (avec une forme particulière de la loi $p(Y|X)$).

3.2 L'algorithme EM *Expectation Maximization*

L'algorithme EM est une méthode très générale pour approcher une densité de probabilité. Il est généralement utilisé pour calculer l'estimation la plus probable à partir de données incomplètes, dans le cadre de modèles paramétriques. En se plaçant dans le cadre général défini précédemment, on définit X comme étant les variables cachées et Y les variables que l'on peut mesurer. Un échantillon complet est formé de $Z = (X, Y)$ où seul Y est observable. C'est-à-dire que EM accorde le même rôle fonctionnel aux deux variables.

À partir de $f(X, Y|\theta)$, la famille de génératrices des exemples complets et de $g(Y|\theta)$, la famille de génératrices d'observations, l'objectif est alors d'estimer θ par $\hat{\theta} = \arg\max_\theta L(\theta)$ avec $L(\theta) = \log g(Y|\theta)$.

Au lieu de chercher à maximiser la fonction inconnue $f(X, Y|\theta)$, l'algorithme EM va chercher à maximiser, de façon itérative, l'espérance conditionnelle de $\log f(X, Y|\theta')$ connaissant Y pour la valeur courante de θ [Dempster77]. Plus formellement [Celeux95], on définit

$$k(X|Y,\theta) = \frac{f(X,Y|\theta)}{g(Y|\theta)} \tag{3.1}$$

et

$$g(Y \mid \theta) = \int_{X\in\mathcal{X}} f(X, Y \mid \theta)\, dX. \tag{3.2}$$

On définit $\mathcal{Q}(\theta'|\theta)$ comme

$$\mathcal{Q}(\theta'|\theta) = E\left[\log P(X, Y \mid \theta') \mid \theta, Y\right] \tag{3.3}$$

$$\mathcal{Q}(\theta'|\theta) = \int_{\mathcal{X}} \log f(X, Y|\theta')\, k(X|Y,\theta)\, dX. \tag{3.4}$$

L'algorithme EM consiste donc à

- initialiser θ, *i.e.* choisir $\theta^{(0)}$, puis à itérer les deux étapes suivantes ;
- calculer $\mathcal{Q}(\theta|\theta^{(i)})$ à $\theta^{(i)}$ donné ;
- choisir $\theta^{(i+1)} = \arg\max_\theta \mathcal{Q}\left(\theta|\theta^{(i)}\right)$ par calcul de $\frac{\partial \mathcal{Q}\left(\theta|\theta^{(i)}\right)}{\partial\theta} = 0$.

Les deux dernières étapes sont souvent dénommées E (*expectation*) et M (*maximization*), respectivement et elles sont réitérées jusqu'à convergence apparente.

La propriété principale de EM est d'augmenter à chaque itération la vraisemblance de l'estimée *i.e.* $\mathcal{L}\left(\theta^{(i+1)}\right) \geqslant \mathcal{L}\left(\theta^{(i)}\right)$ avec $\mathcal{L}\left(\theta^{(i+1)}\right) = \mathcal{L}\left(\theta^{(i)}\right) \Leftrightarrow \mathcal{Q}\left(\theta^{(i+1)}|\theta^{(i)}\right) = \mathcal{Q}\left(\theta^{(i)}|\theta^{(i)}\right)$.

Sous certaines conditions de régularité, la séquence $\{\theta^{(i)}\}$ converge vers un point stationnaire de $L(\theta)$. Ce point peut cependant être un minimum local.

EM peut être vu comme une version déterministique de l'échantilloneur de Gibbs. Ils sont tous deux utilisés pour approcher dans le cas de données manquantes. EM travaille sur les moyennes de variables inconnues en utilisant des statistiques que l'on espère suffisantes alors que l'échantilloneur de Gibbs échantillonne les variables inconnues.

Parmi les nombreuses applications de l'algorithme EM à l'estimation de densités de probabilité, un cas particulier mérite d'être relevé : l'estimation de mélange de gaussiennes. EM permet en effet l'estimation des coefficients de mélange en même temps que les paramètres de chacune des gaussiennes.

3.3 Introduction aux méthodes « hiérarchiques »

On se place dans le cas où les variables manquantes (non-observées) sont les états sous-jacents d'un système. C'est-à-dire que les variables aléatoires sont des processus aléatoires et le processus X_k n'est pas observable. Cette optique est différente de celle utilisée par EM et ici X et Y ne jouent pas du tout le même rôle dans le modèle. Plus précisément, on suppose la dépendance des variables observées Y_k à l'instant k, par rapport aux variables cachées X_k :

$$Y_k = f_b(X_k, Y_1, ..., Y_{k-1}, Y_{k+1}, ..., Y_T) = f_b(X_k). \tag{3.5}$$

On suppose également que X est un processus de Markov du premier ordre :

$$X_{k+1} = f_a(X_1, X_2, ..., X_k) = f_a(X_k). \tag{3.6}$$

Ces deux événements sont généralement connus sous les noms d'**émission** et de **transition**, selon la terminologie des chaînes de Markov cachées ; les symboles s'y référant seront B et A, respectivement.

On a donc le système général

$$\left\{ \begin{array}{rcl} X_{k+1} & = & f_a(X_k) \\ Y_k & = & f_b(X_k) \end{array} \right. \tag{3.7}$$

Ce système peut être discret, c.-à-d. avec un nombre d'états fini et discret ou continu avec $X \in \mathbb{R}^n$. Dans le premier cas, ce système décrit une **chaîne de Markov** et les fonctions f_a et f_b sont probabilistes alors que dans le second, ce sont des multiplications matricielles ($f_a(X_k) = A_k X_k$) et le système est un **système dynamique linéaire**.

La section 3.7 décrit le cas du système discret alors que le cas continu est décrit dans 3.6. Pour ces deux systèmes, on est confronté à deux problèmes :

- la reconnaissance(5), qui cherche à estimer les états sous-jacents X les plus probables compte tenu des observations Y ;

(5). étape aussi dénommée **restauration** de X, terme provenant du vocabulaire du traitement du signal pour lequel Y est considéré comme une version bruitée de X, bien qu'ici le mot bruit doive être considéré dans un sens très général (voir également la modélisation en champs de Markov cachés, p. 184).

– l'estimation des paramètres (les fonctions f_a, f_b et les bruits éventuels).

Ces deux problèmes sont traités pour chacun des deux systèmes.

Il existe de très nombreuses analogies entre ces deux systèmes et elles sont résumées dans la section 3.8. Avant même de commencer à différencier les deux cas, l'algorithme général *forward-backward* est présenté dans la section suivante, suivi par une présentation de deux algorithmes de reconnaissance.

3.4 L'algorithme *Forward-Backward*

L'algorithme *forward-backward* estime de façon récursive l'état courant X_k par maximisation de sa probabilité conditionnelle aux observations. Ces probabilités sont bien évidemment utilisées pour la phase de reconnaissance, mais elles sont également très utiles pour la phase d'estimation des paramètres. Le calcul des probabilités forward et backward est présenté ici dans sa version discrète mais son utilisation dans le cas continu est une simple extension.
On pose (6)

$$\mathcal{A}_k(u,x) = P(X_k = x \,|\, X_{k-1} = u) \tag{3.8}$$

et

$$\mathcal{B}_k(x,\, y) = P(Y_k = y \,|\, X_k = x). \tag{3.9}$$

3.4.1 Les probabilités Forward

Le passage en-avant consiste à estimer $P\left(X_k \,|\, \{Y\}_1^k\right)$. Il est cependant plus simple de calculer la probabilité jointe et on définit

$$\alpha_k(x) = P\left(\, X_k = x,\, \{Y\}_1^k \,\right). \tag{3.10}$$

En utilisant la propriété markovienne du processus et les indépendances relatives entre les processus qui affirment que

$$P(Y_k = y \,|\, X_1, \dots, X_T, Y_1, ..., Y_{k-1}, Y_{k+1}, ..., Y_T) = P(Y_k = y \,|\, X_k = x), \tag{3.11}$$

on obtient

$$\alpha_k(x) = \mathcal{B}_k(x,y) \sum_{u \in \mathcal{X}} \alpha_{k-1}(u)\, \mathcal{A}_{k-1}(u,x), \tag{3.12}$$

les probabilités de transition des variables cachées avec $\mathcal{X}$ l'espace des valeurs du processus X. L'état filtré estimé (en ne tenant compte que des observations passées) est l'état qui maximise cette quantité, c.-à-d. :

$$\hat{X}_k = \arg\max_x \, \alpha_k(x). \tag{3.13}$$

(6). de manière plus rigoureuse, il faudrait définir $\mathcal{B}_k(x,\, y)$ comme la densité de probabilité $p(y \,|\, X_k = x)$.

3.4.2 Les probabilités Backward

Après le temps T, on peut estimer les probabilités conditionnelles en remontant le temps et on définit :

$$\beta_k(x) = P\left(\{Y\}_{k+1}^T \mid X_k = x\right) \tag{3.14}$$

De la même façon que pour $\alpha_k(x)$, $\beta_k(x)$ se calcule récursivement par

$$\beta_k(x) = \sum_{u \in \mathcal{X}} \mathcal{A}_k(x,u)\, \mathcal{B}_{k+1}(u,y)\, \beta_{k+1}(u). \tag{3.15}$$

Le calcul des probabilités forward et backward se heurte à des difficultés d'ordre numérique et dans certains cas, il est préférable de remplacer les probabilités conjointes par les probabilités *a posteriori*, et $\alpha_k(x)$ et $\beta_k(x)$ peuvent être reformulées comme suit (voir [Benmiloud95])

$$\begin{aligned} \alpha_k(x) &= P\left(X_k = x \,|\, \{Y\}_1^k\right) \\ \beta_k(x) &= \frac{P\left(\{Y\}_{k+1}^T \,|\, X_k = x\right)}{P\left(\{Y\}_{k+1}^T \,|\, \{Y\}_1^k\right)} \end{aligned} \tag{3.16}$$

3.5 Deux algorithmes de reconnaissance

Comme précisé plus haut, le problème de la reconnaissance (ou restauration) de la séquence $\{X\}$ consiste à estimer la séquence d'états $\{X_k\}_1^T$ la plus probable, au sens de certains critères, en fonction des observations $\{Y_k\}_1^T$. Les sections suivantes présentent deux méthodes bayésiennes, correspondant à deux fonctions de perte différentes, de la restauration lissée ($E[X_k|\{Y\}]$, en tenant compte également de toute la séquence des observations) de $\{X\}$.

3.5.1 Mode de la marginale *a posteriori*

Cette méthode est également connue sous le nom : maximum de vraisemblance *a posteriori* local. Le principe de la méthode **MPM** est la maximisation de la probabilité de chacun des états *a posteriori* à chaque instant, et cela, indépendamment de toute information sur la structure globale de la séquence $\{X\}$. Comme $P\left(X_k = x|\{Y\}\right) = \alpha_k(x)\beta_k(x)$, l'estimé lissé de l'état au temps k est donné par

$$\tilde{X}_k = \arg\max_x \alpha_k(x)\beta_k(x)\ . \tag{3.17}$$

Cette technique peut également être utilisée dans le contexte des champs de Markov cachés. Elle est alors approchée par des méthodes itératives introduites dans [Marroquin87], voir en annexe, p. 184.

3.5.2 Maximum *a posteriori*

La méthode du Maximum *a posteriori* maximise la probabilité *a posteriori* de tout le chemin parcouru à travers les états. Cette probabilité est définie par

$$\phi_k(x) = P\left(\mathcal{C}(x,k), \{Y\}_1^k\right) \tag{3.18}$$

où $\mathcal{C}(x,k) = (x_1(x), ..., x_{k-1}(x), x)$ est le chemin le plus probable depuis le départ et finissant sur l'état x au temps k. $\phi_k(x)$ s'obtient de façon récursive

$$\phi_k(x) = \max_{u \in \mathcal{X}} P\left(Y_k = y, X_k = x \mid X_{k-1} = u\right) \; \phi_{k-1}(u) \tag{3.19}$$

avec

$$P\left(Y_k = y, X_k = x \mid X_{k-1} = u\right) = \mathcal{A}_k(u,x)\, \mathcal{B}_k(x,y). \tag{3.20}$$

L'état filtré au temps k connaissant $\{Y\}_1^k$ est alors estimé par l'algorithme de Viterbi selon

$$\check{X}_k = \arg \max_{x \in \mathcal{X}} \phi_k(x). \tag{3.21}$$

Il est également possible d'obtenir une estimation lissée en effectuant un passage en reculant de la forme

$$\check{X}_T = \check{X}_T \tag{3.22}$$

$$\check{X}_{k-1} = \xi_{k-1}(\check{X}_k) \tag{3.23}$$

avec

$$\xi_k(x) = \arg\sup_u \; \mathcal{A}_k(u,x)\, \phi_k(x). \tag{3.24}$$

La solution analytique de la recherche de ce Maximum *A Posteriori* (ou **MAP**) est donnée par l'algorithme de **Viterbi** (7) (voir [Forney73] pour une présentation générale). Cette technique peut également être utilisée dans le contexte des champs de Markov cachés. Elle est alors approchée par recuit simulé, comme introduit dans [Geman84].

3.5.3 Résumé des méthodes de reconnaissance

Les méthodes classiques de reconnaissance peuvent être classées en fonction des observations utilisées (filtrage ou lissage) et en fonction de l'information utilisée sur les états (indépendants ou en séquence). Ces méthodes peuvent donc être regroupées au sein du tableau suivant (voir également [Delyon93]) :

	Filtrée	Lissée
Etat indépendant	$\hat{X}_k = \arg\max_x \; \alpha_k(x)$ (Forward)	$\tilde{X}_k = \arg\max_x \; \alpha_k(x)\beta_k(x)$ (Forward-Backward, MPM)
Séquence	$\check{X}_k = \arg\max_x \; \phi_k(x)$ (Viterbi filtré)	$\check{X}_T = \check{X}_T$, $\check{X}_{k-1} = \xi_{k-1}(\check{X}_k)$ (Viterbi)

TAB. 3.1: *Résumé des méthodes de restauration bayésienne.*

(7). Cet algorithme a été originalement proposé par A. Viterbi dans [Viterbi67].

3.6 Les systèmes dynamiques linéaires

Ces systèmes forment une grand part des systèmes étudiés dans les domaines du contrôle et de l'identification, c'est-à-dire de l'automatique. L'hypothèse faite ici sur la loi de (X, Y) est très forte puisqu'elle impose une dépendance linéaire entre Y et X.

3.6.1 Le modèle

Les systèmes dynamiques linéaires du premier ordre se présentent sous la forme générale d'un système de deux équations appelées transition (équation d'état) et observation respectivement :

$$\begin{cases} X_{k+1} &= A_k X_k + CU_k + w_k \\ Y_k &= B_k X_k + v_k \end{cases} \tag{3.25}$$

où $X_k \in \mathcal{X}$ (et $\mathcal{X} \subset \mathbb{R}^n$) est l'état courant du système, $U_k(1 \times m)$ est la commande du système, c'est-à-dire son véritable vecteur d'entrée (de dimension m), via la matrice de commande $C(n \times m)$. La sortie mesurable du système à l'instant k est Y_k qui est un vecteur de $\mathbb{R}^p$. $A_k(n \times n)$ relit l'état à l'instant k à celui à l'instant $k+1$. La matrice d'observation $B_k(p \times n)$ permet de relier l'état X_k à la sortie Y_k dans le cadre de l'équation d'observation. w_k et v_k sont

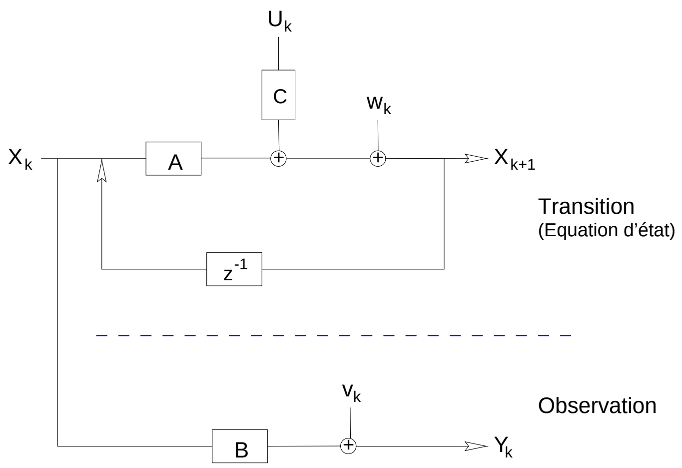

FIG. 3.2: *Le système dynamique linéaire du premier ordre : schéma synoptique.*

des bruits représentant respectivement l'imprécision sur l'état et sur la mesure de la sortie. Ce sont tous les deux des bruits blancs gaussiens avec des matrices de covariances respectives $Q(n \times n)$ et $R(p \times p)$, c.-à-d. :

$$\begin{array}{rclrcl} E[w_k] &=& 0, & E[w_k w_l^T] &=& Q\delta_{kl} \\ E[v_k] &=& 0, & E[v_k v_l^T] &=& R\delta_{kl} \\ E[w_k v_l] &=& 0, & \forall k, l &\in& \{1, ..., T\}^2 \end{array} \tag{3.26}$$

Ce système est représenté dans la figure 3.2.
On peut déjà faire les remarques suivantes :

- la présentation faite ici est du domaine temporel discret mais son extension au temps continu est directe [Auger98] ;

- le système et les équations qui en découlent sont décrits dans le cas général de matrices A et B dépendantes du temps (A_k, B_k) bien que, le plus souvent, elles soient considérées comme constantes ;

- le filtre de Kalman fonctionne également avec des bruits quelconques vérifiant (3.26) [Radix70, p. 102].

Les probabilités de transition $\mathcal{A}(X_{k-1}, X_k) = P(X_k \mid X_{k-1})$ et d'émission $\mathcal{B}(X_k, Y_k) = P(Y_k \mid X_k)$ sont alors définies (par abus de langage sur les réalisations/variables) par :

$$\mathcal{A}(X_{k-1}, X_k) = \frac{1}{\sqrt{(2\pi)^n |Q|}} \exp\left\{-\frac{1}{2}\left(X_k - A_k X_{k-1}\right)^T Q^{-1}\left(X_k - A_k X_{k-1}\right)\right\} \tag{3.27}$$

et

$$\mathcal{B}(X_k, Y_k) = \frac{1}{\sqrt{(2\pi)^p |R|}} \exp\left\{-\frac{1}{2}\left(Y_k - C_k X_k\right)^T R^{-1}\left(Y_k - C_k X_k\right)\right\}. \tag{3.28}$$

En utilisant la propriété Markovienne de ce système, la probabilité jointe des séquences d'états et d'observations est donnée par

$$P(\{X\}, \{Y\}) = P(X_1) \prod_{k=2}^{T} \mathcal{A}(X_{k-1}, X_k) \prod_{k=2}^{T} \mathcal{B}(X_k, Y_k). \tag{3.29}$$

Le problème général posé à partir de l'équation (3.25) est l'estimation de certains paramètres θ au temps k (θ_k), en tenant compte de la séquence d'observation $\{Y\} = Y_1, ..., Y_T$. Deux sous-problèmes sont généralement différenciés [Rauch65], selon les valeurs relatives de k et T :

- le **lissage** qui consiste à fonder l'estimation de θ_k sur la totalité des observations $Y_1, Y_2 ...., Y_T$, c.-à-d. $E\left[\theta_k \mid \{Y\}\right]$. Le lissage est non-causal dans le sens où il utilise des observations futures pour estimer l'état courant. Ici, on a $k \leqslant T$.

- le **filtrage** qui consiste à estimer θ_k récursivement à partir de l'observation Y_k et du passé, c.-à-d. $E\left[\theta_k \mid \{Y\}_1^k\right]$. Cela revient au lissage mais pour $k = T$.

On peut également chercher à prédire θ_k pour $k \geqslant T$.

3.6.2 Estimation des états

Le problème ici est d'estimer les états X_k. Le filtre de Kalman original apporte une réponse filtrée (comme son nom l'indique) où X_k est estimé connaissant la séquence des observations passées, $\{Y\}_1^k$. Cette méthode classique est décrite dans la section suivante. C'est exactement l'analogue du passage forward de l'algorithme forward-backward. Plus précisément, dans un espace d'états continu, (3.12) devient

$$\alpha_k(x) = \mathcal{B}(x,y) \int_{u\in\mathcal{X}} \alpha_{k-1}(u)\mathcal{A}(u,x)\,du, \tag{3.30}$$

où $\alpha_k(x)$ suit une loi normale. De la même façon, les sommes deviennent des intégrales sur l'espace d'états continu pour les autres équations.

Il existe également une version lissée de cette estimation (introduite dans [Rauch65], voir également [Digalakis93]) qui tient compte de la totalité de la séquence $\{Y\}$ pour estimer X_k. Cette dernière version est en fait une incarnation de l'algorithme forward-backward et est décrite à la page 26. Le filtrage de Kalman est apparu dans le début des années 60 [Kalman60] et a eu un retentissement très important [Sorenson85].

Le filtrage de Kalman classique

Le filtrage de Kalman [Kalman60, Radix70, Bertein98] a pour objectif général l'estimation de l'état $X_k \in \mathbb{R}^n$ d'un système discret du premier ordre en fonction des observations passées $\{Y\}_1^k$. L'objectif est donc l'estimation de $E\left[X_k|\{Y\}_1^k\right]$.

On définit $\hat{X}_{k|k-1}$ comme l'état estimé *a priori*, c.-à-d. connaissant tout le passé du processus jusqu'à l'instant $k-1$ compris.

$$\hat{X}_{k|k-1} = P(X_k \mid Y_1, ..., Y_{k-1}) \tag{3.31}$$

$\hat{X}_{k|k}$ est l'état estimé *a posteriori*, c.-à-d. connaissant également la sortie courante Y_k et

$$\hat{X}_{k|k} = P(X_k \mid Y_1, ..., Y_k). \tag{3.32}$$

On peut alors estimer les matrices d'erreur de covariance *a priori* $P_{k|k-1}$ et *a posteriori* $P_{k|k}$ par :

$$P_{k|k-1} = \mathrm{E}\left[\left(X_k - \hat{X}_{k|k-1}\right)\left(X_k - \hat{X}_{k|k-1}\right)^T\right] \tag{3.33}$$

$$\text{et } P_{k|k} = \mathrm{E}\left[\left(X_k - \hat{X}_{k|k}\right)\left(X_k - \hat{X}_{k|k}\right)^T\right]. \tag{3.34}$$

Le filtrage de Kalman se compose de deux étapes exécutées itérativement. La première étape est la propagation de la prédiction sur les états. Celle-ci s'effectue selon les équations suivantes (pour les moments d'ordre 1 et 2) :

$$\begin{cases} \hat{X}_{k+1|k} &= A_k \hat{X}_{k+1|k} + CU_k \\ P_{k+1|k} &= A_k P_{k|k} A_k^T + Q_k \end{cases} \tag{3.35}$$

Le passage de l'estimation *a priori* ($\hat{X}_{k|k-1}$) vers celle *a posteriori* ($\hat{X}_{k|k}$) s'effectue en tenant compte de la nouvelle mesure de la sortie Y_k et plus précisément de la différence entre Y_k et la sortie estimée par $B_k \hat{X}_{k|k-1}$. Cette différence

$$\nu_k = Y_k - B_k \hat{X}_{k|k-1} \tag{3.36}$$

est souvent dénommée « résidu » ou « innovation » apportée par la mesure et la seconde étape du filtrage de Kalman comporte alors une correction qui peut s'écrire :

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} \nu_{k+1}. \tag{3.37}$$

La matrice $K_k (n \times m)$ est le gain de Kalman et elle minimise la matrice de covariance de l'erreur *a posteriori* (3.34). Cette minimisation s'effectue par substitution de $\hat{X}_{k|k}$ tel que donné par (3.37) dans (3.34), calcul des espérances et de leurs dérivées par K_k et résolution à dérivée nulle.

On trouve alors un K_k qui minimise (3.34) et qui peut se mettre sous la forme :

$$K_k = \frac{P_{k|k-1} B_k^T}{B_k P_{k|k-1} B_k^T + R_k}. \tag{3.38}$$

Si la covariance de l'erreur de mesure R_k s'approche de zéro, le gain K_k fait peser le résidu plus fort et

$$\lim_{R_k \to 0} K_k = B_k^{-1}, \tag{3.39}$$

ce qui donne bien $\lim_{R_k \to 0} \hat{X}_{k+1|k+1} = B_k^{-1} Y_k$. C'est-à-dire que plus l'erreur de mesure diminue, plus le filtre fait confiance à la véritable mesure Y_k au détriment de $B_k \hat{X}_{k|k-1}$. Inversement, quand la covariance de l'erreur *a priori* tend vers zéro, le gain K_k diminue également et

$$\lim_{P_{k|k-1} \to 0} K_k = 0 \tag{3.40}$$

c.-à-d. que plus la covariance *a priori* diminue, plus le filtre accorde de crédit à la prédiction de la sortie par $B_k \hat{X}_{k|k-1}$.

On remarque que le filtre de Kalman effectue bien ce que l'on attend de lui, c.-à-d. qu'il maintient les deux premiers moments de la distribution sur les états, et que la variance de cette distribution est bien $P_{k|k}$

$$\begin{aligned} \mathrm{E}\left[X_k\right] &= \hat{X}_{k|k} \\ \mathrm{E}\left[\left(X_k - \hat{X}_{k|k}\right)\left(X_k - \hat{X}_{k|k}\right)^T\right] &= P_{k|k} \\ P(X_k \mid Y_k) &\sim \mathcal{N}(\hat{X}_{k|k}, P_{k|k}). \end{aligned} \tag{3.41}$$

En résumé, le filtrage de Kalman suppose une connaissance préalable des matrices A_k, B_k, C_k, des matrices de covariance des bruits Q et R ainsi que de l'état à l'origine X_1.

L'estimation des états X_k au temps k compte tenu des observations passées $\{Y\}_1^k$ comporte deux étapes successives, la prédiction *a priori* puis la correction en fonction de la mesure Y_k. Celles-ci sont effectuées par les équations suivantes [Létang93].

Pour la prédiction :

$$\left\{ \begin{array}{lcl} \hat{X}_{k+1|k} & = & A_k \, \hat{X}_{k+1|k} + CU_k \\ P_{k+1|k} & = & A_k \, P_{k|k} \, A_k^T + Q_k \end{array} \right. \tag{3.42}$$

Pour la correction :

$$\left\{ \begin{array}{lcl} \hat{X}_{k+1|k+1} & = & \hat{X}_{k+1|k} + K_{k+1} \, \nu_{k+1} \\ P_{k+1|k+1} & = & P_{k+1|k} - K_{k+1} \, B_{k+1} \, P_{k+1|k} \end{array} \right. \tag{3.43}$$

avec

$$\left\{ \begin{array}{lcl} K_{k+1} & = & P_{k+1|k} \, B_{k+1}^T \, S_{k+1}^{-1} \\ S_{k+1} & = & B_{k+1} \, P_{k+1|k} \, B_{k+1}^T + R_{k+1} \\ \nu_{k+1} & = & Y_{k+1} - B_{k+1} \, \hat{X}_{k+1|k} \end{array} \right. \tag{3.44}$$

L'algorithme de Kalman lissé

Cette estimation des états X_k dépend de la totalité de la séquence $\{Y\}$. La méthode est en fait de type forward-backward [Rauch65, Digalakis93].

Le passage forward se compose du filtre de Kalman classique, équations (3.42), (3.43) et (3.44).

Le passage backward est réalisé par (voir [Shumway82]) :

$$\hat{X}_{k|T} = \hat{X}_{k|k} + D_{k+1} \left(\hat{X}_{k+1|T} - \hat{X}_{k+1|k} \right) \tag{3.45}$$

$$P_{k|T} = P_{k|k} + D_{k+1} \left(P_{k+1|T} - P_{k+1|k} \right) D_{k+1}^T \tag{3.46}$$

$$D_{k+1} = P_{k|k} \, A_k^T \, P_{k+1|k}^{-1}. \tag{3.47}$$

Cette estimation lissée correspond exactement à l'estimation MPM. Cependant, dans le cas d'un système linéaire gaussien, [Delyon93] a montré que les algorithmes MPM et MAP donnent exactement les mêmes estimations, tout comme les versions seulement filtrées (forward et Viterbi filtrée).

Le filtrage de Kalman étendu

Dans la présentation du filtre de Kalman classique, le problème général est d'estimer l'état $\mathcal{X} \subset \mathbb{R}^n$ d'un processus discret du premier ordre qui est décrit par une équation *linéaire*. Pour le cas non linéaire, un filtre de Kalman étendu a été développé. Son principe est très simple, uniquement basé sur une étape de linéarisation autour des états estimés. Le système d'équations (3.25) décrivant le système prend la forme suivante :

$$\begin{cases} X_{k+1} &= f(X_k,\, U_k,\, w_k) \\ Y_k &= h(X_k,\, v_k) \end{cases} \tag{3.48}$$

avec les fonctions f et h décrivant les relations non linéaires de transition et d'observation, respectivement. L'estimation des états repose alors sur l'approximation au premier ordre du développement limité de ces équations. Ceci permet d'aboutir à la définition d'un nouveau système qui est, lui, linéaire et pour lequel un filtre de Kalman classique peut être utilisé (voir par exemple [Welch95]).

3.6.3 Estimation des paramètres

L'algorithme classique de Kalman ne fait qu'estimer de façon itérative l'état caché d'un système dynamique linéaire et ceci en supposant connus tous les paramètres de ce système (les matrices A_k, B_k, C_k, Q et R et l'état à l'origine X_1).

Un niveau de modélisation supérieur consiste à estimer ces paramètres.

Estimation de Q et R

Le premier travail visant à l'estimation de paramètres pour le filtre de Kalman s'est attaché à l'estimation des matrices de bruit Q et R en supposant tous les autres paramètres connus [Mehra70]. Cette méthode se fonde sur l'observation suivante : si le filtre est optimal, la séquence d'innovation ν_k (3.36) est un bruit gaussien(8) blanc (et réciproquement). À chaque fois que le filtre ne satisfait plus cette hypothèse, les estimations de Q et R sont modifiées.

Estimation par l'algorithme EM

La première utilisation de l'algorithme EM pour l'estimation des paramètres d'un système dynamique linéaire a été présentée dans [Shumway82]. La partie E (=*expectation*) de la méthode se base sur les probabilités *a posteriori* estimées par le filtre de Kalman lissé présenté plus haut. Cependant, la technique proposée suppose que la matrice d'observation B_k soit connue. Ce travail a ensuite été étendu au cas général par [Ghahramani96a]. Dans ce dernier travail, l'étape E est toujours composée par le filtre de Kalman lissé alors que la phase M permet l'estimation des matrices B, A, R et Q (supposées constantes pour toute la séquence d'observation).

(8). ici, l'hypothèse Gaussienne des bruits v et w est nécessaire, voir p. 22.

3.7 Les chaînes de Markov cachées

Le modèle présenté dans 3.1 est exactement celui traité par les modèles de Markov cachés, avec la principale différence que les variables X_k sont à valeur dans l'ensemble fini et discret des états (ou classes) $\Omega = \{\omega_1, \ldots, \omega_K\}$.
Une autre différence importante réside dans l'absence d'entrée (la commande U_k de (3.25)[(9)]. Enfin, l'hypothèse faite sur la loi de (X, Y) est relativement souple puisque ce modèle autorise toutes les lois, à partir du moment où il est possible de définir la loi $P(Y|X)$. À ce titre, les modèles de Markov cachés entrent dans le cadre des approches paramétriques de la reconnaissance des formes. Nous présenterons uniquement les principes de base des HMM et nous recommanderons au lecteur intéressé [Rabiner86].

3.7.1 Le modèle

Un modèle de Markov caché (HMM, *Hidden Markov Model*) est un automate à états finis stochastique ; en fait, un double processus stochastique avec un processus stochastique sous-jacent qui n'est pas observable (caché) et qui n'est visible qu'à travers un autre processus stochastique.

Autrement dit, une chaîne de Markov cachée est un processus à temps discret doublement stochastique, ou encore composé de deux processus $X = (X_k)_{k\in\mathbb{N}^*}$ et $Y = (Y_k)_{k\in\mathbb{N}^*}$. Le processus X est une chaîne de Markov et le processus Y est réel (chaque Y_k prend ses valeurs dans $\mathbb{R}$). L'appellation « cachée » signifie que les réalisations de X sont inobservables.

Le processus caché $X = (X_k)_{k\in\mathbb{N}^*}$ est représenté sous la forme d'états (chaque X_k prend ses valeurs dans $\Omega = \{\omega_1, \ldots, \omega_K\}$) alors que l'autre processus $Y = (Y_k)_{k\in\mathbb{N}^*}$ est caractérisé par des probabilités d'émission de symboles.

Une chaîne de Markov cachée peut donc se représenter sous la forme présentée dans la figure 3.3 (à comparer avec la figure 3.2, p. 22).

La loi du processus X est alors déterminée par la loi π de X_1, dite loi initiale, et la suite de matrices de transition $A = (a_{ij})$. Un HMM est donc caractérisé par trois ensembles :

- π distribution initiale des états ;
- A matrice de transition entre états ;
- B les probabilités d'émission de chaque symbole dans chacun des états.

Dans le cas des HMM, les probabilités de transition entre états sont données par une matrice et les densités de probabilité des observations sont conditionnées par les états, et les équations (3.8) et (3.9) peuvent être réecrites sous la forme simplifiée :

$$\mathcal{A}_k(i,j) = P(X_k = \omega_j \mid X_{k-1} = \omega_i) = a_{ij}^k \tag{3.49}$$

$$\text{et } \mathcal{B}_k(j,\, y) = P(Y_k \mid X_k = \omega_j) = b_j^k(y). \tag{3.50}$$

(9). On peu cependant citer des travaux visant à introduire des *Input Ouput HMM* [Bengio94].

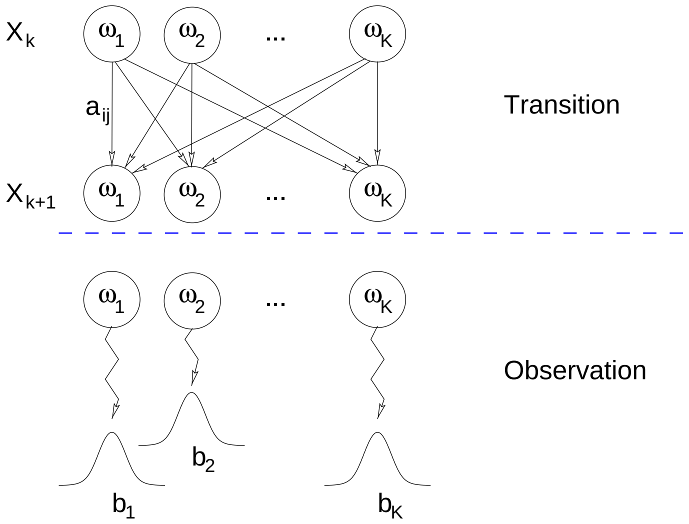

FIG. 3.3: *Une chaîne de Markov cachée : schéma synoptique.*

3.7.2 Estimation des états

La restauration des états X (le décodage, la reconnaissance) s'effectue exactement comme décrit précédemment pour les système à états continus avec, de nouveau, les notations simplifiées pour les probabilités forward α (3.10) : $\alpha_k(i) = P(X_k = \omega_i, \{Y\}_1^k)$ et backward β (3.14) : $\beta_k(i) = P(\{Y\}_{k+1}^T | X_k = \omega_i)$. Plus précisément, l'équation (3.12) devient

$$\alpha_k(j) = b_j^k(y) \sum_{i=1}^{K} \alpha_{k-1}(i)\, a_{ij}^k, \tag{3.51}$$

et (3.15) s'explicite par

$$\beta_k(i) = \sum_{j=1}^{K} a_{ij}^{k+1}\, b_j^{k+1}(y)\, \beta_{k+1}(j). \tag{3.52}$$

L'algorithme le plus utilisé dans ce cadre est l'algorithme de Viterbi, qui, comme présenté dans le § 3.5.2, p. 20 permet de déterminer la séquence d'états la plus probable (maximum *a posteriori*) correspondant à une séquence d'observations donnée[(10)]. C'est d'ailleurs cette méthode que nous allons utiliser pour l'extraction d'un contour dans des images médicales, dans le § 9.4.3, p. 150.

3.7.3 Estimation des paramètres

La phase d'estimation des paramètres, ou d'apprentissage, vise à construire un HMM et, en particulier, à trouver des valeurs aux paramètres du modèle

(10). il existe également des méthodes pour déterminer les N meilleures séquences, les méthodes *N-best*, voir [Stévenin96, chap. 10] choisi. Ces paramètres sont les probabilités de transition (codées dans la matrice A) et les paramètres internes aux distributions d'observation choisies. Pour des raisons d'efficacité, ces densités sont généralement considérées comme gaussiennes et les paramètres sont alors de type (moyenne, variance) (μ_i, σ_i^2). Cette phase d'apprentissage s'effectue de façon itérative à partir de la définition initiale des matrices A et B et π. Une méthode EM ([Baum70, Dempster77]) est généralement utilisée pour optimiser les paramètres. Comme décrit précédemment, cet algorithme se compose de deux phases. La première phase cherche à estimer les probabilités conditionnelles $P(X_k|\{Y_k\})$ et $P(X_k, X_{k-1}|\{Y_k\})$ (ce qui est fait par l'algorithme *forward-backward*), ces probabilités sont utilisées dans la seconde phase pour mettre à jour les paramètres. On doit également fournir des valeurs initiales pour les paramètres. Il n'y a pas de solution optimale pour ce problème, mais [Rabiner86] a montré que le choix initial pour les observations B est très inportant.

Il est également possible d'utiliser d'autres méthodes d'estimation dans le cadre de variables « manquantes », comme les méthodes SEM [Celeux95] et par ICE [Pieczynski94]. On a alors le tableau suivant [Benmiloud95] :

Paramètres à estimer	EM	ICE	SEM
Transition : a_{ij}	Déterministe		Stochastique
Émission : (μ_i, σ_i^2)	Déterministe	Stochastique	

TAB. 3.2: *Les différents types d'apprentissage des paramètres des* HMM.

Extension des HMMs

Une extension des HMMs est présentée dans [Ghahramani96b]. Il s'agit des *Factorials* HMM qui présentent la caractéristique de posséder plusieurs processus cachés (voir figure 3.4 et une autre représentation dans la figure 3.6). Cette méthode est un modèle naturel pour des systèmes dans lesquels l'état inobservable est une configuration d'un ensemble de systèmes dynamiques indépendants. En outre, un *Factorial* HMM est capable de représenter un très grand nombre d'états de façon bien plus efficace qu'un simple HMM pour lequel il serait nécessaire de constituer les états par produit cartésien de tous les états de tous les systèmes dynamiques sous-jacents. [Jordan99] cite l'exemple suivant : à partir de 5 systèmes dynamiques possédant chacun 10 états, la taille de l'espace d'état est de 100.000. Mais les transitions de probabilités sont représentées avec 500 paramètres pour un *Factorial* HMM contre 10^{10} éléments pour la matrice de transition d'un simple HMM.

Les séquences observables par ceux-ci sont générées à partir d'une distribution normale dont la moyenne est la somme pondérée des états des différents processus cachés.

Une architecture assez semblable est présentée dans [Fraser93]. Il s'agit de HMM mixtes pour lesquels les probabilités d'émission ne dépendent pas uniquement de l'état courant mais également des symboles émis dans le passé. Un filtre linéaire autorégressif est associé à chaque état.

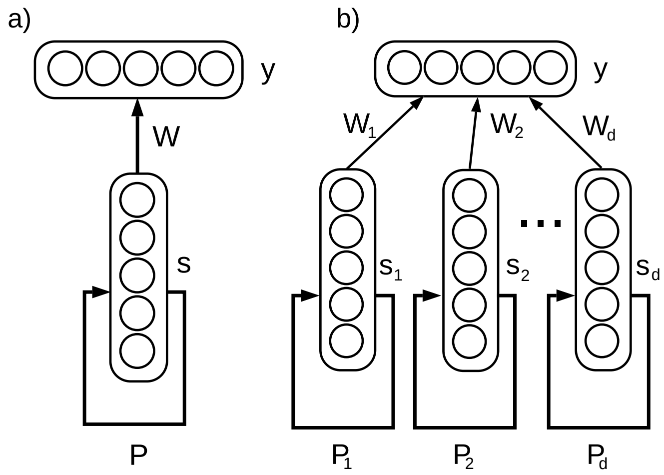

FIG. 3.4: *Comparaison entre : a)* HMM. *b) Factorial* HMM

Ces techniques sont dans leurs formes assez proches des modèles compétitifs décrits dans la section 6.4.

3.8 Analogies entre les systèmes continus (Kalman) et discrets (HMM)

Nous avons donc montré que les systèmes à états continus et discrets ne sont que deux manifestations de la même modélisation et les analogies relevées sont reportées dans le tableau 3.3 de la page suivante.

La représentation sous forme de modèles graphiques des HMMs et des systèmes dynamiques linéaires est exactement identique (voir figure 3.5). Les modèles graphiques sont une forme d'analyse stochastique des processus aléatoires qui permet la représentation simple des relations de causalité entre les variables [Jordan99] et qui autorise le calcul d'inférence. Ainsi les flèches de la figure 3.5 doivent être lue comme des relations de causalité entre la destination et la source.

Selon le même principe, le modèle *Factorial* HMM décrit dans le paragraphe précédent peut se réprésenter sous la forme présentée dans la figure 3.6 (ici avec trois processus cachés).

Chaînes de Markov cachées	**Systèmes dynamiques linéaires**
État	
discret $X_k \in \Omega = \{\omega_1, \dots, \omega_K\}$	continu $X_k \in \mathbb{R}^n$
Observations	
probabilités d'émission $b_j^k(y)$	équation d'observation $Y_k = B_k X_k + v_k$
Reconnaissance des états X_k	
forward	Kalman
Viterbi filtré	Kalman
Marginale *a posteriori* (MPM)	Kalman lissé
Maximum a posteriori (MAP, Viterbi)	Kalman lissé
Estimation des paramètres θ (EM)	
EM : Baum-Welch [Baum70, Benmiloud95] ICE, SEM [Benmiloud95]	[Delyon93, Ghahramani96a]

TAB. 3.3: *Analogies entre les systèmes à états discrets et continus.*

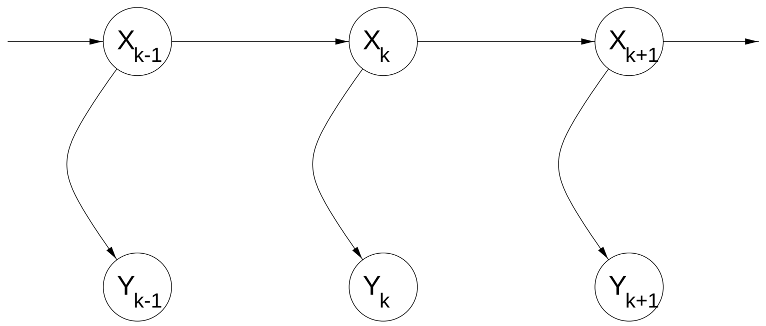

FIG. 3.5: *Les systèmes hiérarchiques représentés comme des modèles graphiques.*

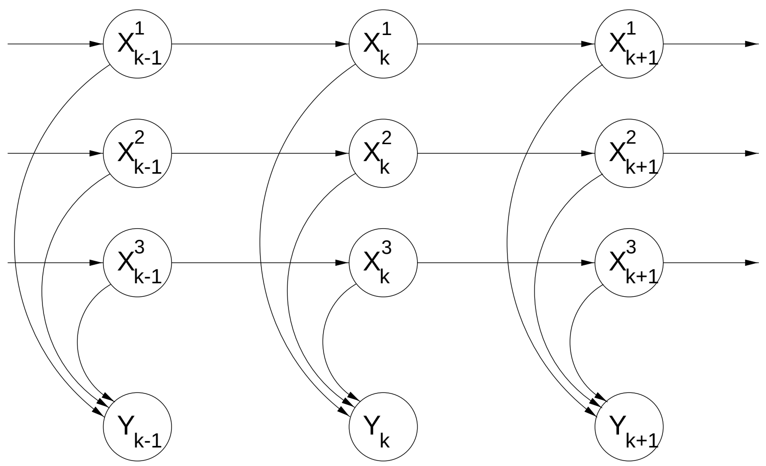

FIG. 3.6: *Les modèles Factorial* HMM *représentés comme des modèles graphiques.*

Deuxième partie

Prédiction par modélisation neuronale : application aux batteries rechargeables

Chapitre 4

Introduction

Les deux chapitres précédents ont présenté des méthodes très générales d'estimation statistique de paramètres de modèles. Ce que nous allons essayer de réaliser dans ce chapitre est l'application des idées des chapitres précédents en proposant une modélisation à états d'un phénomène physique complexe pour résoudre un problème industriel concret. Cette modélisation est fondée sur deux réseaux de neurones organisés de façon hiérarchique.

Le problème industriel, consiste en la gestion intelligente de batteries rechargeables, et plus particulièrement de celles insérées dans des appareils portables, marché en pleine expansion. Ces batteries possèdent des caractéristiques très complexes, difficilement modélisables mathématiquement. Elles sont également très sensibles à l'environnement extérieur et aux conditions d'utilisation. En outre, l'implantation souhaitée sur un appareil portable réduit considérablement le champ des techniques utilisables. Le système proposé s'appuie sur la définition d'un système hiérarchique proche du domaine de l'automatique pour permettre une prévision fiable du temps de décharge restant.

4.1 Le problème

La croissance du nombre d'appareils portables est très importante ces dernières années, tant en volume qu'en nouveaux produits, ainsi qu'en atteste le formidable succès des téléphones mobiles. Mais on peut également parler des téléphones sans fils, des rasoirs électriques, brosses à dents, séche-cheveux, des baladeurs, des ordinateurs, jeux électroniques, des téléviseurs, des petits appareils de mesures (thermomètres, multimètres), petits appareils domestiques, outillages électriques ainsi que de nombreux autres produits actuellement à l'étude. Outre ces petits appareils, un autre marché à très fort potentiel est en train d'émerger : celui des voitures électriques. Ce marché global d'appareils énergiquement autonomes est devenu très concurrentiel et de très importantes ressources sont consacrées à la recherche et au développement de nouvelles fonctionnalités. Cependant, tous ces appareils ont en commun leur dépendance en une source d'énergie efficace pour l'utilisateur, surtout que la batterie peut représenter plus de la moitié du poids, du volume et du coût de la plupart des appareils portables. Cette efficacité se décline sur de nombreux aspects tels que : capacité maximale, capacité par unité de poids, vitesse de charge, taux de décharge en inactivité (auto-décharge), nombre de cycles de charge-décharge, résistance aux conditions extérieures (particulièrement la température), coût, possibilité de recyclage, ... De très nombreuses équipes de recherche en électrochimie tentent de développer des batteries « parfaites », mais il semble que ce soit l'usage envisagé de la batterie qui dicte le type de batterie à utiliser.

En effet, outre les besoins spécifiques des différents appareils portables, de nombreux paramètres très importants au bon fonctionnement des batteries sont directement sous la responsabilité de l'utilisateur final, comme les conditions d'utilisation et surtout la gestion de l'utilisation dans le temps (durée et fréquence de l'utilisation et des phases de charge, décharge, repos).

On s'intéresse au problème de prévision suivant : fournir à l'utilisateur d'un appareil portable une prévision fiable du temps d'utilisation restant, c.-à-d. du temps qu'il reste avant que la capacité interne de la batterie ne soit trop faible pour pouvoir alimenter le système.

Cette prévision est bien évidemment utile pour l'utilisateur mais elle l'est également pour la batterie. En effet, les systèmes de prévision actuels ont une marge d'erreur importante et on impose à la prévision d'être forcément pessimiste afin de ne pas avoir de coupure inopinée. Les systèmes actuels activent l'alarme dès que l'utilisation dépasse 70% de la capacité nominale. On a alors un système qui sous-utilise sa batterie, ce qui augmente le nombre de cycles et donc le vieillissement de la batterie. La plupart des systèmes actuels sont basés sur un coulombmètre qui mesure la charge entrante et sortante de la batterie sous la forme $Q = I * t$. Cependant, le courant I est très difficile à mesurer avec précision, d'où les mauvais résultats de ces méthodes. Beaucoup de sociétés travaillent actuellement sur ce problème mais, même si leurs « résultats » sont toujours proclamés haut et fort : 2% d'erreur dans des conditions « normales » (sic) d'utilisation, leurs techniques ne sont jamais décrites. [Gérard97] contient un aperçu des nombreuses affirmations trouvées dans la presse.

4.2 Les difficultés

4.2.1 Les batteries

Les batteries rechargeables sont des éléments électro-chimiques chargés de transformer de l'énergie électrique en énergie chimique et inversement (voir [Linden95] et [Notten84], par exemple, pour de plus amples descriptions). Même si leur principe est relativement simple, elles présentent des caractéristiques complexes très difficilement modélisables. Une rapide présentation des réactions électro-chimiques dans les batteries Ni-Cd est donnée dans l'annexe A.

La figure 4.1 présente quelques courbes de décharge de ces batteries, ainsi que le temps à prédire pour chacune d'entre elles, c.-à-d. le temps correspondant à la limite inférieure de la tension nécessaire pour alimenter l'appareil portable. On peut déjà faire des remarques très importantes à propos de ces courbes de décharge :

- elles ont toutes la même forme générale (décroissance rapide suivie d'un plateau et enfin chute brutale de la tension) ;
- les temps à prévoir sont facilement identifiables sur chacune des courbes ;
- ce temps est très variable (du simple au double).

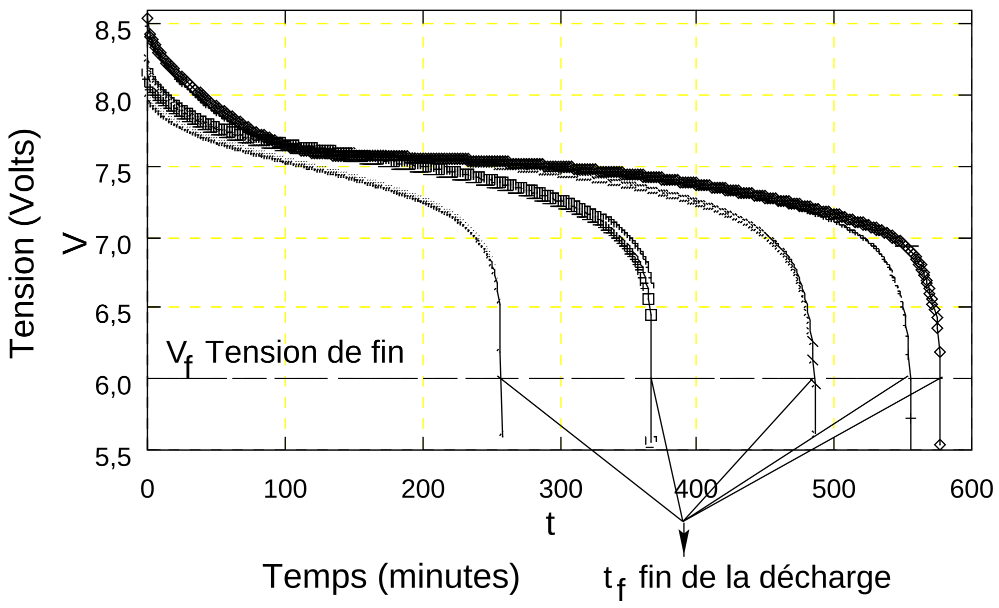

FIG. 4.1: *Quelques courbes de décharge à température constante.*

Les batteries présentent un autre phénomène très difficilement prévisible : le vieillissement. Plus une batterie a fait de cycles (charge puis décharge), moins ses décharges durent. Cet effet est cependant très non-linéaire, comme cela est montré dans la figure 4.2. Globalement, on peut dire que le comportement des

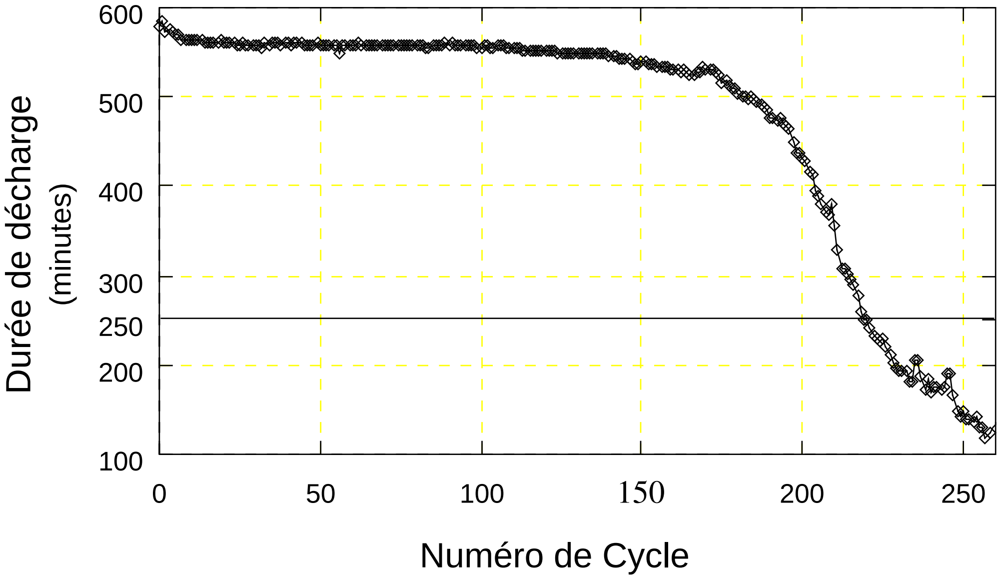

FIG. 4.2: *Le vieillissement des batteries.*

batteries dépend étroitement de leur passé. Elles sont ainsi sensibles à l'utilisation effective. Ces paramètres « contextuels » sont très importants et ils doivent être intégrés dans notre modélisation sous la forme de paramètres de second niveau. Une étude statistique ainsi que des discussions avec les spécialistes Philips des batteries nous ont conduit à sélectionner les paramètres « contextuels » suivants :

- V_0, la valeur de la tension à la mise en fonction de l'appareil ;
- ΔV_0, la dérivée de la tension à la mise en fonction de l'appareil ;
- S, l'état de charge de la batterie, estimé comme la différence entre un ratio de charge et un ratio de décharge, ceux-ci étant calculés comme la somme des rapports des temps mesurés en charge (resp. décharge) sur les temps maximaux nominaux respectifs ;
- t_R, le dernier temps de repos, c.-à-d. le temps passé entre la fin de la dernière charge et le début de la décharge courante ;
- T_R, la température extérieure moyenne pendant ce temps de repos t_R.

Ces variables sont regroupées dans $\varphi = (V_0, \Delta V_0, S, t_R, T_R)$ et semblent contenir assez d'information sur l'état de la batterie pour pouvoir en déduire la forme de la prochaine décharge (voir également [Patillon96c]).

Afin d'avoir une bonne modélisation, il faut tenir compte (être capable de s'adapter) de ces conditions extérieures, mais ces cellules électro-chimiques montrent un autre phénomène très important : elles présentent d'importantes différences individuelles. Ainsi, deux batteries issues du même lot de fabrication peuvent avoir des temps de décharge qui diffèrent de ± 15 %. En outre, plus les batteries vieillissent, plus ces différences peuvent s'accroître et après le même temps de fonctionnement, certaines batteries pourtant fabriquées de façon identique peuvent durer deux fois plus longtemps que d'autres. Il faut donc tenir compte de cette diversité individuelle pour avoir une prévision fiable.

Outre toutes les caractéristiques présentées jusqu'alors, certaines batteries rechargeables (et particulièrement les Ni-Cd, Nickel-Cadmium) présentent un effet mémoire qui se caractérise par un important mimétisme de la décharge courante vis-à-vis des précédentes et qui peut avoir pour conséquence une décharge très raccourcie (malgré de bonnes conditions de charge). Ce phénomène est cependant assez mal expliqué et ne sera pas pris en compte explicitement dans notre modélisation.

4.2.2 Le cas du multimètre portable

Dans notre application particulière, il s'agit d'un multimètre portable haut de gamme de la société américaine FLUKE illustré dans la figure 4.3 (dimensions réelles : 24,5×13,5×5,5 cm). Outre les paramètres listés dans la section précédente, les batteries sont également très sensibles à un autre paramètre qui peut normalement varier durant une décharge : le courant. Cependant, dans le cas du multimètre, les décharges s'effectuent à puissance constante et le paramètre de courant n'est donc pas libre.

Le multimètre portable utilisé est alimenté par un pack de six batteries Ni-Cd. C'est ce type de batterie qui domine actuellement le marché bien que de nouvelles batteries à composition chimique différente commencent à s'imposer (Ni-MH, Nickel-Métal Hydrure et Li-Ion, Lithium-Ion). Dans la suite, nous ne parlerons que de « batteries » bien que le fait de travailler avec des packs rende la situation bien plus complexe, compte tenu des nombreuses interactions internes aux packs.

Le fait de travailler avec un appareil portable introduit de très nombreuses limitations sur le type de méthodes utilisables pour réaliser l'objectif recherché : l'affichage de l'information sur le temps d'utilisation restant. Ces limitations sont principalement :

- le temps de calcul disponible qui doit être réduit le plus possible afin de fournir l'information à l'utilisateur dans un délai très bref,
- la puissance de calcul du microprocesseur embarqué (un MOTOROLA 68306) relativement modeste,
- le surcoût en termes de calcul créé par la prévision du temps restant doit se concentrer pendant les périodes « creuses » afin de ne pas gêner l'utilisation courante de l'appareil,
- la mémoire doit être réduite (surtout celle de type volatile et réinscriptible),
- les capteurs de données ne doivent pas être nombreux ni extrêmement précis afin de ne pas entraîner de surcoûts trop importants,

– les conditions générales d'utilisation d'un appareil portable qui nécessitent l'emploi de méthodes robustes aux situations inattendues et qui doivent toujours rendre possible la « désactivation » de toute nouvelle fonctionnalité adaptative, pour prévenir tout cas de dérive temporelle.

FIG. 4.3: *Le multimètre portable* FLUKE *867.*

4.3 Formalisation du problème

Il n'existe pas – à notre connaissance – de modèle physique permettant une description complète et précise du comportement des batteries. La prévision doit donc se fonder sur des outils statistiques capables d'apprentissage à partir d'exemples.

La fonctionnalité recherchée s'apparente donc à un problème de prévision mais avec des contraintes très lourdes sur le type de méthode utilisable. En outre, cette prévision doit être adaptative, à la fois aux conditions d'utilisation et aux comportements individuels, c.-à-d. à des quantités qui n'évoluent pas sur la même échelle de temps. Nous proposons de faire un rapide tour d'horizon des très nombreuses méthodes envisageables issues de différentes communautés et de proposer une formalisation de ce problème.

Il existe une très abondante littérature issue des domaines de l'économétrie, du traitement du signal et de la statistique qui vise à traiter des problèmes de prévision par étude des **séries temporelles**. Le principe général est d'essayer de prévoir la valeur suivante de la série en fonction des valeurs passées et d'un certain modèle de la courbe. Parmi les très nombreuses techniques, on s'intéressera plus particulièrement aux méthodes neuronales. Ce domaine et les nombreuses architectures utilisées sont très largement présentés dans [Canu95] ainsi que les résultats d'une compétition organisée par le *Sante Fe Institute* [Weigend93]. Cette modélisation est cependant difficilement utilisable pour l'application visée pour deux raisons principales. La première est d'ordre technique, puisque la prévision à un pas de temps proposée par ces méthodes imposerait de coûteux calculs durant la décharge, quand les capacités du portable seraient bien plus utilement dédiées à répondre aux besoins de l'utilisateur. La seconde raison réside dans le manque de « visibilité » et de « clarté » de telles méthodes. En effet ce type de méthode ne permet généralement pas l'introduction explicite de connaissances de haut niveau sur le modèle sous-jacent au phénomène, ne différencie pas plusieurs rythmes de variation des variables et surtout n'introduit aucune garantie contre le risque de dérive temporelle.

Nous proposons plutôt de modéliser la totalité de la courbe de décharge par une fonction paramétrée. Une nécessité de robustesse et le besoin d'un véritable modèle générique ont conduit au choix d'une approche fondée sur la modélisation des courbes de décharge sous la forme :

$$t = f_\lambda(V, T) \tag{4.1}$$

où T est la température (qui peut varier au cours d'une décharge) et f est chargée d'approximer le temps t qu'il reste avant d'atteindre des valeurs V et T particulières. $\lambda \in \Omega$ représente les paramètres influençant directement la forme de la courbe, c.-à-d. le comportement général des batteries. Ω représente l'espace de ces paramètres avec en général : $\Omega \subset \mathbb{R}^r$. La valeur de ces paramètres doit être adaptée vis-à-vis des variables contextuelles φ, telles que définies dans le paragraphe précédent. Cela permet de constuire un modèle moyen de comportement, mais il est aussi important de modifier ces valeurs en fonction des comportements individuels. Cette adaptation doit être effectuée en ligne.

Cette approximation de la fonction « inverse » de celle présentée dans la figure 4.1 permet de déterminer directement le temps restant t_f en plaçant V_f dans (4.1), en supposant T constant jusqu'à la fin de la décharge. Ce modèle mathématique de courbe possède également l'avantage de pouvoir être facilement remplacé par un modèle électro-chimique explicite dès que celui-ci sera disponible [11]. On se place donc dans le cadre de l'**identification** et plus précisément de l'identification adaptative directe, tel que présenté dans la figure 4.4 (voir également [Stearns88]).

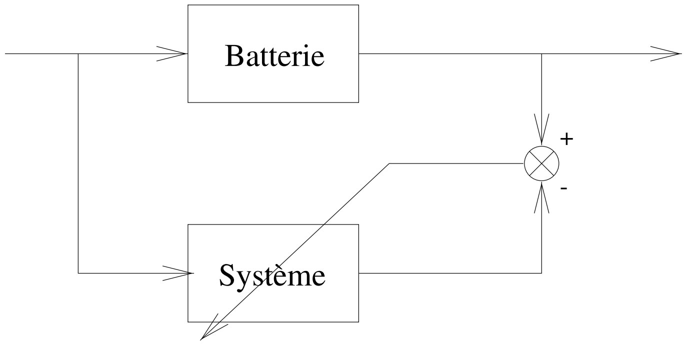

FIG. 4.4: *Le principe de l'identification adaptative directe.*

Ces théories de l'automatique et de l'identification sont un autre champ d'étude très dynamique dans les méthodes en ligne (on pourra en trouver une revue dans [Goutte96]). Parmi les nombreuses techniques proposées, les travaux de A. Levin et K. Narendra (voir [Narendra90], [Levin93] et [Levin95]) font référence. Le principe, fondé sur la formalisation des systèmes dynamiques telle que présentée dans la partie 3.6, repose sur la modélisation d'un système par les équations à états suivantes :

$$\left\{ \begin{array}{rcl} X_{k+1} & = & f(X_k, U_k) \\ Y_k & = & h(X_k) \end{array} \right. \tag{4.2}$$

Cependant, la plupart des systèmes proposés ne fonctionne que sous des hypothèses très contraignantes comme la linéarité des fonctions f et h (filtre de Kalman, voir la partie 3.6), la dimensionnalité limitée à 1 pour les entrées comme les sorties (systèmes SISO : *single input, single output*) et surtout l'observabilité des variables d'états X_k. Ce n'est que récemment que de nombreux chercheurs ont proposé de traiter le problème général de (4.2). Dans le cas où les variables d'états ne sont pas mesurables, il existe deux méthodes pour identifier le système par des techniques neuronales, comme présentées dans la figure 4.5.

La première consiste à coller au plus près à la définition du modèle (4.2) et à construire un réseau de neurones pour chacune des fonctions f et h. Le problème est alors de rétropropager l'erreur commise par le réseau de neurones implémentant f dans celui implémentant h (par rétro-propagation dans le temps [Rumelhert86] ou par apprentissage temporel récurrent en temps-réel

(11). Les experts Philips des batteries travaillent actuellement à un tel modèle.

[Williams89]), ce qui conduit à un apprentissage très lourd et très lent à converger. Une solution étendue consiste à remplacer les poids synaptiques par des filtres linéaires auto-régressifs (d'abord proposés dans le cadre des réseaux non bouclés [Wan93] puis dans celui des réseaux bouclés [Aussem95]), c.-à-d. à introduire un codage temporel des activités synaptiques, plus conforme au modèle biologique [Vaucher96]. Cette formalisation permet d'introduire une mémoire interne au réseau.

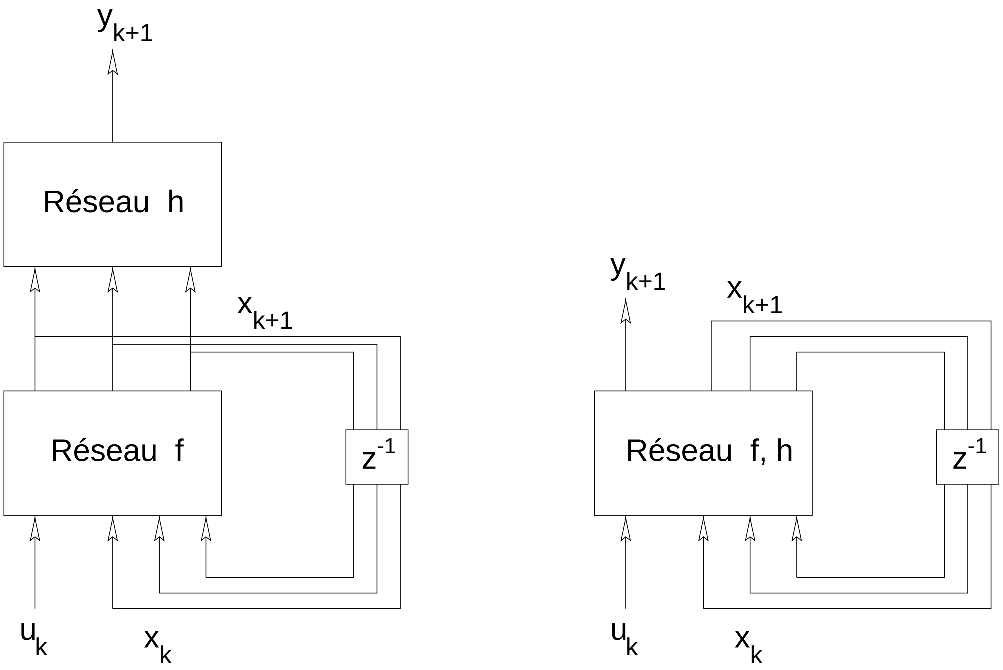

FIG. 4.5: *Exemple de réseau prédicteur d'état : forme canonique d'un réseau bouclé.*

La seconde méthode, la plus courante, cherche à surmonter les problèmes d'apprentissage introduits par la récursion à l'intérieur des réseaux, en dépliant la relation de récurrence sur une fenêtre temporelle de taille fixe (voir [Nerrand93, Rivals95]). Cet « artifice » permet de fournir au réseau une certaine mémoire externe, un certain contexte temporel [Jordan86], à défaut d'une représentation temporelle interne, et au prix d'une limite sur la capacité du réseau à capturer des propriétés temporelles s'étendant sur de longs intervalles de temps. Cette pratique est très utilisée pour le traitement de la parole et l'on pourra se reporter avec intérêt à la bibliographie présentée dans [Stévenin96].

Certains auteurs proposent d'aller encore plus loin et de supprimer toute référence aux variables d'état [Levin95, Levin96]. Ainsi, le système (4.2) s'écrit sous la forme récursive suivante :

$$y_{k+1} = \phi(y_k, y_{k-1}, ..., y_{k-l+1}, u_k, u_{k-1}, ..., u_{k-l+1}) \qquad (4.3)$$

qui a exactement le même comportement entrée-sortie, au prix d'une indétermination des états [12]. C'est cette fonction ϕ que l'on peut tenter d'approcher par réseaux de neurones. Tout comme les travaux sur les séries temporelles cette implantation récursive n'est pas applicable à notre problématique puisqu'elle conduit à de « gros » systèmes difficilement « compréhensibles ».

Cependant, afin d'avoir de relativement bonnes performances pour la modélisation de décharges de batterie, il est nécessaire d'utiliser de l'information contextuelle. Cette information extérieure est même si importante qu'en fait elle conditionne presque totalement la forme de la décharge. Cette forme est décrite dans notre modèle de décharge (4.1) par les paramètres λ alors que les variables décrivant l'environnement ont été regroupées dans φ (voir le paragraphe précédent). Il s'agit donc de modéliser cette influence sur le modèle de courbe, ce que nous faisons par l'introduction d'une fonction $\lambda = h(\varphi)$.

Le système proposé comporte donc deux étages qui fonctionnent selon l'équation suivante [13] :

$$\begin{cases} t &= f_\lambda(V, T) \\ \lambda &= h(\varphi) \end{cases} \tag{4.4}$$

h possède le rôle d'un sélectionneur qui doit choisir la fonction la plus adaptée parmi la famille de fonction f. Ce choix se fait en fonction des variables contextuelles φ. C'est cela qui différencie cette modélisation à étages de celles généralement présentées puisque la variation des paramètres y est le plus souvent autonome, c'est-à-dire sans prise en compte du contexte extérieur.

Le principe de fonctionnement visé est le suivant : l'adaptation doit être faite en fonction du comportement moyen (exprimé par $h(\varphi)$) et du passé de la batterie en cours d'utilisation, afin de mieux prévoir son comportement futur.

On aboutit donc au problème suivant : l'identification d'un modèle dont les paramètres se modifient au cours du temps.

En dehors du domaine de l'automatique, une autre approche envisageable consiste à constituer des groupes de batteries présentant des caractéristiques homogènes, tant pour leur relation avec le monde extérieur que pour leur comportement individuel face au vieillissement. Un premier essai de création de groupes a été effectué en utilisant les cartes topologiques de T. Kohonen [Kohonen82, Kohonen89] et est décrit dans [Gérard95]. L'idée est de contruire des ensembles homogènes de comportement (de paramètres λ) et de définir une courbe de décharge « typique » pour chacun de ces comportements. On peut cependant essayer d'aller plus loin et tenter d'intégrer ces deux étapes en étudiant d'autres modèles compétitifs plus complets (dans le sens où le regroupement ne se fait pas sur des valeurs statiques représentatives du comportement, mais sur des unités capables de calcul).

Modèles compétitifs

Par « modèles compétitifs », nous désignons ce qui est généralement regroupé sous le nom : *mixture of experts*.

(12). Ce modèle NARMA s'appelle série-parallèle ou *teacher forcing model* dans la littérature [Aussem95, p. 97]

(13). Une autre possibilité est d'include T dans φ et de tout recalculer quand T change.

Le principe est simple : plusieurs experts ont en parallèle la même tâche à effectuer. Un module « juge » généralement dénommé *Gating Network* est chargé soit de sélectionner l'expert qui lui paraît le plus pertinent dans la situation actuelle, soit de combiner les réponses de l'ensemble ou d'un sous-ensemble des experts. Le plus souvent le juge se fonde sur les mêmse informations que les experts, mais il peut également fonctionner en n'utilisant que les « scores » des experts. La figure 4.6 présente cette architecture [Weigend95].

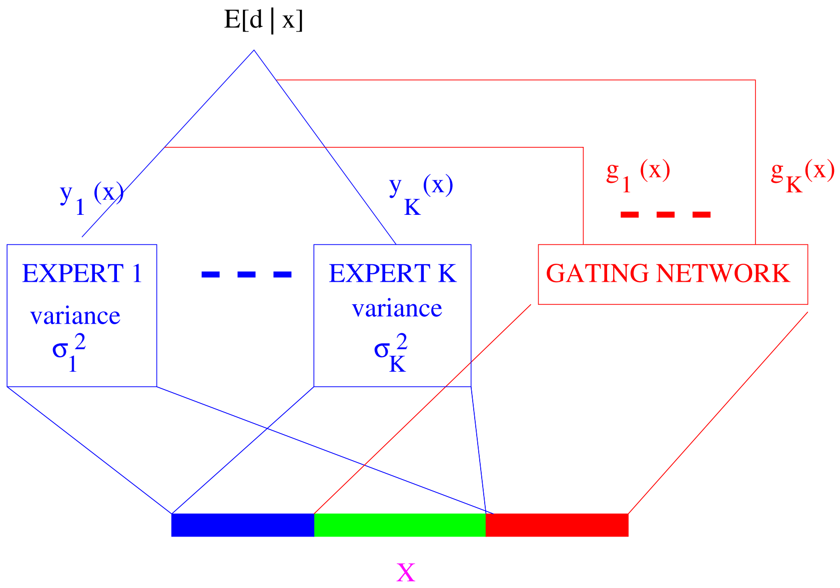

Fig. 4.6: *Architecture des* gated experts.

Dans cette application, les fonctions des experts ainsi que celle du juge sont réalisées par des réseaux de neurones standard (des perceptrons multicouches) (14). Le problème intéressant est l'apprentissage. Au début, les experts ne possèdent aucune expertise et sont indifférenciés. De même, le juge prend ses décisions de façon aléatoire. L'apprentissage, qui s'effectue selon l'algorithme EM, permet dans le même temps la spécialisation des experts et la constitution d'un juge efficace. Cette méthode est utilisée pour découvrir plusieurs modes de fonctionnement au sein d'une série temporelle. Les experts sont particulièrement aptes à travailler avec certains de ces modes et c'est le juge qui est chargé de repérer tout changement de mode, afin de modifier les confiances qu'il porte aux réponses des différents experts.

Ce thème peut être décliné sous de nombreuses formes. Ainsi M. Jordan dans [Jordan95] présente une autre version de cette architecture qui utilise un apprentissage de type montée de gradient et [Meila95] présente une architecture semblable où la fonction « Juge » est réalisée par un HMM.

(14). La sortie du *gating network* est de type *softmax*.

Une extension directe des algorithmes précédents consiste à organiser de façon hiérarchique le groupe experts + juge. Cela donne l'architecture *hierarchical mixture of experts* (HME) de la figure 4.7 qui est tirée de [Jordan94].

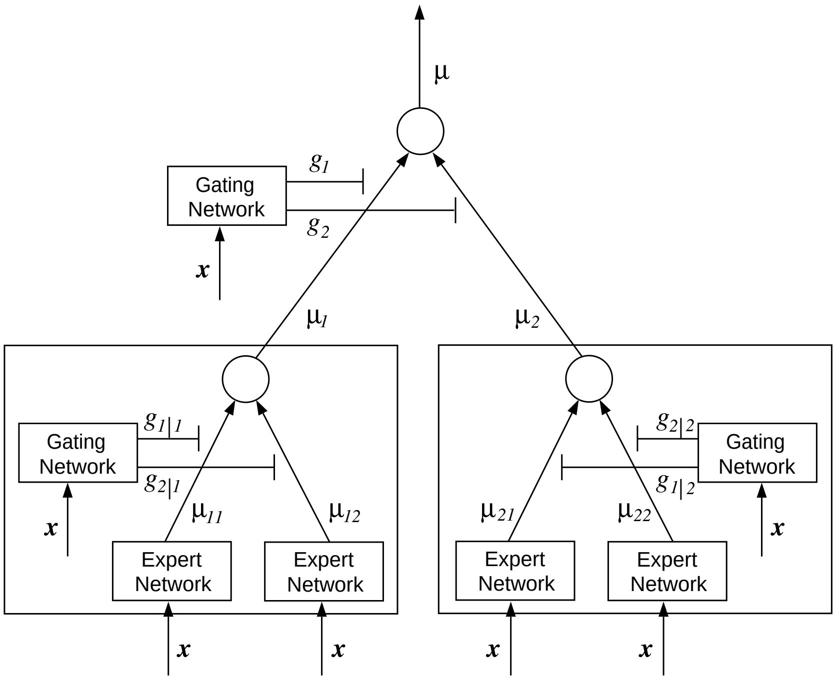

FIG. 4.7: *L'architecture* HME

Chapitre 5

Méthode proposée : l'apprentissage d'un modèle neuronal évoluant au cours du temps

Afin de construire un système capable de prévoir de façon fiable et précise le temps de décharge restant avant l'arrêt de l'appareil portable, le chapitre précédent a décrit les difficultés à surmonter et a proposé une modélisation hiérarchique. Dans ce chapitre, ce modèle hiérarchique sera instanciatié par des techniques capables d'apprentissage à partir d'exemples et cela pour deux raisons essentielles :

- l'absence de modèle mathématique satisfaisant,
- la souplesse des méthodes d'apprentissage.

Après une étude des modèles concurrents envisageables, le modèle original proposé [Gérard99c] utilise donc deux réseaux de neurones organisés de façon hiérarchique et capable de s'adapter aux conditons extérieures. Le premier réseau est un simple modèle d'une courbe de décharge alors que le second est responsable de l'adaptation aux données contextuelles et estime les paramètres du premier réseau. Cette estimation est rendue complexe par la forme même d'un espace de poids synaptiques. Afin de simplifier cet espace, un algorithme d'apprentissage original est proposé [Gérard96]. Un système incrémental est en outre proposé ; il permet une adaptation en ligne aux variabilités comportementales démontrées par les batteries [Gérard97].

5.1 Généralités

Le problème général, tel que décrit dans le chapitre précédent, se présente sous la forme d'une modélisation d'un phénomène physique qui dépend de deux types de variables temporelles dont les variations ne s'effectuent pas sur la même échelle de temps. La première approximation dépend de paramètres qui eux-mêmes évoluent avec le temps. C'est un problème général qui est le plus souvent mal traité, voire ignoré (voir la remarque préliminaire de C.M. Bishop dans la préface de [Bishop95] : « *An important assumption ... is that the processes which give rise to the data do not themselves evolve with time. Techniques for dealing with non-stationary sources of data are not so highly developed, nor so well established, as those for static problems* »). Dans notre cas, l'évolution des paramètres « lents » est guidée par le contexte représenté par les variables φ.

Le système proposé comporte donc deux étages qui fonctionnent selon l'équation suivante [(15)] :

$$\left\{ \begin{array}{lcl} t & = & f_\lambda(V, T) \\ \lambda & = & h(\varphi) \end{array} \right. \tag{5.1}$$

où h possède le rôle d'un sélectionneur qui doit choisir la fonction la plus adaptée parmi la famille de fonctions f. Ce choix se fait en fonction des variables contextuelles φ. C'est cela qui différencie cette modélisation à étages de celles généralement présentées puisque la variation des paramètres y est le plus souvent autonome.

Puisqu'il n'existe pas de modèle fiable de batterie, le système doit se baser sur des composantes issues d'une phase d'apprentissage. Comme les réseaux de neurones à couches sont connus pour leur capacité d'approximateur universel [Hornik89], ce sont de bons candidats pour implanter à la fois les fonctions f_λ et h.

Le § 5.2 décrit les différents tests effectués pour définir la structure du modélisateur de la courbe de décharge (l'approximateur réalisant f_λ). Cette structure doit être habilement choisie puisque d'elle découlent la nature des paramètres λ ainsi que la taille et la complexité de l'espace des paramètres Ω. Le choix final s'est porté sur un réseau de neurones à couches (§ 5.2.3). On a alors un espace de poids synaptiques en sortie du sélectionneur h. Plusieurs techniques ont été utilisées pour réduire la complexité de cet espace. Ces techniques sont l'objet du § 5.2.3.

5.2 Modélisation de la courbe de décharge

La tâche de ce premier modèle est simple : il doit apprendre à approcher une courbe de décharge. En effet, les modèles physiques et mathématiques actuels sont bien trop limités. Le laboratoire de recherche néerlandais de Philips travaille actuellement à une modélisation de batterie sous la forme d'un circuit électrique. C'est un travail en cours auquel nous avons participé en construisant un système

(15). Une autre possibilité est d'inclure T dans φ et de tout recalculer quand T change.

d'optimisation (à base de réseaux de neurones) des nombreux paramètres de ce modèle.

Les courbes que nous cherchons à modéliser sont issues d'un banc de mesure automatique de la société FLUKE [(16)]. Il s'agit bien là d'une limite des méthodes à base d'apprentissage : elles sont généralement assez gourmandes en données. Et dans le cas des batteries, pour avoir une courbe de décharge, il faut attendre 12 h pour la charge puis (après un temps de repos variable) environ 10 h pour la décharge. Néanmoins, tant qu'un modèle physique fiable ne sera pas disponible, seules les techniques d'approximation par apprentissage sont envisageables. Il s'agit alors de bien définir la base de données nécessaire. Au moment de la création de notre système nous disposions de 5200 courbes de décharge pour 20 batteries. C'est sur cette base de données que nous avons testé trois méthodes pour modéliser les courbes de décharge : l'approximation par polynômes de Tchebycheff, les arbres de neurones et les réseaux de neurones.

5.2.1 Les polynômes de Tchebycheff

Un polynôme de Tchebycheff de première espèce $T_n(x)$ de degré n est défini par la relation de récurrence suivante [Laurent72] :

$$\begin{aligned} T_{n+1}(x) &= 2\,x\,T_n(x) - T_{n-1}(x) \\ \text{avec} \;\; T_0(x) &= 1 \;\; \text{et} \;\; T_1(x) = x \; . \end{aligned} \tag{5.2}$$

Si la fonction à approximer $f(x)$ est à valeur dans [-1,1], on définit N coefficients c_j, $j = 0, ..., N$ par

$$c_j = \frac{2}{N} \sum_{k=1}^{N} f(x_k)\; T_j(x_k) \tag{5.3}$$

$$\text{où} \;\; x_k = \cos\left(\frac{(2k-1)\pi}{2N}\right) \tag{5.4}$$

sont les racines du polynôme $T_N(x)$. La fonction f est alors approchée par

$$f(x) \approx \sum_{k=0}^{N-1} c_k\, T_k(x) - \frac{1}{2} c_0 \; . \tag{5.5}$$

Si N est assez grand pour que l'approximation (5.5) soit presque parfaite, l'approximation tronquée jusqu'au degré m (avec $m < N$) est alors la « meilleure » approximation polynomiale de degré m [Press92].

Pour les courbes de décharge, $N = 100$ coefficients sont nécessaires pour avoir une très bonne approximation. Malheureusement, pour avoir une bonne approximation (du même ordre que celle obtenue par réseaux de neurones (§ 5.2.3)), il est nécessaire de conserver $m = 60$ paramètres. L'espace des paramètres Ω est alors trop vaste pour pouvoir être efficacement exploré par le sélectionneur h.

(16). Nous tenons à remercier ici S. Swift and P. Heydron de la société FLUKE pour nous avoir fourni cette base de données.

5.2.2 Les arbres de neurones

Les arbres de neurones sont apparus à la fin des années 80 à la fois comme une amélioration des arbres de décision de la communauté *Machine Learning* [Quilan86] et comme une méthode de construction automatique de réseaux de neurones (voir [Sirat90] et [Marchand90]). Une date importante est 1984 avec la publication du livre *Classification and Regression Trees* de Breiman *et al.* [Breiman84] qui décrit le système CART. Tous ces travaux sont basés sur une architecture hiérarchique d'unités (neurones) très simples qui ne font que des comparaisons scalaires avec un seuil.

Florence d'Alché-Buc a ensuite introduit les arbres de neurones hybrides équipés d'une phase d'apprentissage par trio [d'Alché-93], [d'Alché-94]. Ils sont hybrides car les fonctions calculées par les neurones peuvent être de natures différentes et plus complexes que les simples tests $x_k < S$. L'apprentissage par trio consiste à optimiser plusieurs niveaux hiérarchiques en même temps. Cette méthode permet de construire des arbres beaucoup plus petits et efficaces. Ainsi le problème simulé de [Breiman84] p.216-265 est résolu par CART avec un arbre comportant 12 noeuds alors qu'un seul trio (3 noeuds) donne de meilleurs résultats ($RE^{ts} = 0,14$ au lieu de $RE^{ts} = 0,17$, voir la référence pour la définition de ce critère d'erreur).

Le problème posé consiste en l'approximation des données artificielles Y générées à partir de X_i $i = 1..10$, de la façon suivante :

$$P(X_1 = -1) = P(X_1 = 1) = \frac{1}{2} \tag{5.6}$$

$$P(X_i = -1) = P(X_i = 0) = P(X_i = 1) = \frac{1}{3} \quad \text{i=2..10}$$

$$\begin{cases} Y = 3 + 3X_2 + 2X_3 + X_4 \text{ si } X_1 = 1 \\ Y = -3 + 3X_5 + 2X_6 + X_7 \text{ si } X_1 = -1 \end{cases} \tag{5.7}$$

Le principe général de l'arbre est de partitionner l'espace d'entrée en régions sur lesquelles des réponses différentes sont apportées. Les arbres sont surtout efficaces en classification puisque la régression pose le problème de la continuité aux frontières de ces régions. Afin d'introduire un passage continu entre les régions, un système basé sur deux réseaux de neurones a été construit. Le premier réseau, inspiré du travail de I.K. Sethi ([Sethi90] et [Sethi95]), est chargé d'implanter l'arbre (préalablement construit par l'algorithme du trio). C'est-à-dire que chacune de ses sorties (de type ET) représente une région particulière et est plus ou moins « allumée » (valeur proche de 1) en fonction de l'appartenance du point courant à cette région. Le deuxième réseau est chargé de calculer la sortie globale du système en fonction de l'information topologique issue du premier réseau et de la valeur de l'entrée. Ce deuxième réseau peut être considéré comme un mélange pondéré d'experts. Afin de rendre la réponse encore plus continue, les paramètres de certaines parties des réseaux peuvent être estimées par une phase d'apprentissage. Cela permet d'effectuer une phase d'apprentissage globale afin de peaufiner les résultats obtenus par l'arbre, qui restent malgré tout relativement locaux.

Le principe de combiner phase d'apprentissage global par les réseaux de neurones et phase d'apprentissage local par l'arbre est intéressant mais il est relativement lourd pour une application embarquée.

5.2.3 Les réseaux de neurones

Comme les réseaux de neurones à couches sont connus pour leur capacité d'approximateur universel, ce sont de bons candidats pour implanter les fonctions f_λ. [Cybenko89] et [Funahashi89] ont prouvé la possibilité d'approcher des fonctions continues par des réseaux de neurones, au sens de la norme uniforme sur les compacts. Les réseaux de neurones considérés sont de type réseau à une couche cachée (composée de neurones à fonction d'activation non-linéaire), et à une couche de neurones de sortie linéaires. Dans le premier cas, l'hypothèse sur la fonction d'activation est qu'elle a pour limite 0 en $-\infty$ et 1 en $+\infty$, tandis que dans le second, la fonction doit être croissante et bornée. Les fonctions non continues, mais mesurables, peuvent aussi être approchées, mais dans un sens moins fort [Hornik89].

Suivant ces travaux, la structure choisie pour le réseau de neurones pour approximer les courbes de décharge est la suivante : la couche cachée est composée d'unités à fonction d'activation tanh et une sortie linéaire. L'algorithme d'apprentissage est la rétro-propagation du gradient classique (voir [Hertz91] pour tous les détails).

5200 courbes de décharge ont été ainsi apprises (par 5200 réseaux de neurones) avec une précision moyenne de 3 minutes et une erreur maximale de 15 minutes. Ces erreurs quadratiques sont mesurées sur l'ensemble des courbes de décharge. Un réseau est donc capable d'approcher précisément n'importe quelle courbe de décharge. Si un seul réseau est utilisé pour toutes les courbes de décharge, l'erreur est beaucoup plus importante, de l'ordre de 52 minutes en moyenne. Cela est bien sûr dû à la dispersion des courbes de décharge, telle que présentée dans la figure 4.1, p. 39.

Comme on désire disposer d'un seul modèle pour toutes les décharges, cette unification doit être réalisée par une autre fonction.

On dispose ainsi d'une base de données de 5200 vecteurs de poids correspondant à 260 cycles pour 20 batteries. C'est-à-dire que les réseaux de neurones ont transformé (codé) les courbes de décharges dans une représentation plus compacte, mais relativement complexe à analyser.

Les paramètres λ sont donc des poids synaptiques, ce qui ne va pas sans poser quelques problèmes, vu la complexité d'un tel espace.

Simplification de l'espace de poids

Si $f_\lambda(V, T)$ est calculée par un réseau de neurones, λ sont des poids synaptiques. Un espace de poids synaptiques est très complexe puisqu'il comporte de nombreuses symétries et que les poids qui le composent ne sont pas indépendants entre eux. Le premier type de symétrie est un changement de signes de tous les poids entrant et sortant d'un neurone caché. Ce changement (il y en a 2^M possibles pour un réseau avec une couche cachée de M unités) n'introduit aucune modification à la fonction globale du réseau. Le second type de symétrie est une permutation des positions des neurones (et des poids qui lui sont attachés) de la couche cachée. Au total, outre certaines situations particulières, il y a donc $M!2^M$ possibilités d'implanter exactement la même fonction entrée-sortie par un réseau de neurones avec une couche cachée composée de M unités. Pour tenter de réduire cette complexité, trois méthodes ont été essayées (17).

Vérification des activations

La première solution consiste à vérifier que les activations des neurones de la couche cachée sont « cohérentes ». Au cours d'une décharge, ces activations évoluent. En général, ces neurones sont saturés (activation $= \pm 1$) au début et à la fin de la décharge. Le test consiste à vérifier que ces saturations vont dans le même sens ($+1$ au début et -1 à la fin, par exemple) pour tous les approximateurs de courbe. Lorsqu'un mouton noir est détecté, le réseau correspondant est ré-entraîné après avoir été réinitialisé avec les valeurs des poids obtenues pour la courbe précédente (celle du cycle précédent pour la même batterie). Cela permet de « nettoyer » la base de données des poids en réduisant les symétries liées à la position des neurones cachés. C'est une technique efficace mais elle est un peu excessive, et, surtout, elle ne permet pas la constitution d'une structure de réseau qui empêcherait toute symétrie *pendant* le fonctionnement en ligne.

Réduire la complexité du réseau

Après avoir effectué ce « nettoyage » un peu brutal, on a pu remarquer que les variances des poids de la deuxième couche sont (en moyenne) plus importantes que celles des poids de la première couche ($0,12$ contre $0,06$). Il a donc été décidé de calculer une première couche moyenne et de ré-entraîner tous les réseaux en maintenant cette couche fixe. Cela revient à effectuer une transformation non-linéaire déterministe des entrées. La non-linéarité est alors supprimée des réseaux. Ceux-ci n'effectuent plus qu'une combinaison linéaire de ces entrées modifiées. Les résultats sont relativement bons, c'est-à-dire seulement un peu en dessous de ceux obtenus avec des réseaux complets (augmentation d'environ 10% de l'erreur). Cependant, l'étude de la complexité de cet espace par CTRBF (18) (*Constructive Tree Radial Basis Function*) [Gentric93] ne permet guère de distinguer les réseaux complets des réseaux tronqués (1000 prototypes stockés contre 921). La réduction de l'espace n'a donc pas apporté la réduction escomptée de sa complexité.

Apprentissage itératif des pentes

Un algorithme original d'apprentissage a été développé [Gérard96]. L'idée générale est la spécialisation des neurones par différenciation de leurs pentes internes. On obtient ainsi des neurones qui se spécialisent dans des régions à courbure différente de l'espace d'entrée ; l'objectif étant de réduire les symétries

(17). Ces méthodes ont été appliquées sur les 140 premiers cycles des 20 batteries, soit sur 2800 courbes de décharge.
(18). Outil développé aux LEP.

possibles au sein d'un réseau. L'algorithme développé cherche à optimiser les paramètres de gain g_i introduits dans le calcul de l'activation o_i du neurone i :

$$o_i = f(g_i I_i) = f\left(g_i \sum_{j=1}^{N_i} w_{ij} o_j \right) \tag{5.8}$$

où :

- f est une fonction d'activation non-linéaire (tangente hyperbolique) ;
- N_i est le nombre de neurones j connectés à l'entrée du neurone i ;
- w_{ij} est le poids de la connexion reliant le neurone j au neurone i ;
- I_i est la somme pondérée des entrées : $I_i = \sum_{j=1}^{N_i} w_{ij} o_j$.

g_i est alors un paramètre de gain qui est relié au degré d'inclinaison de la fonction non-linéaire d'activation. Plus g_i est grand, plus la fonction réalisée par le neurone i se rapproche d'une fonction échelon :

$$S(x) = \begin{cases} 1 & \text{quand } x \geq 0 \\ -1 & \text{quand } x < 0 \end{cases} \tag{5.9}$$

Par référence à la physique statistique, g est aussi vu comme l'inverse d'un facteur de *température*. Ce facteur de gain peut être modifié afin de l'adapter à chacun des neurones. Une première méthode simple a été proposée dans [Looney96]. Il s'agit d'un raffinement des pentes par essais - erreurs. Une constante est ajoutée aux pentes. Si la modification permet de diminuer l'erreur, elle est conservée ; dans le cas contraire, elle est supprimée. C'est une procédure simple mais peu efficace.

Une méthode plus intéressante découle de la lecture de l'équation (5.8) qui montre que les facteurs g_i peuvent être considérés comme des poids. Une descente de gradient par rapport à ce paramètre g_i peut ainsi être calculée. Cette approche a d'abord été proposée dans [Movellan87] et elle est l'objet principal de [Kruschke91] et [Zurada93]. [Makram-Ebe89] l'utilise également.

Il s'agit de dériver l'algorithme de rétropropagation en partant de $\partial E/\partial g_i$ au lieu de $\partial E/\partial w_{ij}$. E est la fonction de coût à minimiser. Dans ce qui suit on prendra $E = \sum_k (S_k - Y_k)^2$, soit l'erreur quadratique entre les sorties calculées par le réseau (S_k) et les sorties attendues (Y_k).

Un calcul semblable à celui de la rétro-propagation conduit à la modification suivante des pentes :

$$\Delta g_i = -\eta \, I_i \, d_i \tag{5.10}$$

où η est un gain d'apprentissage. En outre, si le neurone i est une sortie :

$$d_i = 2 \, f'(g_i I_i) \, (S_i - Y_i). \tag{5.11}$$

S'il appartient à une couche cachée, on a :

$$d_i = f'(g_i\, I_i) \sum_h d_h\, g_h\, w_{hi} \tag{5.12}$$

où h parcourt les neurones vers lesquels i envoie une impulsion.

Cet algorithme a été utilisé dans [Kruschke91] pour entraîner des réseaux en même temps que l'algorithme de rétro-propagation classique sur les poids. C'est-à-dire que les pentes sont traitées comme des nouveaux poids. Tous ces paramètres sont modifiés en même temps. Les résultats sont intéressants avec une accélération importante de la convergence (d'un facteur 2,2 pour le XOR), une normalisation automatique des poids et la possibilité d'ajouter certains termes à la fonction de coût pour contraindre l'évolution des pentes. Cette dernière fonctionnalité est justement recherchée par certains auteurs. Face à certains types de problèmes, il est intéressant d'imposer une variation des pentes. La première idée est de faire tendre ces pentes vers de grandes valeurs afin de donner à la fonction d'activation une forme facilement réalisable en électronique telle que la fonction seuil (5.9). Il s'agit alors de forcer l'évolution des pentes selon un schéma prédéfini. À chaque changement de valeur de pente, les poids sont réappris et ceci de façon itérative. Une telle procédure est décrite dans [Corwin94].

La même idée de modifications guidées peut être utilisée dans le but d'ajouter petit à petit de la non-linéarité au sein des fonctions d'activation. Cette approche est assez courante et on peut trouver des exemples d'applications dans [Codrington94] et [Yu94].

Une autre étude réalisée au LEP [Makram-Ebe89] montre tout l'intérêt de la diminution automatique de la température (inverse du gain) des fonctions d'activation. Cette modification permet une accélération importante de la convergence. Elle permet aussi d'obtenir de meilleures performances en généralisation, surtout pour des problèmes de classification.

Cependant, modifier une pente a un effet beaucoup plus important que modifier un poids puisque toutes les normes des poids des connexions entrantes au neurone sont modifiées. *A priori*, il semble donc qu'effectuer les modifications des poids et des pentes en même temps soit un peu trop brutal. De plus, ce qui nous intéresse ici, ce n'est pas l'optimisation globale du réseau mais plutôt l'optimisation de sa structure, via les gains. L'application de cet algorithme à certaines courbes va donc servir à spécifier la *structure* du réseau qui plus tard sera utilisé pour approximer toutes les courbes.

L'algorithme choisi est donc de type itératif. Il est schématisé dans la figure 5.1. Dans cette figure, un nombre indicatif d'époques est aussi donné, ainsi qu'un ordre de grandeur du nombre d'itérations de la boucle (10x).

Une descente de gradient sur les poids est d'abord effectuée (donnant le résultat $\mathcal{A}$), puis l'algorithme entre dans un cycle : apprentissage des poids puis des pentes. La descente de gradient sur les pentes est en général plus courte afin de ne pas trop perturber le réseau. La sortie de la boucle (résultat $\mathcal{B}$) est déterminée par un seuil sur l'erreur commise (ainsi que le nombre maximum d'itérations). Afin de valider les résultats, les poids du réseau sont réinitialisés (les pentes gardant leur valeur issue de l'apprentissage) et un apprentissage standard des poids est effectué (résultat $\mathcal{C}$). On vérifie ainsi que les pentes modifiées

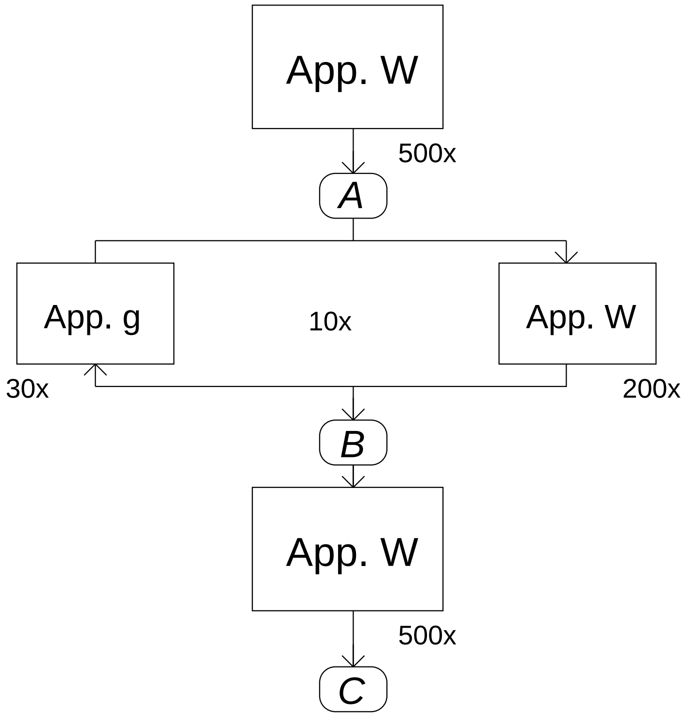

FIG. 5.1: *Algorithme d'apprentissage itératif des pentes : synoptique.*

permettent un meilleur apprentissage. En outre, certaines pentes peuvent devenir presque nulles durant l'apprentissage. Les neurones correspondant sont alors inutiles et ils sont supprimés. L'algorithme effectue donc également une optimisation de la taille du réseau, ce qui est important pour réduire la complexité du système en vue de son implantation dans des appareils portables. À titre

Résultat	en $\mathcal{A}$	en $\mathcal{B}$	en $\mathcal{C}$
Nombre de neurones	5	2	2
Erreur (en %)	6,44	0,74	0,96
Variance	7.10^{-2}	8.10^{-3}	1.10^{-2}

TAB. 5.1: *Résultat de l'approximation de la courbe $y = x^3$ par la méthode d'apprentissage itératif des pentes.*

d'exemple, la table 5.1 présente les résultats obtenus par cet algorithme itératif pour l'apprentissage de la fonction $f(x) = x^3$. Alors que la rétro-propagation classique (point $\mathcal{A}$) restait bloquée dans un minimum local, le mouvement des pentes a permis de l'en sortir (point $\mathcal{B}$). Les valeurs des pentes obtenues sont intéressantes en elles-mêmes puisqu'une rétro-propagation classique sur un réseau les utilisant (point $\mathcal{C}$) donne de meilleurs résultats qu'en $\mathcal{A}$ (et ceci avec 3 neurones de moins sur la couche cachée). Il existe de légères différences entre les résultats en $\mathcal{B}$ et en $\mathcal{C}$ prouvant que l'indépendance poids-pentes n'est pas parfaite. Mais les résultats obtenus en recopiant les pentes (en $\mathcal{C}$) restent très bons.

On peut aussi remarquer dans (5.10), (5.11) et (5.12) qu'il existe un certain rapport entre la pente g et le gain d'apprentissage η. Une étude de ce phénomène par [Thimm96] a montré que si la pente passe de 1 à β, alors η est divisé par β^2. Notre algorithme permet donc d'avoir un pas d'apprentissage différent et variable pour chacun des neurones du réseau. D'autre part, il existe un lien direct entre g et la norme des poids. En fait, [Kruschke91] a montré que les gains adaptatifs accélèrent l'apprentissage en modifiant la longueur, mais pas la direction, des vecteurs poids.

Cette méthode a été utilisée à la fois pour f_λ et pour h. En ce qui concerne f_λ, elle a permis de réduire la taille du réseau utilisé pour approximer toutes les courbes de décharge. Le réseau final (NN1) ne comporte plus que 2 neurones sur la couche cachée, c'est-à-dire 9 poids. La figure 5.2 présente les valeurs de deux

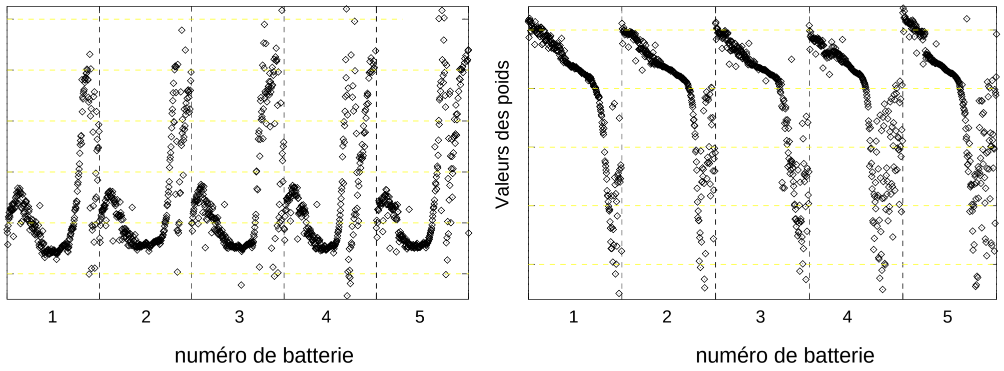

FIG. 5.2: *Les valeurs de deux poids de* NN1 *pour 260 cycles de 5 batteries de test ; chaque point est le résultat d'un apprentissage neuronal.*

des poids de NN1 obtenus pour 260 cycles pour 5 batteries. Dans cette figure, chaque point est le résultat de l'apprentissage d'un réseau NN1. On peut voir que l'apprentissage est « doux » puisque les poids appris dans des conditions proches ont des valeurs proches. Il est aussi « propre » puisque le codage du comportement (vieillissement) par ces poids présente la même régularité pour toutes les batteries (sauf peut-être à la fin de leur vie).

La fonction h est calculée par un deuxième réseau de neurones : NN2 qui comporte 5 unités d'entrées, 3 neurones cachés et 9 sorties (les 9 poids du réseau NN1).

5.3 L'adaptivité aux comportements individuels

Les batteries présentent d'importantes différences individuelles : deux batteries issues du même lot de fabrication peuvent avoir des temps de décharge qui diffèrent de ±15 %. En outre, plus les batteries vieillissent, plus ces différences peuvent s'accroître et après un temps de fonctionnement identique, certaines batteries peuvent durer deux fois plus longtemps que d'autres. Il faut donc tenir compte de cette diversité individuelle pour avoir une prévision fiable.

Le principe de fonctionnement visé est le suivant : l'adaptation doit être faite en fonction du comportement moyen (exprimé par $h(\varphi)$) et du passé de la batterie en cours d'utilisation, afin de mieux prévoir son comportement futur.

Comme le système adaptatif a de relativement bonnes performances, il doit servir de base à la constitution du nouvel algorithme. Bien évidemment, NN1 a un rôle plutôt passif. C'est donc sur NN2 que doit porter le travail de l'adaptation au comportement individuel. D'autres considérations vont intervenir dans le choix de l'algorithme, comme les limites inhérentes à l'aspect embarqué, la flexibilité et l'utilisation du modèle de comportement appris par NN2.

Deux voies sont alors envisageables : l'apprentissage en ligne ou l'adjonction externe de nouvelles capacités. Une étude de la littérature est présentée dans les sections suivantes pour ces deux axes de recherche.

5.3.1 Apprentissage en ligne

L'apprentissage en ligne consiste à modifier les poids d'un réseau à chaque fois qu'un nouvel exemple lui est présenté. Le réseau doit être capable d'acquérir la nouvelle information au bout de quelques exemples sans détruire les connaissances déjà acquises. C'est le dilemme stabilité-plasticité exprimé pour la première fois par S. Grossberg [Grossberg76].

L'approche la plus complète et précise du domaine est celle poursuivie dans [Biehl94] et par D. Saad et S. Solla ([Saad95a], [Saad95b], [Solla96] et [Solla95]). Ces travaux décrivent de façon exacte (sous forme d'équations différentielles) l'évolution d'un réseau soumis en continu à un flot de données et dont les paramètres sont modifiés à chaque exemple. La technique consiste à faire apprendre à un réseau « esclave » les données issues d'un réseau « maître », et à calculer l'évolution des paramètres du réseau esclave au cours de l'apprentissage, en fonction des paramètres du réseau maître. Ces travaux permettent d'écrire de façon explicite l'évolution des poids d'un réseau au cours de son apprentissage. Cependant, cela reste surtout une approche théorique peu appliquée et peu applicable car elle nécessite de connaître le réseau maître qui a généré les données que l'on cherche à apprendre.

D. Park [Park91a] propose une méthode pour adapter les poids d'un réseau (ayant déjà appris) à des données issues d'un processus évoluant lentement dans le temps. L'objectif est proche de celui que l'on s'est fixé mais les méthodes employées ne sont pas compatibles avec un usage embarqué. L'algorithme proposé nécessite en effet le calcul de matrices de Hessien de relativement grande dimension. Un algorithme plus intéressant a été proposé dans [Ziegler92] : il s'agit d'introduire un terme de plasticité pour chaque poids du réseau. Les poids stabilisés représentent l'information déjà acquise. L'apprentissage en ligne doit donc en priorité porter sur les poids encore « instables ». Ce terme, calculé durant l'apprentissage, est une mesure locale de la stabilité du poids. La méthode est performante puisqu'elle permet un apprentissage rapide sans la contrainte du stockage de l'ensemble des exemples d'apprentissage. Elle possède cependant deux inconvénients au vu de notre application : la possibilité de modifications très importantes des poids ainsi que l'obligation de doubler le nombre de paramètres du modèle.

H. Chen et V. Soo ont publié à ce jour trois articles ([Chen92], [Chen93] puis [Chen95b]) sur l'apprentissage adaptatif en ligne. Leur principale idée est d'ajouter un terme à la fonction de coût pour forcer le réseau à bien répondre au dernier exemple présenté. Cette méthode ne permet pas d'apprendre correctement la totalité de la base d'apprentissage, cependant elle est efficace dans les situations où l'environnement évolue de façon lente et irrémédiable. Leur deuxième publication [Chen93] introduit une normalisation en ligne des données d'entrée ainsi qu'une partie supplémentaire spécifiquement chargée des données incohérentes. Lors de l'apparition de tels nouveaux exemples (qui sont détectés automatiquement), le réseau doit être capable de modifier de façon importante ses paramètres internes. Le nécessaire ré-apprentissage est alors effectué à partir du nouvel exemple et de données re-synthétisées par le réseau, compte tenu du fait que les exemples d'apprentissage passés ne sont pas sauvegardés. Cependant, cette étape de re-synthétisation d'une base d'apprentissage n'est pas explicitée de façon suffisamment claire. Leur troisième article reprend les idées exposées dans [Ziegler92].

Il existe de très nombreux autres travaux sur l'apprentissage en ligne. Cependant, la plupart d'entre eux (comme [de Angulo95]) ne travaille pas sur la règle d'apprentissage classique de la rétro-propagation. Ceci pose un problème de cohérence des algorithmes et donc de la taille du logiciel à embarquer dans l'appareil.

Les théories de l'automatique et de l'identification sont un autre champ d'étude très dynamique dans les méthodes en ligne (on pourra en trouver une revue dans [Goutte96]). Parmi les nombreuses techniques proposées, les travaux de K. Narendra (voir [Narendra90], [Narendra95] et [Levin96]) font référence (voir le chapitre précédent). Parmi les nombreux travaux publiés à ce sujet, [Hao94] traite d'un problème très proche du nôtre : le système possède une phase d'apprentissage hors ligne utilisant la rétro-propagation du gradient ; l'apprentissage est ensuite raffiné en ligne en adaptant les paramètres du réseau au système courant. Ce travail présente des résultats intéressants même dans le cas où la sortie du système est modifiée au cours de l'apprentissage en ligne. Bien que les résultats publiés jusqu'à présent soient prometteurs, ils sont obtenus avec un réseau ne comportant qu'une entrée et qu'une sortie. Ce travail peut néanmoins nous être utile pour définir notre moyen d'adaptation puisque la méthode utilisée est très simple : elle est exclusivement basée sur l'apprentissage par rétro-propagation.

Un autre champ d'étude est constitué des modèles hiérarchiques dynamiques, qui sont des extensions des modèles hiérarchiques décrits dans 3.7.

5.3.2 Apprentissage incrémental

L'intérêt pour les méthodes incrémentales n'est pas nouveau et de nombreux algorithmes ont été proposés. Ce type d'algorithme tente en fait de répondre à deux exigences disctinctes. La première entièrement hors-ligne, de type **constructive**, cherche à construire un système dont la complexité est directement adaptée au problème courant à traiter. La seconde, plus proche de l'apprentissage en ligne, cherche à complexifier le système à chaque apparition d'une situation nouvelle. La principale différence d'implantation entre ces deux approches réside dans la nécessité de stocker (et de faire défiler à nouveau) l'ensemble de la base de données d'apprentissage à chaque complexification du système. La plupart des méthodes constructives s'attache à l'optimisation de la *structure* du modèle et s'autorise donc de tels calculs. Ces méthodes sont alors surtout intéressantes pour leur étude du critère de nouveauté, mais c'est surtout dans le deuxième groupe (beaucoup moins étudié car beaucoup plus difficile) que nous allons chercher une solution à notre problème. L'idée, en effet, est d'ajouter des unités de calcul à chaque fois que le comportement prévu n'est pas suffisamment proche de la réalité et d'effectuer un apprentissage en ligne des paramètres de cette unité en plus. Le risque d'un apprentissage par cœur de la dernière situation ne pose pas trop de problèmes dans notre cas puisque les paramètres évoluent à peu près dans la même direction.

La première classe de ces algorithmes incrémentaux est constituée de méthodes « constructives ». L'incrémentalité tente alors de répondre au problème du sur-apprentissage ; lorsqu'un système à apprentissage comporte de (trop) nombreux paramètres, il va tenter de reproduire de façon (trop) fidèle les données d'apprentissage. Les performances en généralisation seront alors faibles. La première méthode pour traiter ce problème est la suppression de paramètres (voir [Leray98] pour une présentation des méthodes existantes). L'autre méthode est la construction d'un système adapté à la tâche à effectuer. J. Elman a écrit un rapport intéressant dont le titre est éloquent : *Incremental Learning or the importance of starting small* (19). Dans ce rapport, il montre que l'apprentissage incrémental n'est pas seulement utile, mais aussi nécessaire à l'acquisition de concepts complexes. Avant de pouvoir appréhender un nouveau concept dans la totalité de sa complexité, il faut d'abord être capable de traiter des sous-tâches. La progression se fait alors brique par brique. Il démontre ces liens en prenant l'exemple du temps d'attention des enfants qui s'allonge au fur et à mesure de leur évolution : au début, ce temps est très court et les problèmes complexes (les phrases longues) ne peuvent être représentés ; seules les phrases courtes et simples peuvent être traitées. Cela permet aux jeunes de ne pas être noyés dans un flot d'information.

Le réseau doit donc commencer *ex-nihilo* puis construire sa structure au cours de l'apprentissage, lors de la présentation de nouveaux exemples. Cette méthode est la base des algorithmes *Tiling* [Mézard89], *Upstart* [Frean90] et *Tower and Pyramid* [Gallant86]. Ils génèrent des réseaux avec des topologies différentes et donc des critères de décision différents. Les comportements de ces trois algorithmes sont étudiés et comparés dans [Chen95a]. Cependant, ils sont tous composés d'unités linéaires ou à seuil, ce qui ne peut être utilisé dans notre cas. En outre, ils sont souvent limités à des entrées-sorties de type binaire ou vecteur de bits. Il existe beaucoup d'autres méthodes dans la littérature. Cependant, elles partagent toutes la même caractéristique : elles ne fonctionnent que pour des tâches de classification. Ces tâches ont en effet plusieurs caractéristiques utiles pour ce type d'algorithmes : un exemple nouveau est facilement détectable, les seuils pour déterminer les zones de confusion sont dérivés de façon

(19). L'apprentissage incrémental ou l'importance de débuter petit.

naturelle et elles se prêtent facilement aux traitements en parallèle.

Une seconde classe de méthodes incrémentales est constituée par l'architecture *Cascade-Correlation* introduite par S. Fahlman [Fahlman90]. Comme dans les méthodes décrites précédemment, le réseau se construit itérativement à partir d'une structure ne comportant pas de neurone caché. À chaque fois qu'une amélioration des performances est ressentie nécessaire, une nouvelle unité cachée est introduite. Cette unité est plus spécifiquement chargée d'apprendre le nouvel exemple ; plus précisément, elle doit apprendre la différence entre la précédente réponse du réseau et la nouveauté présente dans les exemples de l'instant. Cet algorithme a de nombreux avantages : les neurones ne sont pas forcément linéaires, les entrées et sorties peuvent être scalaires et l'apprentissage est rapide puisque de nombreux poids sont fixés pendant l'apprentissage. La méthode possède donc des caractéristiques intéressantes mais la totalité des exemples d'apprentissage doit être sauvegardée pour être présentée au réseau à chaque nouvel apprentissage. Ce n'est donc pas une méthode véritablement en ligne. Elle n'est donc pas envisageable pour une application embarquée.

La dernière catégorie de méthodes se compose de la longue série des algorithmes ART (*Adaptive Resonance Theory*), *Adaptive Resonance Theory/textbf.* ART a d'abord été développé par S. Grossberg dans [Grossberg76] comme une tentative de réponse au dilemne stabilité-plasticité. Ce dilemne consiste à essayer de concilier deux exigences très différentes. La première vise à construire un système stable, aux propriétés de convergence bien établies alors que la seconde s'attache à définir un système malléable, capable de s'adapter rapidement à de nouvelles situations dans un contexte en évolution.

[Moore98] est une introduction à ART 1 et [Carpenter88] contient un récapitulatif des différents modèles ART. ART 1 regroupe des vecteurs binaires. Dans sa forme initiale, il était entièrement non-supervisé. ART 2 travaille avec des vecteurs d'entrée scalaires et est donc beaucoup plus intéressant. La première version de ART 2 est présentée dans [Carpenter87]. Dans son mode non-supervisé, il effectue la même tâche que l'algorithme de Kohonen : le regroupement sélectif des exemples. ART 2 possède également un mode supervisé et une version incrémentale. Cette dernière version ajoute une unité interne à chaque fois qu'un exemple inconnu est présenté et que la modification de l'existant résulterait en la destruction des exemples déjà appris. Cette caractéristique est très intéressante pour l'application aux batteries. Cependant les algorithmes ART sont relativement complexes, en tout cas bien trop lourds pour pouvoir être implantés sur un appareil portable.

Le dernier algorithme étudié est le CTRBF : *Constructive Tree Radial-Basis-Function*. Cette méthode a été développée aux LEP par Ph. Gentric et H. Withagen, elle a été appliquée à un problème de reconnaissance de l'écriture en ligne [Gentric93]. Tout comme ART, CTRBF possède un mode non-supervié et un mode supervisé. C'est ce dernier mode qui nous intéresse ici. Chaque classe est définie par autant de RBF que nécessaire. Une unité RBF (*Radial Basis Function*) est déterminée par deux paramètres : la position de son centre et son rayon. La position est un vecteur w de dimension d et le rayon r décrit la taille de l'espace autour de son centre qui doit intéresser la RBF. Une telle unité calcule sa sortie y en fonction du vecteur d'entrée x (de dimension d) selon une gaussienne :

$$y = \frac{1}{\sqrt{(2\pi)^d \,|\Sigma|}} \exp^{-\frac{1}{2}(x-w)^T \Sigma^{-1}(x-w)} \tag{5.13}$$

où Σ est une matrice $d \times d$ diagonale (indépendance des dimensions) qui représente la forme de la zone active pour les différentes directions. Du point de vue pratique – et comme cela se pratique couramment – la matrice Σ sera la matrice identité ($\Sigma = I_{d,d}$, toutes les variances sont égales) et le terme de normalisation, constant pour toutes les unités, sera négligé. La méthode générale est proche de celle des réseaux RBF [Park91b]. Ces réseaux neuronaux possèdent deux couches, la première (la couche cachée) est composée d'unités RBF, chacune active et prenant une décision dans une partie de l'espace d'entrée alors que la couche de sortie (généralement composée d'unités linéaires) rassemble les activités des unités cachées afin de prendre une décision globale. Dans le cadre de la classification, cette décision est généralement prise conformément au choix exclusif de l'unité gagnante, celle qui est la plus active [(20)]. La principale différence entre les réseaux RBF et l'algorithme CTRBF réside dans la structure interne de ce dernier, qui au lieu de constituer une couche d'unités en parallèle, cherche à utiliser la caractéristique hautement locale des unités RBF pour constituer un arbre qui divise l'espace d'entrée en zones de plus en plus petites. Cet arbre économise de très nombreux calculs de cellules RBF, puisque seules les cellules concernées seront activées. Mais l'intérêt principal de cette méthode pour notre problème est son caractère incrémental. Chaque fois qu'une nouvelle situation est rencontrée, l'algorithme la sauvegarde comme centre d'un nouveau noyau et modifie ainsi la réponse globale du système.

5.3.3 La solution retenue

L'idée est donc d'ajouter à NN2 (le réseau calculant la fonction h) un module additionnel qui sera modifié en ligne de façon à tenir compte du comportement de *la* batterie en cours d'utilisation. Comme les contraintes en termes de mémoire sont strictes, le stockage de données doit être réduit le plus possible. Une façon élégante de résoudre le problème est d'effectuer une rétro-propagation à la fin de la décharge, en comparant la courbe mesurée et celle prédite par h(NN2). Plus précisément, le nouveau système incrémental est décrit dans (5.14) et sous forme schématique dans la figure 5.3. Pour des raisons d'implantation, nous avons choisi d'isoler la partie incrémentale du reste du système en charge du comportement moyen, mais cette méthode est équivalente à l'ajout de nouvelles capacités de calcul pour h (de nouveaux neurones dans la couche cachée de NN2).

$$\left\{ \begin{array}{rcl} t & = & f_\lambda\,(V, T) \\ \lambda & = & h\,(\varphi) + \lambda_k \\ \lambda_k & = & k\,(\varphi) \end{array} \right. \tag{5.14}$$

(20). Cette règle est souvent appelée WTA pour *winner take all*.

$k(\varphi)$ est une unité qui est chargée de prendre en compte les différences entre le comportement moyen (estimé par h) et le comportement individuel de la batterie en cours d'utilisation. Cette unité est en fait un troisième réseau de neurones (NNA) réduit à un seul neurone sur la couche cachée, ce qui est équivalent à ajouter un neurone à la couche cachée de NN2. Au début, ce neurone est inactif.

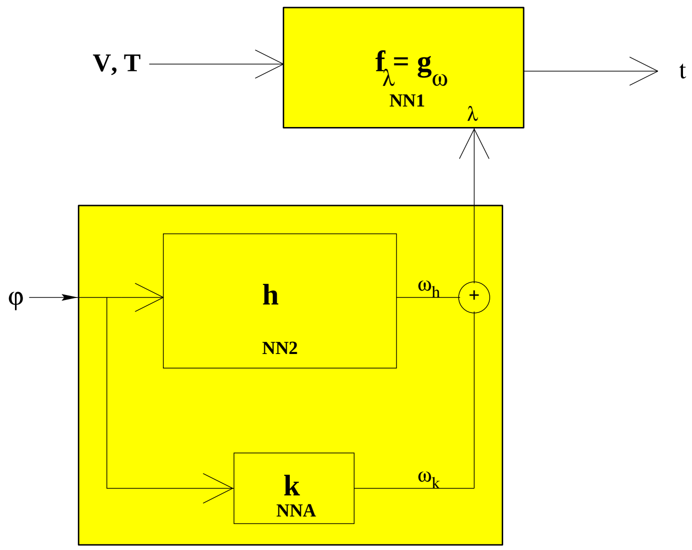

FIG. 5.3: *Schéma synoptique de l'algorithme incrémental.*

Ses poids sont modifiés en ligne de manière à réduire la différence entre la courbe prédite f_λ et celle qui a été mesurée. L'apprentissage s'effectue de la façon décrite dans la figure 5.4, en la lisant de haut en bas. Durant la décharge, les valeurs décrivant la courbe (V, T et t) sont enregistrées. La courbe est ensuite présentée à un réseau NN1 qui est chargé de l'apprendre. L'apprentissage, de type rétro-propagation du gradient, permet d'estimer les valeurs des poids λ^* qui auraient dû être données par la fonction f_λ (le réseau NN2). La différence entre ces valeurs obtenues *a posteriori* λ^* et celles *a priori* λ donne λ_k^*, c.-à-d. les valeurs qui auraient dû constituer les sorties de NNA. À partir de ces sorties λ_k^* et des valeurs de contexte φ, un très court apprentissage de NNA est effectué.

En fait, l'apprentissage de NNA est constitué d'un seul passage du seul exemple disponible. Il agit exactement comme un régulateur. Pour en illuster le fonctionnement, la figure 5.5 présente les prévisions obtenues de façon standard et adaptative pour une batterie particulière tout au long de sa vie.

Au début, les deux prévisions sont assez proches. Au cycle 56, la batterie est « perturbée ». Le système incrémental met environ 10 cycles pour recoller aux temps mesurés alors que le système standard commet de plus grosses erreurs. De même, les prévisions incrémentales sont meilleures à la fin de la vie de la batterie, bien que les deux prévisions soient assez instables. Les résulats obtenus sur ce cas particulier de batterie montre bien la potentialité de l'algorithme incrémental à « poursuivre » la batterie en cours d'utilisation. Une analyse quantitative des résultats est présentée dans le chapitre suivant.

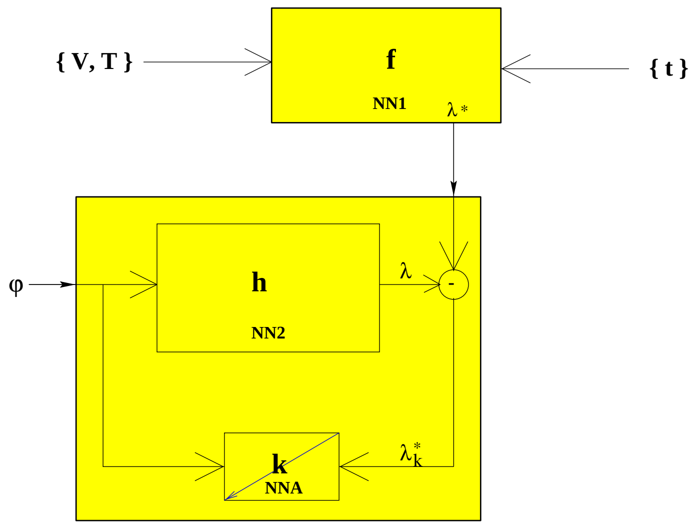

FIG. 5.4: *Schéma synoptique de l'apprentissage avec l'algorithme incrémental.*

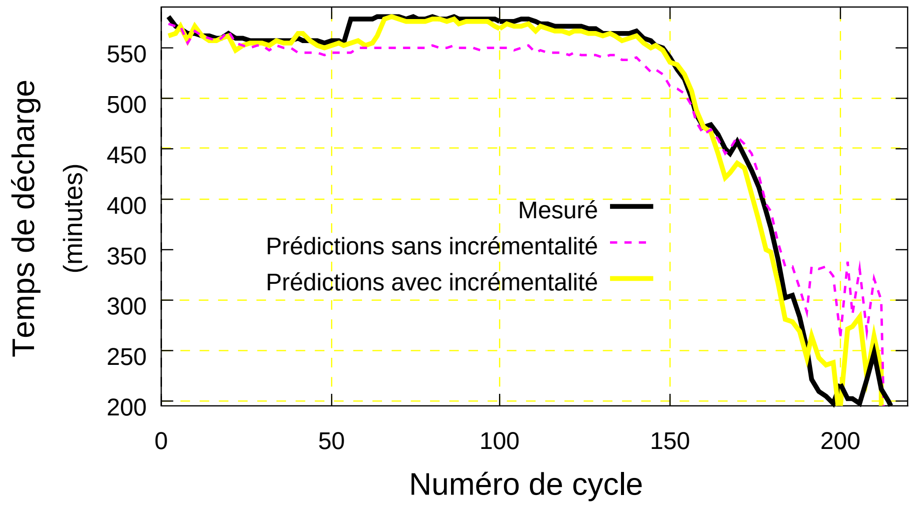

FIG. 5.5: *Prévisions adaptatives (non incrémentales) et incrémentales pour une batterie de test.*

Chapitre 6

Résultats, conclusion et discussion

Le système proposé dans le chapitre précédent est en fait double. Le système adaptatif standard propose une prise en compte d'informations contextuelles pour estimer les valeurs des paramètres d'un modèle neuronal de courbes de décharge. Afin de tenir compte des comportements individuels prononcés des batteries, un système incrémental a été construit. Il permet de s'adapter en ligne à la variabilité comportementale, vis-à-vis de la réponse à l'environnement.

Les résultats obtenus sont présentés dans ce chapitre ; ils sont relativement bons, et en tout cas bien meilleurs que les spécifications demandées. L'incrémentalité permet d'obtenir des performances encore meilleures en s'adaptant à la batterie en cours d'utilisation. Ce système adaptatif a été implanté sur un prototype d'un multimètre portable, tel que décrit dans la section 6.2.

Ce chapitre se termine par une présentation des nombreuses possiblités d'utilisation de la formalisation proposée pour la résolution de très nombreux problèmes existant dans des domaines très variés.

6.1 Résultats

Les résultats obtenus par les deux systèmes (adaptatif standard *i.e.* sans incrémentalité, et incrémental) sont bons : l'erreur moyenne est proche de 10 min pour un événement qui peut intervenir dans un intervalle de 10 heures.

	Adaptatif	**Incrémental**
Erreur moyenne	3,11 %	1,47 %
Erreur moyenne	14 min	6 min
Variance (x100)	9,78	4,93
Erreur inférieure à 1.0 %	23 %	64 %
Erreur inférieure à 5.0 %	86 %	96 %
Erreur inférieure à 5 min.	22 %	61 %

TAB. 6.1: *Résultats des prévisions adaptatives (non incrémentales) et incrémentales.*

Le système incrémental est plus complexe à mettre en œuvre mais possède de meilleures performances. Les résultats ont été calculés sur les 5200 courbes de décharge de batteries dont la décharge dure plus de 250 minutes, ce qui est moins de la moitié de la durée nominale de ces batteries (10 h.). Ces performances des deux systèmes sont comparées dans la table 6.1. La figure 6.1 compare les prévisions obtenues par le système incrémental avec les temps mesurés. On peut également y voir les lignes correspondant à ±5 % d'erreur.

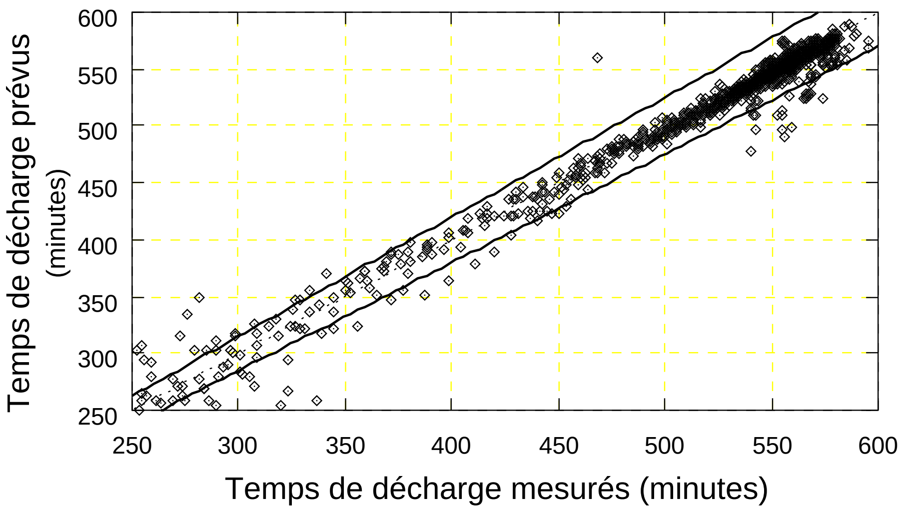

FIG. 6.1: *Prévisions incrémentales en fonction des temps mesurés ; lignes d'erreur à 5%.*

6.2 Implantation

L'ensemble du système a été implanté aux LEP (par Jean-Noël Patillon, en collaboration avec FLUKE) sur un des prototypes de multimètre portable de la société FLUKE (voir figure 4.3, p. 43). L'implantation a été faite de façon à réutiliser le plus possible les composants déjà existants dans le multimètre. Le calculateur embarqué, un Motorola 68306, le convertisseur analogique - numérique ainsi que l'écran LCD du multimètre ont donc été mis à profit. Au niveau composant, seuls un capteur de température (placé près de la batterie) et une horloge (pour avoir des mesures régulièrement espacées durant la décharge) ont dû être ajoutés. Le schéma de l'implantation de l'algorithme sur le prototype du multimètre est présenté dans la figure 6.2. Il est dû à Jean-Noël Patillon [Patillon96a].

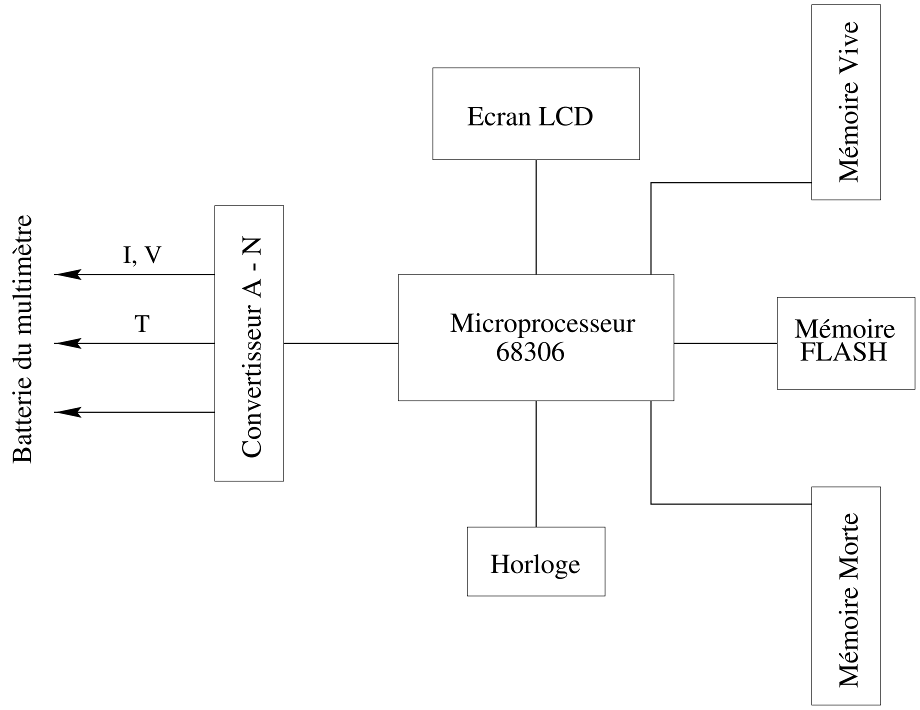

FIG. 6.2: *Schéma de l'implantation de l'algorithme incrémental proposé sur le prototype du multimètre.*

Afin de faire fonctionner l'algorithme de prévision, il a été nécessaire d'ajouter de la mémoire, comme décrit dans le tableau 6.2.

Le comportement moyen de la batterie $h(\varphi)$, estimé par NN2, a été implanté en ROM (mémoire morte, car $h(\varphi)$ est issu d'un processus d'apprentissage hors-ligne effectué préalablement). De la mémoire vive (RAM, plus coûteuse, et qu'il faut alimenter électriquement) est uniquement nécessaire pour sauvegarder le réseau NN1 courant, tel que calculé par NN2. Comme le contenu de toute mémoire vive est effacée à chaque réinitialisation, un moyen de stockage permanent doit être utilisé pour les variables de contexte. La seule solution envisageable est

	Adaptatif		Incrémental	
	Contenu	Taille	Contenu ajouté	Taille totale
ROM	NN2	20 Ko	algo BP	60 Ko
RAM	NN1	1 Ko	Dernière courbe	5 Ko
EEPROM	φ	100 octets	NNA	500 octets

TAB. 6.2: *Mémoire nécessaire pour faire fonctionner le système adaptatif en version non incrémentale et en version incrémentale.*

une mémoire de type « Flash EEPROM ». La quantité de ce type de mémoire (très coûteuse) est très faible, puisqu'elle ne concerne que les variables contextuelles φ, soit environ 100 octets en fonctionnement standard, non adaptatif. Au total, les quantités de mémoire nécessaires pour l'algorithme standard sont donc faibles. Dans le cas du système adaptatif, de la mémoire supplémentaire est nécessaire pour sauvegarder :

- l'algorithme de rétro-propagation du gradient, en mémoire morte ;
- la courbe de tension pour la décharge en cours, en mémoire vive ;
- les paramètres de NNA dans une « Flash EEPROM ».

L'algorithme a été construit de sorte que les calculs les plus lourds (apprentissage par rétro-propagation de NNA) soient effectués pendant la charge de l'appareil, c.-à-d. quand la charge de calcul du micro-processeur est la plus faible.

Cela démontre que les réseaux de neurones peuvent être utilisés dans des appareils de tous les jours. En effet la majorité des appareils portables actuels (rasoirs, téléphones GSM,...) ont des capacités – tant en puissance de calcul qu'en mémoire – suffisantes pour faire tourner ces algorithmes en quasi temps-réel.

6.3 Les améliorations à l'étude

L'implantation sur microprocesseur a amené des questions sur les possiblités de gain, tant en taille de stockage qu'en vitesse de calcul. L'analyse de l'algorithme a montré que le nombre de points stockés pour décrire une courbe de décharge pouvait être grandement réduit. Dans l'implantation actuelle, chaque courbe de décharge est sur-échantillonnée sur 800 points régulièrement espacés. Le sur-échantillonnage est effectué par passage d'une ligne spline. Cette méthode a l'avantage de donner toujours le même nombre de points quelle que soit la courbe, mais au détriment de l'encombrement de la mémoire et de la vitesse d'apprentissage des réseaux de neurones chargés de « coder » les courbes. Ces questions deviennent critiques pour l'implantation sur un microprocesseur.

Comme la méthode hiérarchique exposée se fonde sur le codage des courbes en vecteurs de poids, des courbes se ressemblant vont être codées par des vecteurs proches. Ainsi, le vecteur de poids moyen a une signification en termes de courbe moyenne. Un test préliminaire a donc été effectué en initialisant les réseaux de neurones avec ces poids moyens au lieu de la traditionnelle initialisation aléatoire. La phase d'apprentissage peut alors se concentrer sur sa véritable tâche : capturer la différence entre la courbe présente et le comportement moyen des batteries. Cette tâche étant plus simple, le nombre de points nécessaires pour décrire la courbe peut être réduit de façon drastique sans perte de précision. Ainsi, la figure 6.3 présente les erreurs obtenues lorsque le nombre de points utilisés est réduit de 400 jusqu'à 25.

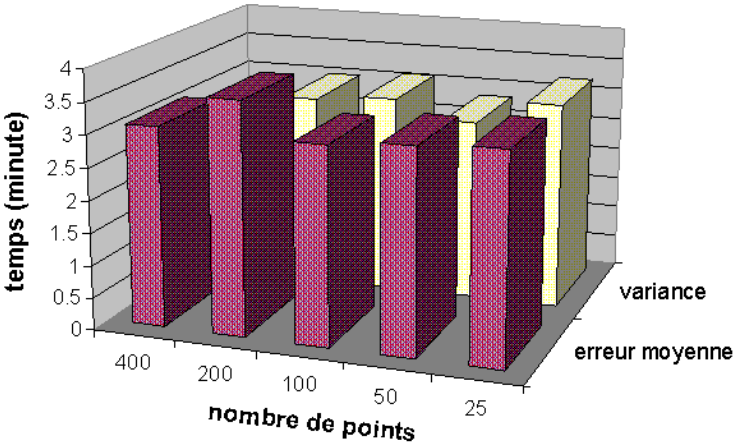

FIG. 6.3: *Erreur moyenne de prévision en fonction du nombre de points utilisés pour décrire les courbes de décharges.*

Il semble donc que l'on puisse ne prendre que 25 points régulièrement espacés sur la totalité de la courbe pour la décrire. Cette remarque permettra de réduire fortement le surcoût en mémoire vive de la méthode incrémentale.

Une amélioration fonctionnelle est également à l'étude. Pour le moment, l'apprentissage de la dernière courbe de décharge ne peut s'effectuer que lorsque celle-ci est presque complète. C'est-à-dire que l'utilisateur doit attendre la décharge complète de sa batterie pour que l'adaptation incrémentale puisse avoir lieu. Ce mode de fonctionnement n'est évidemment pas satisfaisant. Aussi, une étude préliminaire a tenté de mesurer la sensibilité des réseaux NN1 à la longueur de la courbe présentée. La courbe que l'on cherche à apprendre est donc de plus en plus tronquée, puis elle est présentée à un réseau de neurones « moyen ». L'erreur est ensuite calculée sur la totalité de la courbe. Le réseau de neurones, ne voyant que le début de la courbe, est chargé de modifier le comportement moyen afin de s'adapter malgré tout. Les conditions d'apprentissage sont bien évidemment plus difficiles ici, puisque le réseau se trouve confronté à une courbe ne présentant pas la forme génerale « normale ». L'apprentissage est ainsi très vite stoppé, au bout de quelques itérations seulement (5 dans l'expérience suivante). La figure 6.4 présente les résultats d'une telle expérience, pour le cycle 12 de la batterie 4. En fait l'expérience a été réalisée à la fois pour les 800 points de la méthode standard et les 25 points obtenus comme décrit précédemment. Les résultats n'étant pas différents, seuls ceux obtenus avec les courbes à 25 points sont présentés.

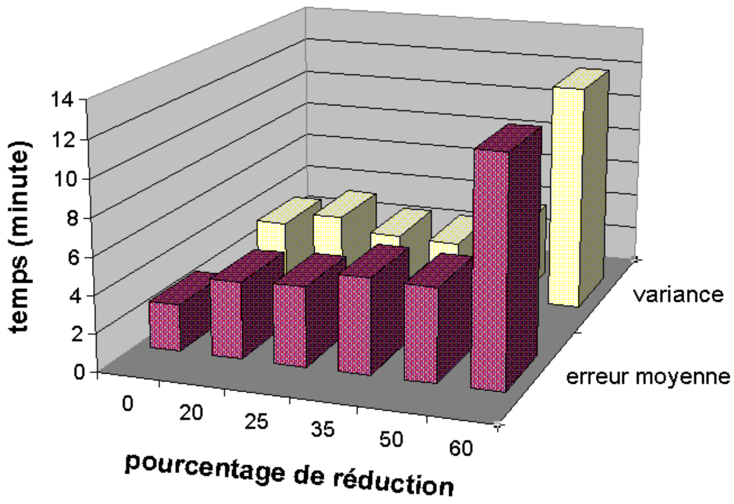

FIG. 6.4: *Erreur moyenne de prévision en fonction du taux de réduction des courbes de décharge.*

Cette étude préliminaire permet d'envisager l'adaptation, même dans le cas de décharges incomplètes. Une règle de mise en fonction de l'adaptation incrémentale peut être établie en fonction de la durée de la dernière décharge ou, plus exactement, de la différence de durée entre deux décharges consécutives. Un problème important demeure : la constitution d'une base de données de décharges de batteries qui puisse couvrir l'ensemble des cas possibles et ceci pour différentes valeurs des paramètres de contexte (température, temps de repos, conditions de charge et de décharge). Les conditions de définition d'une telle base de données sont données dans [Patillon96b].

6.4 Rapprochement avec les méthodes existantes

Le système que nous avons construit est de type adaptatif, dans le sens où il modifie ses réponses en fonction d'un environement contextuel. Comme déjà exposé dans la section 4.3, p. 44, ce même objectif est recherché par les méthodes issues de l'automatique (filtre de Kalman) et les techniques utilisées dans le domaine de la prévision dans les séries temporelles. Cependant, nous avons également doté notre système de possibilités incrémentales afin de le rendre capable de s'adapter en ligne à des situations nouvelles ou des comportements individuels particulièrement marqués. Cette partie tente de faire un rapide tour des méthodes existantes permettant de s'attaquer à ce type de problème, en commençant par les extensions des modèles déjà présentés (tels que le filtre de

Kalman, les séries temporelles, les modèles compétitifs) puis en présentant une architecture hiérarchique neuronale, assez proche de celle que nous proposons.

Dans le modèle général du filtre de Kalman, tel que présenté dans la partie 3.6, il est possible de rendre les processus sous-jacents évolutifs, par l'introduction de la notion du temps dans les éléments constitutifs des équations d'état et d'observation : concrètement, les matrices A, B et C peuvent évoluer et doivent donc être indexées par un indice de temps discret A_k, B_k et C_k. Cette introduction du temps, pourtant totalement intégrée au formalisme du filtre, n'est que très rarement utilisée car il devient alors difficile d'assurer la convergence (raison du succès du filtre de Kalman). Mais le principal problème à l'utilisation du filtre de Kalman pour notre problème de prévision réside dans l'écriture du système (5.1) qui permet la modélisation de la totalité de la courbe de décharge de façon non récursive.

Dans le domaine du traitement de séries temporelles, [Indejehago81] propose d'introduire un vecteur de paramètres évoluant dans le temps. Dans ce formalisme emprunté à celui décrit pour les systèmes dynamiques linéaires (voir p. 22), on ne connaît plus la relation entre X_k et X_{k+1}. On suppose toujours qu'il y a une relation entre l'observation Y_k et le vecteur d'état X_k de la forme suivante :

$$Y_k = X_k^T \lambda_k + v_k \tag{6.1}$$

et que le vecteur de paramètre λ_k évolue linéairement selon :

$$\lambda_{k+1} = \mathrm{C}\lambda_k + \epsilon_k \tag{6.2}$$

où C est une matrice constante et ϵ_k un bruit blanc. Cette modélisation est à rapprocher de notre système décrit par (5.1). En effet, cette même équation pourrait s'écrire dans le cadre de la régression (en ajoutant les bruits), sans modifier le système.

Cependant, le système que nous proposons est plus complexe et donc plus riche (plus adapté à notre problème, en tout cas) car il est foncièrement non-linéaire, contrairement au modèle proposé ici (voir (6.2)). De plus, notre modélisation dynamique des paramètres λ_k dépend directement des variables contextuelles φ, permettant ainsi de rendre le système adaptable aux conditions extérieures. En outre, dans le cas incrémental, NNA tient compte de l'état passé (représenté par λ_{k-1}) pour modifier l'évolution des paramètres.

6.4.1 Modèles compétitifs dynamiques

Les méthodes exposées dans la section 4.3 sont bien de type « hiérarchique » mais le modèle sous-jacent reste toujours constant ; c.-à-d. que l'adaptation au contexte est fixée au préalable. Cette section présente l'évolution de ces méthodes vers des modèles plus dynamiques, plus à même de fonctionner face à des situations imprévues.

Modèles de Markov hiérarchiques

À chaque instant un HMM définit la densité de probabilité des symboles observables comme un mélange des densités définies par chacun des états du processus sous-jacent. Il est donc possible d'introduire une évolution dans le processus sous-jacent. Les *Factorial* HMM (voir p. 30) représentent un premier moyen pour introduire un mouvement au sein du processus caché. Cependant, ce mouvement est complètement stationnaire. [Brugnara92] présente une extension des HMM appelée *Parallel* HMM (PHMM). Il s'agit en fait d'une organisation hiérarchique de type maître-esclave où les probabilités d'émission et de transition du HMM esclave dépendent de l'état du HMM maître.

Modèles compétitifs hiérarchiques

L'architecture en arbre HME (p. 49) peut elle-même évoluer au cours du temps. La figure 6.5 présente cette architecture HMDT ([Jordan96, p. 4]) qui est composée d'un HME (dans la dimension verticale) dont l'évolution dans le temps est décrite par un HMM (dans la dimension horizontale), vue comme un modèle graphique (voir p. 33 pour la représentation d'un HMM).

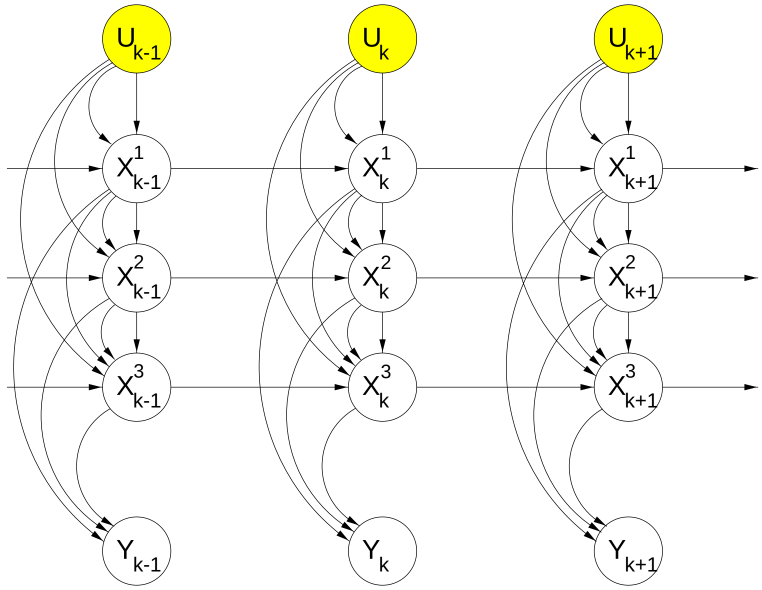

FIG. 6.5: *L'architecture Hidden Markov Decision Tree* (HMDT) *vue comme un modèle graphique.*

6.4.2 L'architecture OWE

Puisqu'il n'existe pas de modèle fiable de batterie, le système doit se baser sur des composantes issues d'une phase d'apprentissage. Nous avons montré comment répondre au problème de la modélisation dans un contexte évoluant dans le temps par une architecture neuronale hiérarchique. Parmi les méthodes proches, on peut citer les travaux de N. Pican ([Pican94b], [Pican96a]) qui présente l'architecture OWE (*Orthogonal Weight Estimator*). Cet algorithme se compose de deux niveaux de réseaux de neurones. Le premier niveau ne comporte qu'un seul réseau qui est chargé d'approximer un phénomène, alors que le second niveau comporte autant de réseaux que de poids du premier réseau. L'architecture OWE est présentée dans la figure 6.6 (voir [Pican95, p. 88]). Un réseau de neurones est

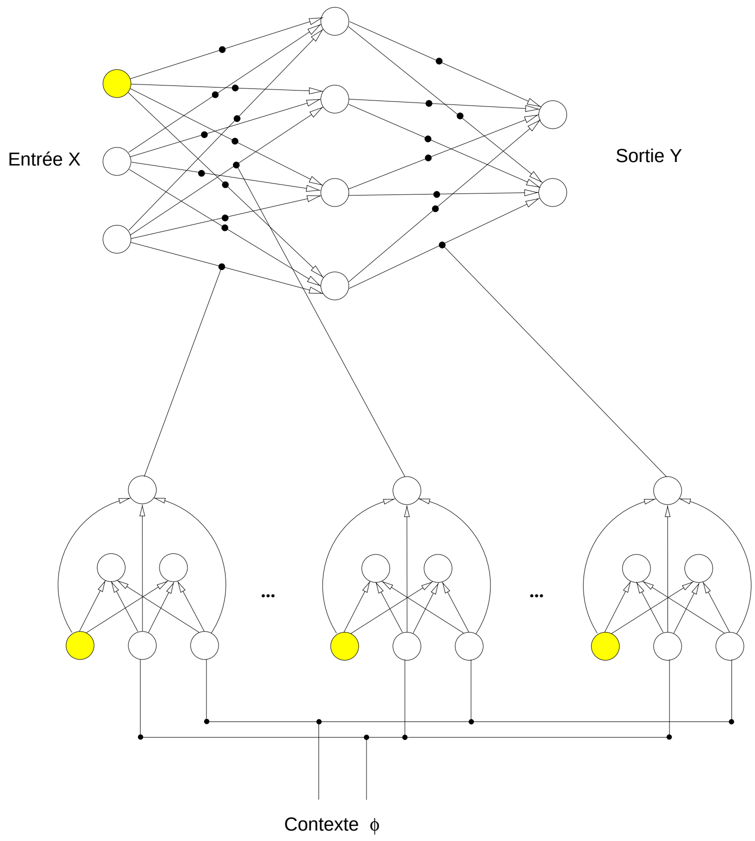

FIG. 6.6: *L'architecture* OWE *: schéma synoptique.*

construit pour chacun des paramètres (poids) du réseau estimateur. La tâche de ces réseaux est d'estimer la valeur des poids en tenant compte de la valeur des variables contextuelles. En outre, ces réseaux sont complètement connectés dans le sens d'utilisation courante (*forward*). Cette architecture a été utilisée pour une tâche de reconnaissance de la parole [Pican96b] et de contrôle [Pican94a].

Le système que nous avons construit est plus simple puisqu'il comporte un modèle global pour tous les paramètres du premier réseau, ce qui permet également de mieux contenir les nombreuses symétries possibles et de contrôler plus facilement la plasticité de la fonction réalisée par le système.

6.5 Perspectives

Les conclusions et prolongements à ce travail peuvent se décliner sous deux aspects : tout d'abord, il est démontré de façon définitive que des modèles issus d'une phase d'apprentissage, réputés complexes, peuvent être utilisés avec profit pour traiter un problème industriel complet et cela même dans les conditions limitées d'un appareil portable. Ensuite, les caractéristiques du modèle proposé : adaptatif au contexte et incrémental vis-à-vis de situations nouvelles (ou de comportements individuels bien marqués), sont intéressantes en général et applicables pour de très nombreuses problématiques.

L'approche développée consiste à proposer une solution connexionniste à un problème d'identification d'un système sensible au contexte extérieur et à apprendre les paramètres de ce modèle neuronal variant au cours du temps. Le système proposé est capable d'adaptation à un certain contexte extérieur. Cette adapation neuronale met bien en valeur les avantages d'une approche à base d'apprentissage : il est possible de construire des systèmes relativement compacts « résumant » un grand nombre de situations différentes. En outre, comme démontré dans ce travail, ce type de système peut très bien être utilisé avec profit même embarqué. Cet aspect mériterait d'être approfondi pour arriver à la conclusion que les systèmes dotés de capacités d'apprentissage ont leur place pour résoudre des problèmes industriels avec très peu de ressources disponibles.

L'approche adaptative proposée ici constitue un outil général applicable dès que le phénomène (ou système) observé évolue au cours du temps en fonction du contexte. Cette problématique peut donc apparaître aussi bien en classification non supervisée qu'en discrimination.

En classification non supervisée, il peut s'agir de l'extraction de modèles d'utilisateurs dont le comportement se modifie au cours du temps.

En discrimination, dans un système de « nez électronique » chargé de détecter certains gaz, il peut être utile de détecter l'évolution de différents capteurs et de regrouper à chaque instant leurs réponses en différentes classes. Les réseaux de neurones se sont déjà montrés capables d'effectuer cette classification. Cependant ce type de capteur souffre d'une dérive temporelle très importante (et, tout comme les batteries, de différences importantes dans le comportement individuel) : leurs réceptacles sensibles s'obtruent petit à petit en fonction des conditions rencontrées (gaz présent, concentration, durée, ...). Le système de classification doit donc être robuste à cette dérive temporelle des mesures ou plutôt intégrer cette dérive dans la modélisation elle-même. Le système proposé pour les batteries peut donc s'appliquer à ce cas.

Parmi les domaines traitant des problèmes similaires, on peut citer l'économétrie, et tous les travaux effectués sur les séries financières pour lesquelles l'influence du contexte est bien évidemment très importante.

Autre domaine d'application : la communauté *Machine Learning* [Widmer97] qui depuis peu cherche à estimer le contexte (ou du moins le changement de contexte directement à partir des données observables). Le problème est ici de modifier les concepts actuellement utilisés dans une situation dès que cette situation évolue. On est donc placé ici dans un cadre de « méta-apprentissage ».

Troisième partie

La détection automatique du Ventricule Gauche

Chapitre 7

Présentation du problème

Dans cette troisième partie un autre problème est traité ; il s'agit de l'extraction d'un contour dans des images médicales et ce premier chapitre propose l'introduction à ce nouveau problème.

Le diagnostic des maladies cardio-vasculaires, la première cause de mortalité dans les pays industrialisés, représente un enjeu très important pour l'avenir. De nombreux examens médicaux existent mettant en pratique l'ensemble des modalités d'imagerie médicale (échographie, résonance magnétique, ...). Cependant, pour le diagnostic et la mesure de paramètres morphologiques et cinétiques du myocarde, la méthode de référence demeure l'exploration radiographique des ventricules cardiaques, après injection d'un produit de contraste.

Ce chapitre présente une introduction au problème traité avec une description de l'intérêt pour l'étude attentive du ventricule gauche, des mesures utiles au diagnostic que l'on peut extraire, ainsi que de l'examen pratiqué : la ventriculographie radiologique.

Le diagnostic passe alors par une nécessaire étape de délinéation des contours dans les images complexes et bruitées que sont les radiographies numériques.

L'objectif général de ce travail est donc la détection *automatique* des contours du ventricule gauche (VG) dans des images rayons X.

7.1 Le contexte biomédical

7.1.1 Les enjeux médicaux

Les maladies cardio-vasculaires représentent la première cause de mortalité dans les pays industrialisés. En 1995, elles ont causé 961 000 décès aux États-Unis (538 000 décès dus aux différentes formes de cancer et 93 000 à des accidents) [(21)] et ont été responsables de 32,3% des décès en France (cancer : 27,6%) [(22)]. Ce phénomène n'est pas récent, d'où l'importante recherche faite dans ce domaine, notamment au niveau du diagnostic qui se doit d'être précoce et précis. Le diagnostic s'effectue, en partie, à partir de mesures morphologiques et cinétiques effectuées sur le muscle cardiaque dont le ventricule gauche est le composant le plus important.

7.1.2 Le ventricule gauche

Le ventricule gauche est l'une des quatre cavités qui composent le muscle cardiaque. Ce muscle est chargé de faire circuler le sang dans tout le corps humain. En fait, le système de pompage complet est composé de trois compartiments : les deux chambres (oreillettes et ventricules) et les artères. La figure 7.1 présente les différentes parties du cœur ainsi que la manière dont il est relié au système circulatoire.

Le ventricule gauche (VG) est très intéressant puisque c'est lui qui a la fonction de pompe la plus importante. C'est en effet à partir de cette cavité que le sang oxygéné est envoyé à tous les membres (supérieurs, tête, tronc et inférieurs) via l'aorte. Ce sang oxygéné est « consommé » par les différents organes avant de revenir vers le ventricule droit (VD) qui est, lui, chargé de le renvoyer vers les poumons pour oxygénation. Le cycle se termine par la ré-entrée du sang dans le ventricule gauche via les veines pulmonaires et l'oreillette gauche.

Le cycle cardiaque se compose de deux phases distinctes et alternées : la systole et la diastole. La phase systolique est la contraction du muscle cardiaque. La systole simultanée des deux oreillettes précède celle, également simultanée, des deux ventricules. La systole ventriculaire débute par la fermeture des valvules auriculo-ventriculaires et se termine par celle des valvules aortiques (pour le VG) et pulmonaires (VD). La phase diastolique, plus longue [(23)], correspond au relâchement du cœur au moment de l'afflux sanguin.

7.1.3 Les points d'appui pour le diagnostic

Le diagnostic des maladies cardiovasculaires s'appuie pour une grande part sur le résultat de l'analyse du comportement du myocarde tant par des mesures morphologiques que cinétiques. Ce diagnostic comporte également une étude de

(21). source : « American Heart Association. 1998 Heart and Stroke Statistical Update » `http://www.americanhearth.org`

(22). source : groupe SC8 de l'INSERM, service d'information sur les causes médicales de décès, `http://sc8.vesinet.inserm.fr:1080`

(23). pour un cœur battant à 75 pulsations par minute, la phase systolique dure environ 360 ms, la phase diastolique 500 ms [West85, chap. 13].

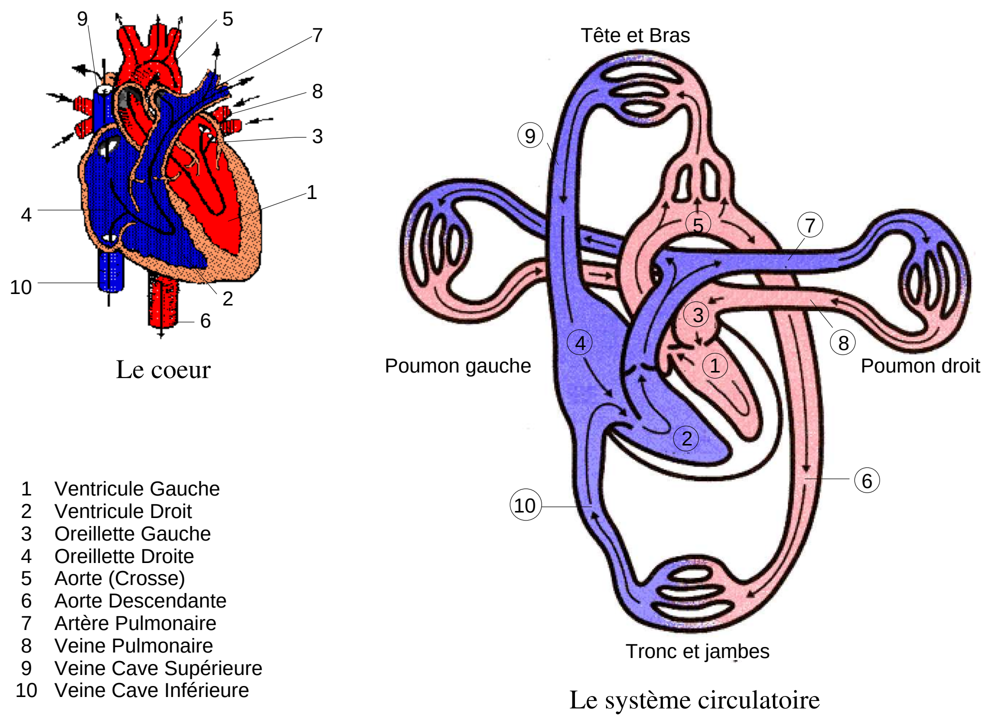

FIG. 7.1: *Description de l'anatomie du muscle cardiaque et de son rôle de pompe du système circulatoire.*

l'opportunité de recourir à la chirurgie.
Un indice important de la santé du muscle cardiaque et de l'appareil circulatoire est la fraction d'éjection ventriculaire (FE), c'est-à-dire le rapport différentiel entre les volumes maximum (fin de diastole, télédiastole, V_{FD}) et minimum (fin de systole, télésystole, V_{FS}) :

$$FE = \frac{V_{FD} - V_{FS}}{V_{FD}} \ . \tag{7.1}$$

La valeur normale de ce rapport, qui mesure la part du volume sanguin éjectée pendant la systole (et donc la performance ventriculaire), est comprise entre 0,55 et 0,80 avec une moyenne de 0,67 ($\pm 0,088$) [Fifer86, Kennedy70]. C'est-à-dire que les 2/3 du volume sanguin présent à la fin de la diastole sont éjectés à chaque battement [West85, chap. 17]. On peut considérer la fraction d'éjection comme étant normale lorsqu'elle est supérieure à 0,55 et on considère qu'il y a dysfonctionnement grave pour des valeurs de FE inférieures à 0,40. Le calcul de ce rapport nécessite la délinéation du ventricule dans les images du cycle. La détermination des instants de télésystole (*fin de systole*) et de télédiastole (*fin de diastole*) est effectuée par observation directe de l'image (ventricule le moins volumineux et le plus, respectivement). Il existe un moyen très simple de repérer la télédiastole par l'enregistrement simultané et synchronisé de l'électrocardiogramme (ECG). Il existe également des études pour des systèmes automatiques de détection de ces instants, directement à partir des images angiographiques et ce, à l'aide d'une méthode utilisant des techniques d'estimation de mouvements (flux optique) [Legrand96].

Un autre point d'appui pour le diagnostic consiste en la détection et l'analyse du mouvement du muscle cardiaque, le myocarde. En effet, l'infarctus du myocarde se caractérise par la nécrose (24) d'une partie du muscle qui perd de sa souplesse et donc de son efficacité. L'ampleur de cette rigidité est donc un élément important pour évaluer l'importance et la gravité de l'attaque. L'estimation de cette rigidité doit donc être menée sur une séquence d'images comportant plusieurs cycles cardiaques complets. Suite au diagnostic d'un infarctus, la mesure de la fraction d'éjection a d'autant plus d'importance qu'elle est directement liée au taux de mortalité à court terme (25).

Calculer la fraction d'éjection ventriculaire nécessite d'estimer avec grande précision le volume du ventricule gauche. Cette estimation devient possible dès lors que l'on est capable de détecter les contours de cet organe dans l'image présentée (section 7.3, p. 92-95 présente les méthodes les plus utilisées). Il est également nécessaire de pouvoir suivre les contours au cours du cycle cardiaque pour évaluer le caractère homogène, harmonieux de la contraction ventriculaire. Il devient alors possible de distinguer plusieurs degrés d'anomalie :

- hyperkinésie : mouvement trop important d'une région du cœur ;
- hypokinésie : diminution localisée de la contraction d'une région ;
- akinésie : immobilité totale de la région concernée ;
- dyskinésie : mouvements non coordonnés ou expansion paradoxale en systole.

La première étape consiste donc à obtenir des images du ventricule se prêtant bien au diagnostic des maladies cardio-vasculaires.

Des examens médicaux basés sur différentes modalités d'imagerie médicale (principalement rayons X, échographie et IRM) ont pour objet cette caractérisation du ventricule (pour une introduction aux différentes techniques de radiologie, se référer à [Aubert95] et [Acharya98]). Parmi ceux-ci, l'angiographie ventriculaire (égalemment dénommée ventriculographie radiologique) revêt une importance particulière comme souligné (en 1993) par J. P. Ollivier *et al.* :

« Historiquement entre 1960 et le début des années 70, l'angiographie ventriculaire sélective a été la méthode quasi-exclusive des progrès décisifs en cardiologie clinique ; ultérieurement, alors qu'émergeaient les techniques non-sanglantes, échocardiographie et scintigraphie cavitaire, elle est restée la méthode de référence, sorte d'étalon-or pour l'ensemble des paramètres morphologiques et cinétiques du ventricule [...] elle reste malgré les brillantes percées des techniques dites nouvelles l'outil quotidien et pour l'instant non remplacé du cardiologue clinicien » [Ollivier93, p. 443].

Cet examen est décrit dans la section suivante.

(24). la mort de tissu cellulaire, ici à la suite d'un défaut d'approvisionnement sanguin.

(25). avec une fraction d'éjection de 0,40, le taux de mortalité à un an est de 10% alors que pour une FE de 0,20, il s'élève à 35% [INSERM95].

7.1.4 La ventriculographie radiologique

La ventriculographie radiologique est un examen particulier de ce qu'il est convenu d'appeler la cathétérisation cardiaque (elle-même proche de l'angiographie : « radiographie des vaisseaux après injection d'un liquide opaque aux rayons X » [GD95]) dont l'objectif est le diagnostic de pathologies liées au cœur et à son entourage proche, comme les artères, par exemple. Elle est prescrite dans de nombreux cas, comme par exemple : les angines difficiles, les douleurs atypiques de la poitrine et la myocardiopathie. Plus d'un million de diagnostics de ce type ont été pratiqués aux États-Unis en 1995 et ce nombre ne cesse d'augmenter [(26)]. En France, le nombre de coronographies est également en augmentation pour atteindre près de 200 000 en 1997, comme présenté dans la figure 7.2 [(27)].

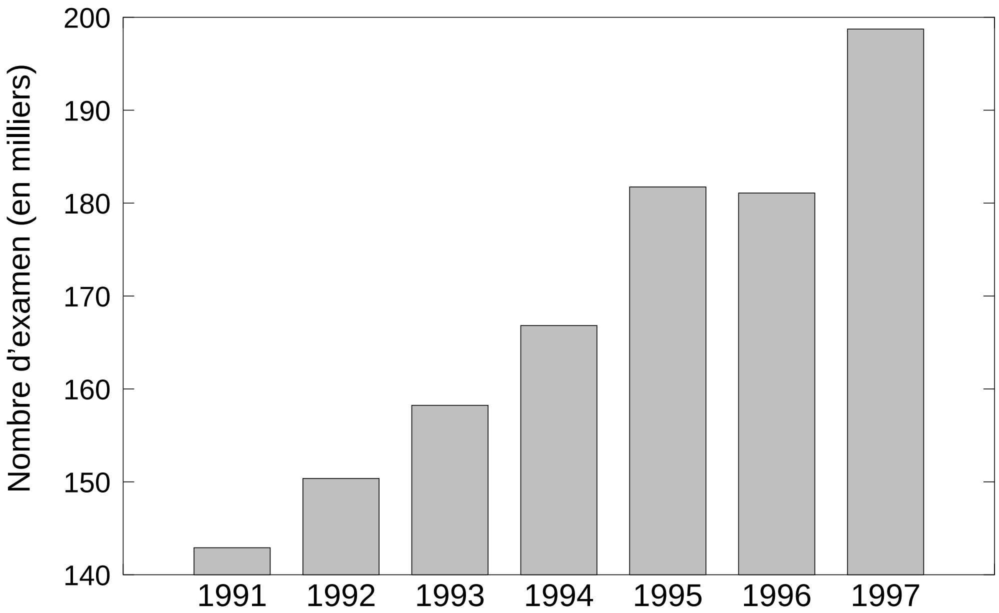

FIG. 7.2: *Évolution du nombre de coronographies pratiquées en France (secteur public+libéral).*

Le principe de l'examen est d'amener le liquide opaque aux rayons X, jusqu'à l'intérieur de la cavité à explorer. Ce liquide est injecté par l'intermédiaire d'un cathéter. Ce tube en plastique souple, visible par rayons X, possède généralement une extrémité en forme de queue de cochon et est percé de nombreux petits trous afin de diffuser de manière la plus homogène possible le produit.

L'examen se pratique sous simple anesthésie locale, à l'endroit de pénétration du cathéter (en général, au niveau de l'artère fémorale qui permet un accès en ligne droite au cœur). Au cours de l'examen, l'ECG (électrocardiogramme)

(26). source : « American Heart Association. 1998 Heart and Stroke Statistical Update » `http://www.americanhearth.org`

(27). Ces informations sont collectées par le Professeur J.-M. Lablanche du CHRU de Lille. Je le remercie ici de me les avoir communiquées.

est enregistré. Une fois le cathéter (un tube en plastique souple) introduit dans une artère, il est poussé par le médecin, sous le contrôle d'images fluoroscopiques (rayons X continus de faible puissance), vers l'endroit souhaité (dans le ventricule dans notre cas). Le produit de contraste est ensuite injecté (à une vitesse comprise entre 12 et 15 ml/s) et les clichés radiographiques sont pris au cours du cycle cardiaque.
La formation d'images par rayons X est réalisée par un système d'acquisition constitué de quatre modules principaux : un tube émetteur de rayons X, un récepteur des photons ayant traversé le corps du patient, un système de formation de l'image à partir de l'information stockée et un support de visualisation. En radiographie standard, les trois derniers modules n'en forment qu'un, puisque les photons transmis viennent directement impressionner le film final. Ce sont les photons primaires, transmis sans interaction qui apportent l'essentiel de l'information servant à former l'image radiographique, c'est-à-dire une mesure de l'atténuation des rayons X par les tissus traversés. L'image obtenue est en fait la projection bidimensionnelle de la distribution volumique des propriétés d'atténuation de tous les tissus irradiés. Cette projection est obtenue par intégration des coefficients linéaires d'atténuation de tous les tissus traversés.

De plus en plus souvent, les radiographies sont de type numérique, sans passer par le film radiographique traditionnel. Dans le cas de ces systèmes numériques, la mesure des absorptions des rayons X par les tissus et l'affichage de l'image obtenue se font sur des systèmes distincts. La fréquence d'acquisition en est un peu réduite sur les appareils actuels (environ 25 images/s contre 50 pour les radiographies standard). Le film obtenu est appelé « ciné-angiogramme » (autres noms de l'examen : ciné-ventriculographie, ciné-angioventriculographie). D'une façon générale, dans ce type de radiographie numérique on peut affirmer que le rapport signal à bruit (SNR) des images obtenues sera proportionnel au produit de la concentration en iode du produit de contraste par la racine de la dose de radiation reçue par le patient [Brody84] :

$$SNR \propto C_{\text{iode}} \times \sqrt{\text{dose}}. \tag{7.2}$$

La projection radiographique classique pour ce type d'examen est la projection OAD 30° (Oblique Antérieure Droite, RAO en anglais), telle que schématisée dans la figure 7.3. Elle est la plus à même de montrer le grand axe du ventricule, c'est-à-dire la distance entre l'apex et la valvule aortique. La section 7.3 présente l'historique de cet examen.

Cet examen est le plus souvent réalisé par un appareil de rayons X monté sur un arceau mobile, ce qui permet un grande liberté de positions (pour la source et le tube intensificateur d'image) tout en conservant un encombrement réduit et une accessibilité importante. La figure 7.4 présente une photographie de la salle de cathétérisme de l'hôpital du Kremlin-Bicêtre, équipé par un Philips Integris V 3000 (28). Un descriptif de cette image est présenté dans la figure 7.5. Cet examen est le plus précis disponible à l'heure actuelle pour mesurer le fonctionne-

(28). Tous mes remerciements au Dr Hébert pour m'avoir permis de prendre cette photographie.

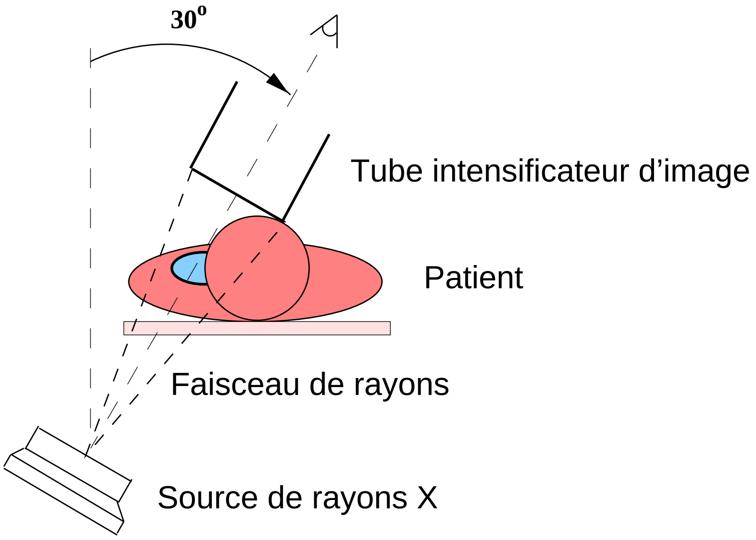

FIG. 7.3: *Description de la projection oblique antérieure droite 30°.*

ment cardiaque, diagnostiquer les maladies des valves, estimer le mouvement des parois cardiaques et déterminer le besoin de recourir à la chirurgie [Denolin93].

La figure 7.6 présente un exemple d'une telle image, ainsi que la description schématique de son contenu.

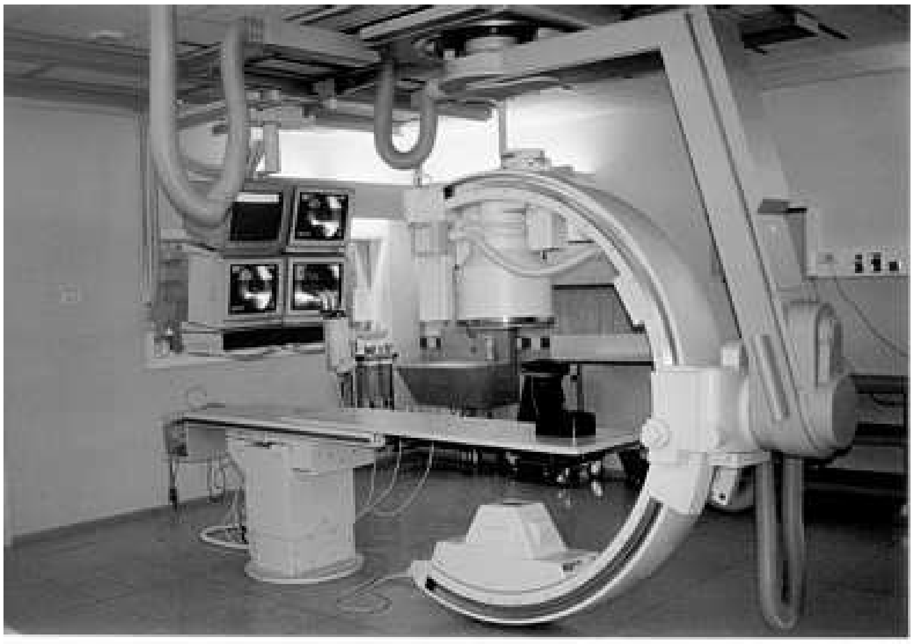

FIG. 7.4: *La salle de cathétérisme de l'hôpital du Kremlin-Bicêtre.*

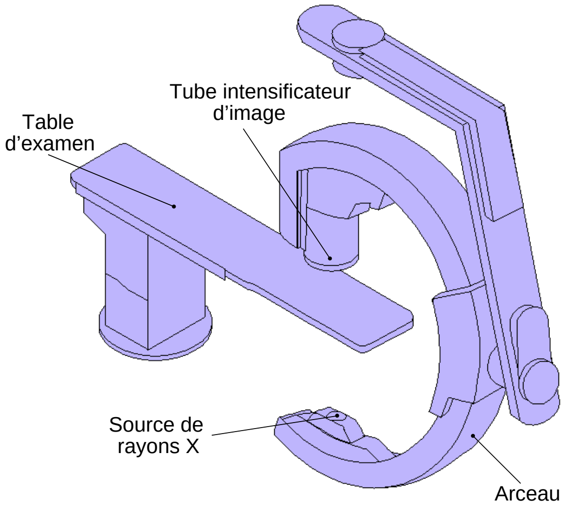

FIG. 7.5: *Schéma d'un Integris, description de l'image 7.4.*

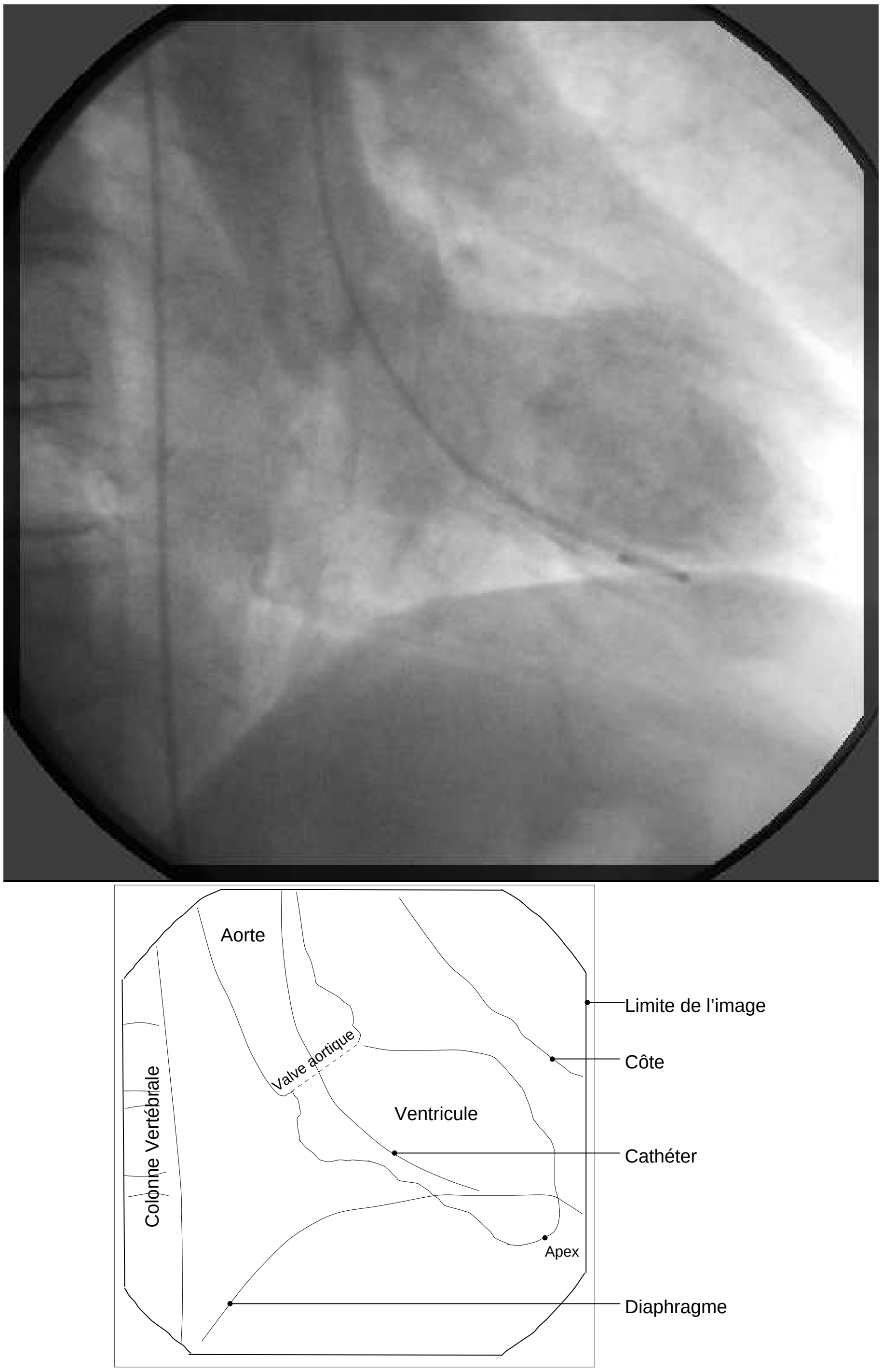

FIG. 7.6: *Un cliché de ventriculographie radiologique et sa decription approximative.*

7.2 La détection des contours dans les images

Bien évidemment, le traitement automatique de telles images peut permettre d'aider le radiologue dans son diagnostic en fournissant des mesures morphologiques sur le ventricule et en augmentant le rapport signal utile à bruit. Un autre avantage potentiel serait la réduction de la dose de radiation, ou celle de la concentration du produit de contraste qui peut être nocif (voir équation (7.2)).

7.2.1 Les besoins spécifiques en traitement d'images

Pour tous les éléments de diagnostic liés à la ventriculographie radiologique, les contours du ventricule doivent être détectés dans les images.
Les contours du ventricule gauche visibles lors d'un tel examen sont présentés dans la figure 7.7 (Docteur J. L. Hébert, Kremlin-Bicêtre, communication privée). Les traits en filet (la valve mitrale et les parties antérieure et posté-

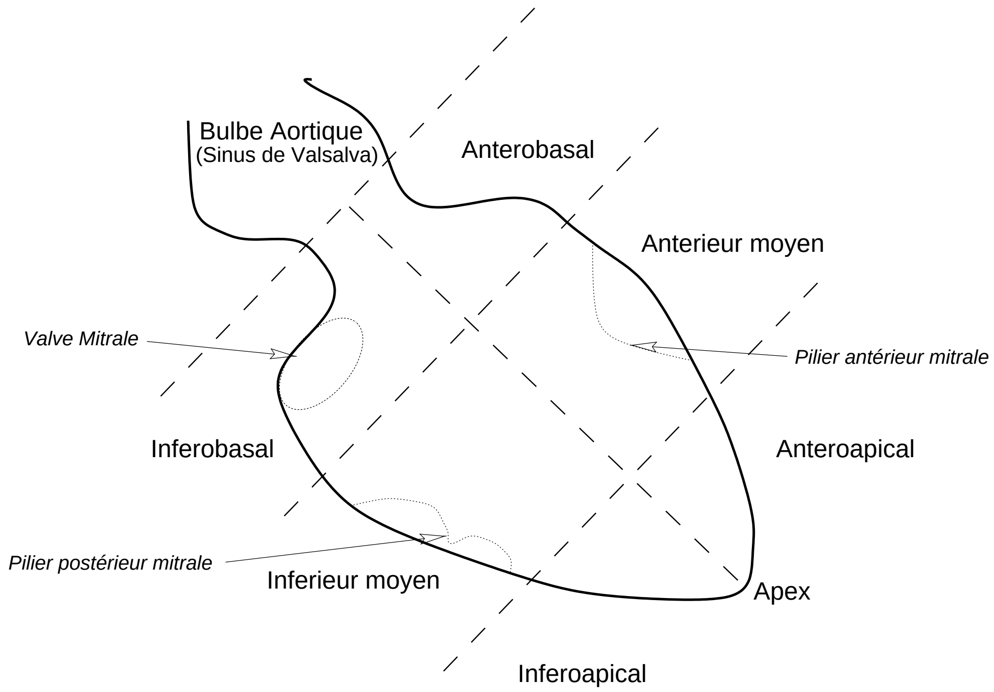

FIG. 7.7: *Anatomie du ventricule gauche dans la projection OAD 30°.*

rieure du pilier de la valve mitrale) représentent des structures qui peuvent être très problématiques pour la détection de contour. En effet elles sont souvent d'un contraste plus important que le véritable contour du ventricule. C'est en particulier le cas pour la partie postérieure du pilier pour lequel il a fallu une convention internationale pour bien définir que le contour à détecter passait au large de cette structure. Cette remarque étant surtout vraie pour l'image obtenue en télésystole.

Malgré tout, ces images sont difficiles à traiter. En effet, les images par rayons X ne sont que des projections et donc tous les organes traversés sont (plus ou moins) représentés en superposition dans l'image. D'autre part, ce qui permet de rendre visible le ventricule sur de telles images, c'est le produit de contraste, or celui-ci peut avoir (et a) des variations importantes de densité. Dans certaines images, le produit de contraste est très peu dense au niveau de l'apex ce qui rend sa détection d'autant plus difficile.

La difficulté du problème réside donc dans l'aspect « automatique » puisque les images ont en général un très mauvais contraste et qu'elles présentent une variabilité très importante.

7.2.2 Les systèmes existants

Les méthodes couramment utilisées pour la détermination de ces contours requièrent l'intervention du médecin.

La première méthode est la plus simple : le médecin dessine le contour à l'aide d'une souris ou d'un crayon optique. Si les images sont sur le support film classique, le médecin dessine sur un calque le contour puis un assistant est chargé de numériser les contours obtenus. Dans tous les cas, cette méthode demande beaucoup de temps au médecin. De plus, l'opération devra être répétée pour chaque cliché, ou tout du moins, pour les images télédiastolique et télésystolique.

La seconde méthode est semi-automatique : après une phase de sélection manuelle de points clé, un contour est automatiquement détecté et modifié manuellement le cas échéant. La première étape consiste à la sélection de points clés dans l'image. Ces points sont (généralement) les deux extrémités de la valve aortique et l'apex. L'apex est défini comme étant le point extrême du ventricule, celui qui est le plus éloigné de la valve aortique. Une fois ces points sélectionnés sur chacune des images, le milieu du ventricule est estimé. L'image subit alors une transformation radiale centrée sur le milieu du ventricule. Les intensités des points le long des rayons sont enregistrées et tout changement brusque est détecté. En effet, un changement brusque d'intensité doit être le signe du contour du ventricule. Pour chaque rayon, le changement le plus brutal est conservé et un premier contour est alors construit en reliant les points de chaque rayon. Ce contour est cependant très bruité et des contraintes de régularité le long du contour sont introduites afin de lisser celui-ci. Au bout de quelques itérations, un compromis est trouvé et le contour dessiné. Malgré tout, cette étape est également semi-automatique et elle requiert la sélection d'un paramètre par le médecin [Brody84, p. 72-73]. En outre, le contour obtenu n'est souvent pas satisfaisant pour le radiologue qui utilise alors la possibilité de modifier manuellement tout ou partie du contour en le redessinant (29).

Une étude comparative des méthodes manuelles et semi-automatiques sur 30 fantômes (c'est-à-dire des cœurs pris sur des cadavres pour lesquels les volumes exacts ont pu être mesurés) a montré que les méthodes semi-automatiques sont

(29). « la plupart des radiologues trouve que [les outils de détection semi-automatique] sont très peu précis » F. Sheehan, communication privée.

plus précises et plus robustes [Beier93, Beier96]. Lors d'une détection manuelle du contour, les volumes calculés sont surestimés de 32, 8 ml (écart-type : 21, 2 ml), alors qu'une détection semi-automatique ne surestime les volumes que de 6, 2 ml (pour un écart-type encore important : 20, 4 ml). Cette étude, effectuée dans le cas très favorable d'images parfaitement contrôlées de ventricules seuls (sans aucun autre organe), démontre bien la sensibilité des résultats obtenus.

Les sections suivantes décrivent les méthodes utilisées pour obtenir des mesures utiles au diagnostic à partir du contour extrait, avec en premier lieu l'estimation du volume, suivi de l'étude de la dynamique du ventricule.

7.3 Estimation du volume du ventricule gauche à partir du contour extrait

Le volume du ventricule gauche (VG) est une donnée de diagnostic importante pour le médecin. À la fois pour estimer les paramètres morphologiques du cœur, mais aussi pour mesurer la fraction d'éjection, indice important de la qualité de la pompe cardiaque. Le problème ici est d'estimer ce volume à partir de projections radiographiques dans lesquelles le contour du ventricule a été délimité.

7.3.1 Historique

Les premières méthodes de mesure du volume cardiaque à l'aide de rayons X datent du tout début des examens radiographiques. La première expérience de radiographie cardiaque remonte à 1916. En 1929, le premier cathétérisme cardiaque (oreillette droite) fut effectué par W. Forssman sur lui-même. Les années 30 furent prolifiques en travaux sur le sujet avec les premières modélisations du cœur sous la forme d'une ellipsoïde en trois dimensions. Mais c'est surtout à partir du développement de l'angio-cardiographie qu'il devint possible d'étudier séparément chacune des cavités cardiaques. Les premières analyses du volume du VG datent de la fin des années 1950. [Arvidsson61] et [Reiber96] contiennent un historique détaillé de cette période pionnière ainsi que de nombreuses références.

Les premières études du début des années 60 [Dodge60, Arvidsson61] utilisaient deux projections perpendiculaires : les vues anteropostérieure et latérale. À partir de ces vues et des contours délimités, les longueurs des axes de l'ellipse modèle sont estimées puis le volume est calculé. Cette méthode, dite de surface/longueur est introduite dans [Dodge60]. En 1965 cette méthode est simplifiée et n'utilise plus qu'une seule des deux vues précédentes. Cela permet de réduire les coûts matériels (un seul tube intensificateur fixe est suffisant) et de diminuer la dose de rayonnement reçue par le patient, avantages permettant d'accepter une perte de précision dans l'estimation du volume.

En 1970, la projection oblique anteropostérieure droite (OAD, RAO en anglais) à 30° s'impose (voir la figure 7.3, page 87) ; elle est plus à même de fournir des informations sur les parois des muscles cardiaques ainsi que sur la présence de maladies liées à la valve mitrale. À l'heure actuelle, la majorité des examens de ce type ne comporte que cette prise de vue en projection OAD 30°.

Une méthode concurrente, qui consiste à « découper » le ventricule en fines tranches de section circulaire puis à utiliser la règle d'intégration de Simpson (voir par exemple [Rousselet83, p. 65]), est également introduite en 1958. Cette méthode est réputée comme plus précise mais nécessite davantage de mesures et de calculs et elle n'est donc pas très utilisée.

Ces modèles simples ne font qu'approximer la forme complexe du VG. Afin de corriger les estimations et pour tenir compte de la présence des muscles, des équations linéaires de correction ont été établies (pour la vue RAO) et validées à partir d'études *post-mortem* dans les années 1970. Depuis, ces équations de correction ont évolué afin de tenir compte de l'avancée technologique des appareils de rayons X [Reiber96].

7.3.2 La méthode « surface/longueur »

La méthode surface/longueur (« Area Length » en anglais) a été introduite dans [Dodge60], elle est décrite en détail dans [Wong86]. Elle considère le ventricule comme un ellipsoïde 3D. Les trois axes de l'ellipsoïde sont estimés à partir des deux projections perpendiculaires : les vues anteropostérieure et latérale. Le volume du ventricule est ensuite estimé par :

$$V = \frac{\pi L}{6} \frac{4A_{\text{ap}}}{\pi L_{\text{ap}}} \frac{4A_{\text{lat}}}{\pi L_{\text{lat}}} \tag{7.3}$$

où A_{ap} et A_{lat} sont les surfaces projetées du ventricule gauche sur les vues anteropostérieure et latérale, respectivement. L_{ap} et L_{lat} sont les plus grands axes visibles dans les projections et L est l'axe le plus grand entre L_{ap} et L_{lat}. L'erreur moyenne d'estimation du volume est alors de 8% [Collins86, p. 231]. Cette formulation est également valide pour les projections obliques, au prix d'une augmentation de l'erreur de 1 à 5 %. La même technique a ensuite été étendue pour le cas d'une seule projection (mono-plan). L'hypothèse qui est alors faite est que l'ellipsoïde est symétrique par rotation autour du grand axe ($L_{\text{lat}} = L_{\text{ap}}$). Cette hypothèse est légitime pour des cœurs non pathologiques en télédiastole. En phase télésystolique l'erreur introduite est raisonable, mais l'hypothèse montre toutes ses limites dans les cas pathologiques. La méthode mono-plan reste cependant la plus utilisée et l'on se place dans ce dernier cas. Le volume est estimé à partir de la seule projection oblique anteropostérieure à 30°.

La méthode surface/longueur mono-plan nécessite le calcul de l'aire A (en pixel^2) ainsi que celui de la longueur du long axe L, défini comme le segment reliant la partie postérieure de la valve aortique à l'apex (en pixel). Les mesures nécessaires sont présentées dans la figure 7.8.

À partir de ces deux mesures et sous l'hypothèse que le VG peut être modélisé par une ellipsoïde 3D, le volume est estimé par :

$$V_m = \frac{8}{3\pi} \frac{A^2}{L} \quad \text{pixels}^3. \tag{7.4}$$

Pour chaque cliché, un facteur de calibration FC lié à l'appareillage de rayon X et à la prise de vue est estimé. Ce facteur mesure la taille d'un pixel et

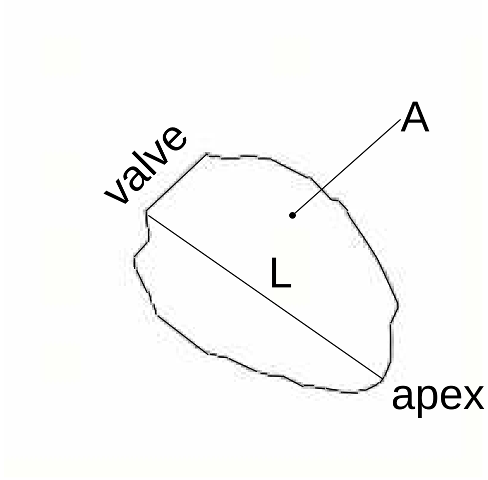

FIG. 7.8: *Les mesures nécessaires pour l'estimation du volume du ventricule par la méthode surface/longueur.*

est généralement exprimé en mm/pixel (30). Dans ce cas, le calcul du volume devient :

$$V_m = \frac{8}{3\pi} \frac{A^2}{L} \ FC^3 \ \text{ ml.} \tag{7.5}$$

Dans le cas des images présentées dans ce document, l'équation de correction est la suivante :

$$V = 0,81 \times V_m + 1,90. \tag{7.6}$$

Il existe 7 méthodes pour estimer le volume à partir de deux vues, en utilisant en plus la vue en oblique antérieure gauche 60° (OAG 60). Le choix pour utiliser une méthode dans un centre particulier dépend plus étroitement d'aspect historiques que techniques. Néanmoins, [Beier96] montre que ce choix peut conduire à de grandes disparités dans l'estimation du volume. Ainsi la méthode de Simpson surestime constamment les volumes (environ $8,5$ ml en moyenne).

À titre d'exemples, les valeurs typiques de volumes du ventricule gauche sont de 150 ml en fin de phase diastolique (télédiastole) et 75 ml en télésystole. Cependant, pour pouvoir comparer ces valeurs entre patients, il faut tenir compte de la morphologie du patient. Les radiologues utilisent donc préférentiellement des valeurs de volume indexées à (divisées par) la surface corporelle (BSA, *Body Surface Area*, en anglais). Il existe plusieurs méthodes pour estimer cette surface. Ces méthodes se fondent sur la taille ainsi que le poids du patient. Il semble que la plus utilisée d'entres elles soit celle due à Dubois & Dubois, qui date de 1916 et qui a été établie à partir de 9 corps et publiée dans [Dubois16]. La surface corporelle S (en m^2) est estimée par

$$S = 0,007184 * P^{0,425} * T^{0,725}, \tag{7.7}$$

(30). Pour ce type d'image, ce facteur vaut environ 0,30 mm/pixel.

où P est le poids en kg et T la taille en cm. Les volumes normalisés V/S moyens pour un adulte non malade sont de 70 ml/m^2 (écart type : 20) en télédiastole et de 24 ml/m^2 (écart type : 10) en télésystole [Kennedy70].

7.4 Estimation de la dynamique des ventricules cardiaques

La dynamique des ventricules cardiaques est un élement très important pour établir le diagnostic de très nombreuses pathologies cardiaques. Les techniques actuelles se basent sur la comparaison des images de ventricules en fin de phase diastolique et systolique, images sur lesquelles les contours ont été délimités. L'analyse se fonde en fait sur la comparaison de ces contours. L'objectif de cette évaluation de la cinétique segmentaire est la détection de zones du myocarde présentants des anomalies de déplacements (hyperkinésie, hypokinésie, akinésie et dyskinésie).
Les paragraphes suivants contiennent une présentation des deux algorithmes les plus utilisés pour les images angiographiques (voir également [Wong86]).

7.4.1 La méthode radiale

Cette méthode suppose que tous les segments du ventricule se contractent en systole vers un centre hypothétique. Ce centre peut être choisi comme

- le centre de gravité du ventricule en télé diastole ou systole,
- le milieu du grand axe en télé diastole ou systole.

Une transformation rigide des images peut être utilisée pour mettre en correspondance les points aux différents instants. La figure 7.9 présente un exemple de la transformation radiale pour un ventricule gauche en OAD, le centre étant le centre de gravité du ventricule en télé diastole. Cette méthode est également connue sous le nom de **méthode de Stanford** ou **méthode de Ingels** [Ingles75] [Karsch80].
La cinétique est mesurée par le taux de raccourcissement des segments radiaires, entre les contours diastolique et systolique. La détection des problèmes éventuels de cinétique s'effectue par comparaison avec un graphique moyen établi pour un ensemble de patient « normaux ».

7.4.2 La méthode de la ligne centrale

Cette méthode plus récente, également connue sous le nom de **méthode de Sheehan**, a été proposée par F. Sheehan (voir [Sheehan86], par exemple). Elle permet de s'affranchir de la variabilité introduite par le sélection d'un centre de contraction. L'idée est de considérer que la cinétique des parois du myocarde est perpendiculaire à la paroi. C'est-à-dire que le mouvement est estimé perpendiculairement au contour télé-diastolique, comme la longueur des segments(31)

(31). normalisée par la longueur du contour diastolique.

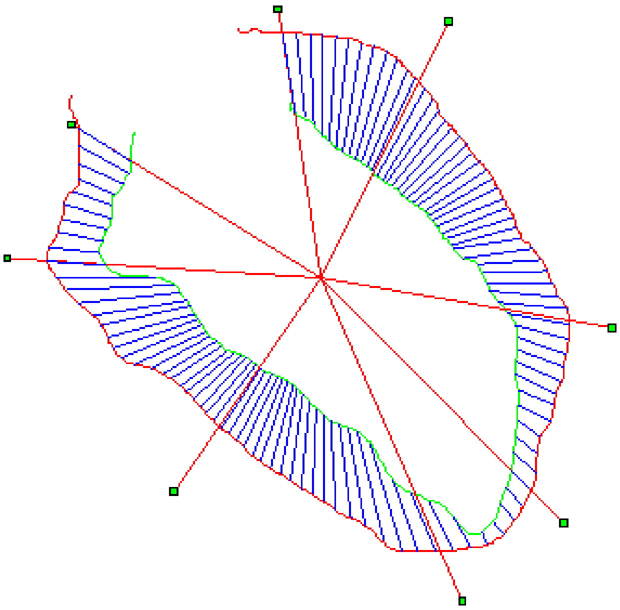

FIG. 7.9: *Le modèle de transformation radiale utilisée pour étudier la dynamique des segments ventriculaires.*

reliant des points régulièrement espacés sur ce contour et l'intersection des normales en ces points avec le contour systolique. En fait l'algorithme est itératif, puisque le résultat peut être modifié en appliquant à nouveau l'algorithme, la ligne centrale (la ligne joignant les milieux des segments) jouant alors le rôle du contour diastolique.

La figure 7.10 présente un exemple de la transformation par ligne centrale, pour un ventricule gauche en OAG. Cependant cette méthode peut sans aucun problème être utilisée pour la vue en OAD et pour le ventricule droit.

Le diagnostic de la cinétique des parties du myocarde s'effectue alors par comparaison avec un graphique moyen établi pour un ensemble de patient « normaux » [(32)], de la même manière que pour la méthode radiaire, comme présenté dans la figure 7.11.

(32). Ce normogramme a été établi avec l'aide du Dr. Hébert [Hébert00].

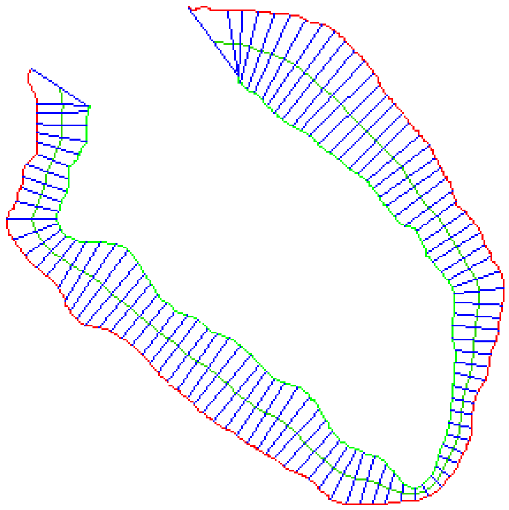

FIG. 7.10: *Le modèle de transformation par ligne centrale utilisée pour étudier la dynamique des segments ventriculaires.*

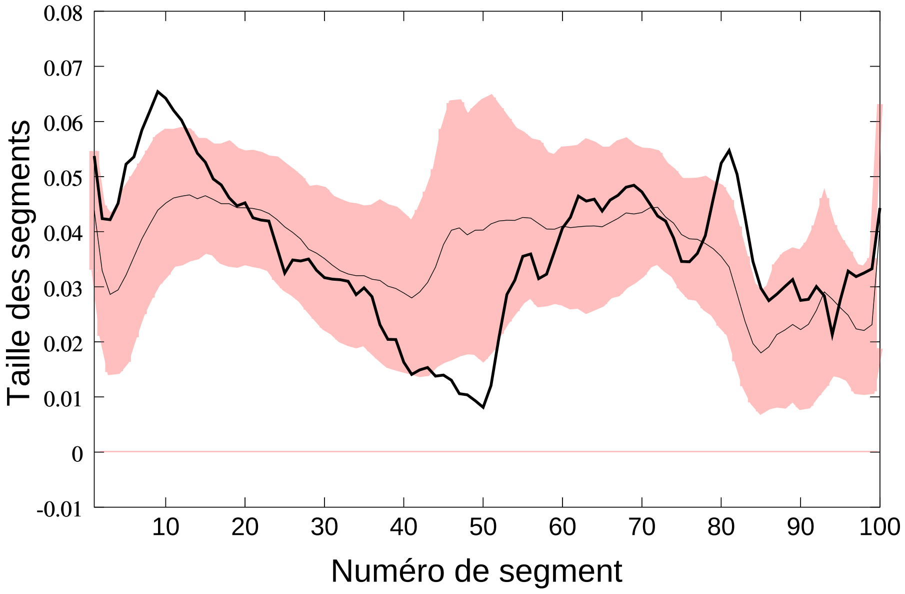

FIG. 7.11: *Étude de la cinétique du ventricule de la fig. 7.10.*

Chapitre 8

Revue critique de l'état de l'art en détection d'objet

Le chapitre précédent a montré que la mesure de quantités utiles au diagnostic à partir de ventriculographie radiologique passe par une nécessaire étape d'extraction de contour.

Ce chapitre présente un rapide panorama des méthodes généralement utilisées en traitement d'images pour extraire un objet dans des images numériques. Les détails techniques sur ces méthodes ont été placés en annexes (annexe B, à partir de la page 183) afin de ne pas complexifier inutilement le discours. On insistera juste un peu plus sur les méthodes qui seront utilisées dans la suite du travail. En outre, quand cela a été possible, une démonstration des méthodes sur les images que l'on cherche à traiter est proposée, ainsi qu'une analyse critique des résultats obtenus.

Les méthodes générales sont d'abord présentées avec un bref aperçu des méthodes dites « régions » avant un descriptif plus complet des méthodes « frontières » et plus particulièrement les méthodes *top-down* qui utilisent des caractéristiques *a priori* sur la forme globale de l'objet à détecter. Cette présentation bibliographique, à vocation pédagogique, peut être rapidement parcourue par les lecteurs familiers du domaine du traitement des images.
Les nombreuses applications de ces méthodes au problème de la détection du ventricule gauche (qui est en quelque sorte devenu un problème test) sont exposées ensuite. Enfin, cela nous permet d'extraire des enseignements et des recommandations pour la constitution d'un système entièrement automatique d'extraction du contour du ventricule gauche dans des images radiologiques.

8.1 Quelques généralités du traitement d'images

Il est estimé que 75% de l'information reçue par un être humain est de type visuelle [Hall79, p. 1]. La réception et l'utilisation de cette information visuelle par un homme sont généralement décrites comme un processus de perception et de compréhension. Quand un ordinateur reçoit et utilise cette information visuelle, le processus est dénommé traitement et reconnaissance d'images (souvent regroupé à tort sous le vocable « traitement d'images », alors qu'un terme plus approprié est « analyse d'image »). La partie traitement proprement dit se compose des étapes de représentation, amélioration, restauration, compression, et reconstruction. La partie reconnaissance recouvre les tâches de compréhension (description) et d'appariement de scènes. L'histoire de ce domaine est relativement récente, bien que l'on puisse défendre l'idée que le métier à tisser de Jacquard en 1801 effectuait bien un travail sur de l'information visuelle.
L'analyse d'image s'effectue à partir d'images numérisées, c'est-à-dire quantifiées spatialement sur un ensemble de sites discrets appelés pixels (en 2D et voxels en 3D). L'information en chacun de ces sites est soit de type binaire (allumé-éteint), niveau de gris ou code couleur. Les techniques de base employées sont issues du domaine du traitement du signal, mais de nombreuses passerelles vers d'autres domaines scientifiques (neuroscience, intelligence artificielle, statistique, automatique,...) ont été mises en place et de nombreuses méthodes ont été spécifiquement développées pour le traitement et l'interprétation de l'information visuelle.
Par rapport au traitement du signal dit classique, la principale caractéristique des images est d'être multi-dimensionnelle (2 pour les images, 3 pour les volumes, voir 4 avec l'ajout de l'information temporelle). En fait, S. Philipp propose dans [Chehikian97] trois modèles de représentation de l'information visuelle présente dans une image qui peut être considérée :

- comme un signal bidimensionnel $I(x, y)$;
- comme une surface dans un espace à trois dimensions $(x, y, I(x, y))$;
- comme la réalisation $x = (x_s)_{s\in\mathcal{S}}$ d'un processus stochastique $X = (X_s)_{s\in\mathcal{S}}$, avec $\mathcal{S}$ l'ensemble des sites (des pixels) de l'image.

Les outils employés et développés dans ce travail visent à la détection automatique d'un objet dans des images numériques. Cette problématique est donc une tâche typique de reconnaissance. La section suivante présente les grandes familles de méthodes utilisées généralement pour permettre la détection d'objet dans des images statiques. Pour ne pas interrompre le discours, certains détails de fonctionnement sont regroupés dans l'annexe B.

8.2 La détection d'objets

La détection automatique d'objets dans les images numériques est un problème très important pour de très nombreuses applications dans de nombreux champs d'activité (sécurité et défense, biologique et médical, vision industrielle et contrôle qualité, météorologie, interaction homme-machine,...). On peut classer les méthodes de détection en deux grandes familles : les approches orientées vers les régions dans l'image et celle fondées sur leurs frontières. La taxinomie suivante reprend celle développée dans [Cocquerez95].

8.2.1 Les approches régions

Le principe général de ces méthodes dites de « segmentation » est de partitionner l'image en des zones homogènes (au sens d'un certain critère) appelées régions (ou classes). On pourra trouver en annexe (annexe B.1, p. 183) un aperçu de ces méthodes.

Cependant, la recherche de régions homogènes est vouée à l'échec dans le cas des ventricules gauches de notre étude. En effet, les images obtenues par ventriculographie radiologique (voir la figure 7.6) présentent le ventricule sous la forme d'un ensemble de tourbillons crées par le flux du sang coloré par le produit de contraste et, de plus, la densité de ce produit dans le sang est loin d'être uniforme. Loin d'apparaître sous la forme d'une zone homogène en niveaux de gris, le ventricule est donc difficilement détectable comme une seule zone par les approches régions. La deuxième critique à l'utilisation des approches régions est plutôt d'ordre méthodologique puisque ce type d'algorithme se base presque exclusivement sur de l'information à caractère local, et il est donc difficile d'y introduire des connaissances globales, par exemple sur la forme de l'objet à détecter.
Nous nous tournons donc vers les approches frontières.

8.2.2 Les approches frontières

Le principe général de ces méthodes est de détecter les variations brusques d'intensité, c.-à-d. les parties hautes fréquences de l'image. Au lieu de chercher des zones homogènes dans l'image, on cherche plutôt les frontières entre ces régions.

D'un point de vue général, les approches frontière peuvent être regroupées dans deux blocs génériques :

- les méthodes *bottom-up* qui utilisent de l'information locale bas niveau pour construire une représentation d'un niveau conceptuel plus élevé. Les plus connues sont les méthodes dérivatives qui se fondent sur le calcul du gradient des points de l'image via une estimation discrète des dérivées. Le problème principal est alors de chaîner de façon cohérente tous les points de gradient fort pour constituer une frontière continue. On peut également citer les méthodes de morphologie mathématiques et les méthodes Markoviennes (voir l'annexe B).

- les méthodes *top-down*, de nature plus globale, qui prennent en compte dès le départ de l'information sur l'objet à détecter. Ainsi, les méthodes variationnelles utilisent une notion de structure de contour et la phase de détection consiste en la recherche d'appariement entre le contour et l'information contenue dans l'image (elles concernent principalement l'ensemble des techniques de contours actifs).

D'un point de vue méthodologique, les méthodes dérivatives constituent une sorte de prérequis, compte tenu du fait qu'elles visent à fournir une information de dérivée qui est utilisée ensuite par la quasi totalité des méthodes d'extraction de frontières. Elles sont donc décrites plus en détail dans le paragraphe suivant. La morphologie mathématique est ensuite présentée de façon succinte, avant un rapide descriptif des méthodes *top-down* et plus particulièrement des contours actifs.

Les méthodes dérivatives

Intuitivement, dans une image numérique, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes. Une manière d'aborder ce problème est de considérer que l'image numérique résulte de l'échantillonnage d'une fonction scalaire $I(x, y)$ à support borné et dérivable en tout point. Les approches dérivatives sont les plus immédiates pour détecter et localiser les variations du signal, les contours étant assimilés aux points de fort gradient ou de dérivée seconde nulle. Au point M de coordonnées (x, y), le vecteur gradient est défini par

$$G(x, y) = \nabla I(x, y) = (G_x \; G_y)^t, \tag{8.1}$$

avec $G_x = \frac{\partial I}{\partial x}$ et $G_y = \frac{\partial I}{\partial y}$. Ce vecteur gradient est normal à la courbe de niveau $I(x, y) = C^{\text{ste}}$ passant par M. L'orientation de ce gradient par rapport à l'axe horizontal est défini par

$$\phi = \tan^{-1}\left(\frac{G_y}{G_x}\right), \tag{8.2}$$

comme représenté dans la figure 8.1. D'un point de vue pratique, la dérivation

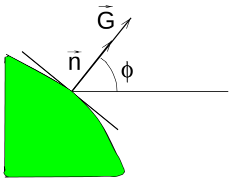

FIG. 8.1: *Le vecteur gradient : définitions.*

d'un signal échantillonné peut être considérée comme équivalente à l'application d'un filtre dérivée de gaussienne au signal original, suivie d'un échantillonnage. En fait, dans le cas discret, une dérivation consiste d'abord en l'application d'un noyau gaussien qui permet de régulariser l'image et la dérivation proprement dite est ensuite approchée par de simples différences finies.
Dans le cas du filtrage linéaire par des filtres à réponse impulsionnelle finie (RIF),

la dérivation s'apparente donc à la convolution de l'image I par un masque de convolution carré h de taille N ce qui s'effectue selon

$$h * I(x,y) = \sum_{i=-N}^{N} \sum_{j=-N}^{N} h(i,j)\, I(x-i, y-j)\ . \tag{8.3}$$

À titre d'exemple, les dérivées directionnelles horizontale et verticale peuvent être obtenues par convolution avec les opérateurs de Prewitt ou de Sobel

$$h_j = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \qquad h_i = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix} \tag{8.4}$$

avec $c = 1$ pour les masques de Prewitt et $c = 2$ pour ceux de Sobel. La figure 8.2 présente le résultat de l'application du filtre de Sobel à une image de ventriculographie radiologique (réduite à une région d'interêt excluant les bords du tube d'intensificateur d'images).

De nombreux autres opérateurs ont été définis pour différentes orientations (Kirsh), pour combiner lissage et dérivation (MDIF, NAGDIF) et pour calculer les dérivées d'ordres supérieurs (voir [Cocquerez95, p. 107-120]). Outre les filtres RIF linéaires présentés jusqu'alors, il existe de nombreux autres filtres qui sont évoqués dans l'annexe B.2, p. 187.

Dans tous les cas, les méthodes dérivatives calculent une approximation des dérivées premières ou secondes qu'il faut ensuite seuiller pour obtenir les contours proprement dits. La première étape de ce seuillage vise à supprimer les points trouvés qui ne sont pas des maxima locaux. Pour ce faire, il suffit de comparer la norme du gradient au point M considéré à celle de points voisins situés dans la direction du gradient, de part et d'autre de M. Le point M est conservé comme point de contour si la norme de son gradient est bien un maximum local, comme schématisé dans la figure 8.3. Il est ensuite possible de seuiller les gradients résultants, la méthode la plus robuste étant un seuillage par *hystérésis* comportant deux valeurs de seuil S_1 et S_2 ($S_1 < S_2$). Seuls les maxima locaux dont la valeur de gradient est supérieure à S_1 et qui appartiennent à une composante connexe comprenant au moins une valeur supérieure à S_2 sont alors conservés.

D'un point de vue général, les contours sont considérés comme des lignes de crête dans l'image de la norme de gradient. Ces lignes de crête doivent alors posséder les caractéristiques suivantes :

- faibles dénivelés le long de ces lignes ;
- forts dénivelés dans les autres directions (particulièrement perpendiculaire).

Outre ces critères, une étude effectuée au LEP [(33)] utilise de l'information de régularité sur les contours en essayant de chaîner les points de gradient important en fonction de leur orientation respective et de leur distance. L'objectif

(33). par S. Makram-Ebeid, communication privée.

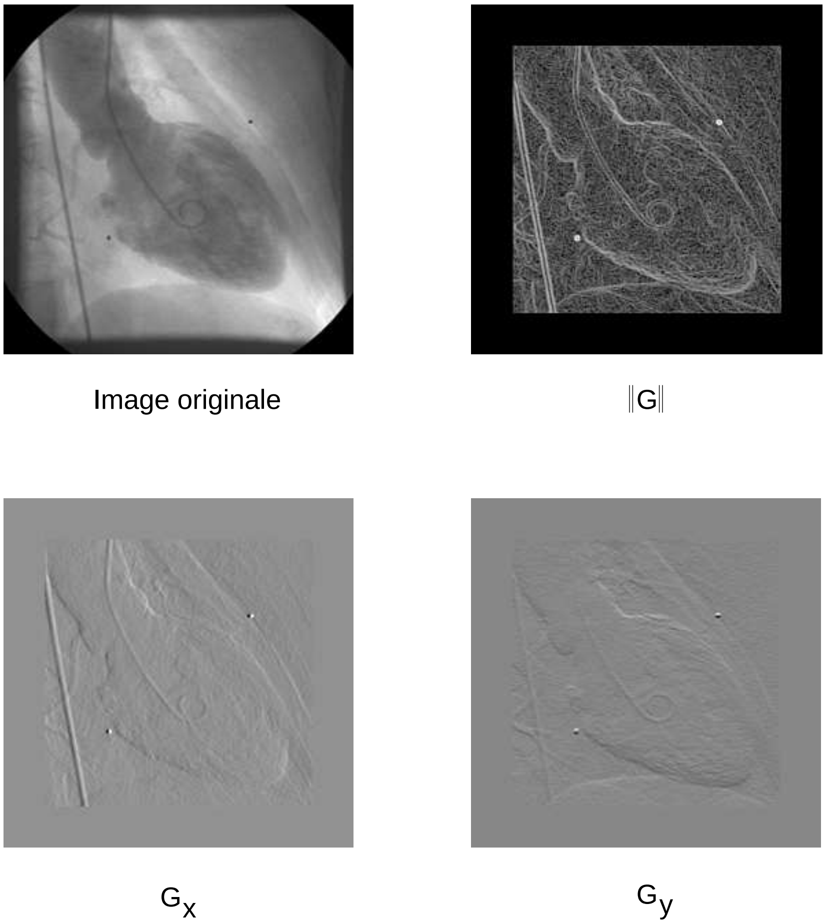

FIG. 8.2: *Exemple d'application du filtre de Sobel.*

est de favoriser la construction de contours longs et de direction bien marquée et ce, par utilisation de techniques issues de la programmation dynamique. Par balayages successifs de l'image dans différents directions et sens, une causalité est définie, et pour chaque point de gradient fort, on cherche à lui trouver les meilleurs prédécesseur et suiveur selon un critère de distance entre les points et d'homogénéité des directions de gradient. À titre d'illustration, la figure 8.4 présente les contours trouvés par cette méthode sur un ventriculogramme. On peut remarquer sur cette image tout l'intérêt de cette méthode de programmation dynamique qui permet de construire des contours longs et relativement rectilignes. Cependant la limite de l'objet ventricule est morcelée sur plusieurs brins de contour et de nombreux autres contours longs sont extraits. Afin d'améliorer cette technique, il faut utiliser de l'information globale afin de déterminer parmi les contours extraits quels sont ceux qui délimitent l'objet et également afin de les chaîner entre eux pour obtenir le contour unique recherché. Plusieurs idées développées dans ce travail ont été mises à profit pour déterminer la structure du système proposée (voir chap. 9).

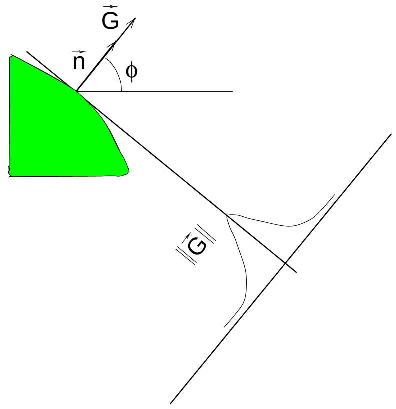

FIG. 8.3: *Principe de l'élimination des gradients non maximaux.*

Les pyramides de filtres

Après l'application d'un filtre passe-bas, il est possible de sous-échantillonner et de calculer ainsi des images de plus en plus petites. On obtient alors une pyramide de filtres. Bien entendu, à partir de l'image la plus petite et d'une pyramide d'erreur, il est théoriquement possible de reconstruire l'image d'origine. L'idée générale est donc de décomposer l'image sur une base différente, comme proposé dans [Mallat89]. On peut également combiner cette pyramide avec différentes tailles de filtre passe-bas pour arriver à une représentation multi-échelle, multi-résolution de l'image. Afin d'extraire des caractéristiques intéressantes, on peut également effectuer le calcul de dérivées pour chacune de ces échelles et chacune de ces résolutions (pour une description mathématique de ces pyramides, voir p. 125 et pour la description d'un mécanisme de reconstrution de l'image, voir l'annexe B.3, p. 189).

La morphologie mathématique

La morphologie mathématique ne relève pas des mêmes domaines que ceux utilisés par les méthodes décrites précédemment. Elle repose en effet sur la théorie des ensembles (et utilise également des notions développées pour la topologie et les probabilités). La morphologie repose sur le concept de forme, assimilé à la notion de structure au sens mathématique du terme. Ce concept nécessite un ensemble de relations ensemblistes (inclusion, intersection) ainsi qu'un objet de référence : l'élément structurant. Caractériser la forme d'un objet revient alors à étudier les relations ensemblistes existant entre l'image de cet objet et l'élément structurant, utilisé comme une sonde parfaitement déterminée.

Dans les cas des images à niveaux de gris, l'image est ainsi assimilée au sous-graphe de sa fonction « intensité », c.-à-d. à une surface de type topographique. Les transformations morphologiques se présentent sous la forme d'un édifice constitué de primitives (les éléments de base que sont dilatation et érosion) et de principes de construction de primitives d'ordre supérieur. Résoudre un problème d'analyse d'images revient alors à choisir des primitives et à les assembler pour

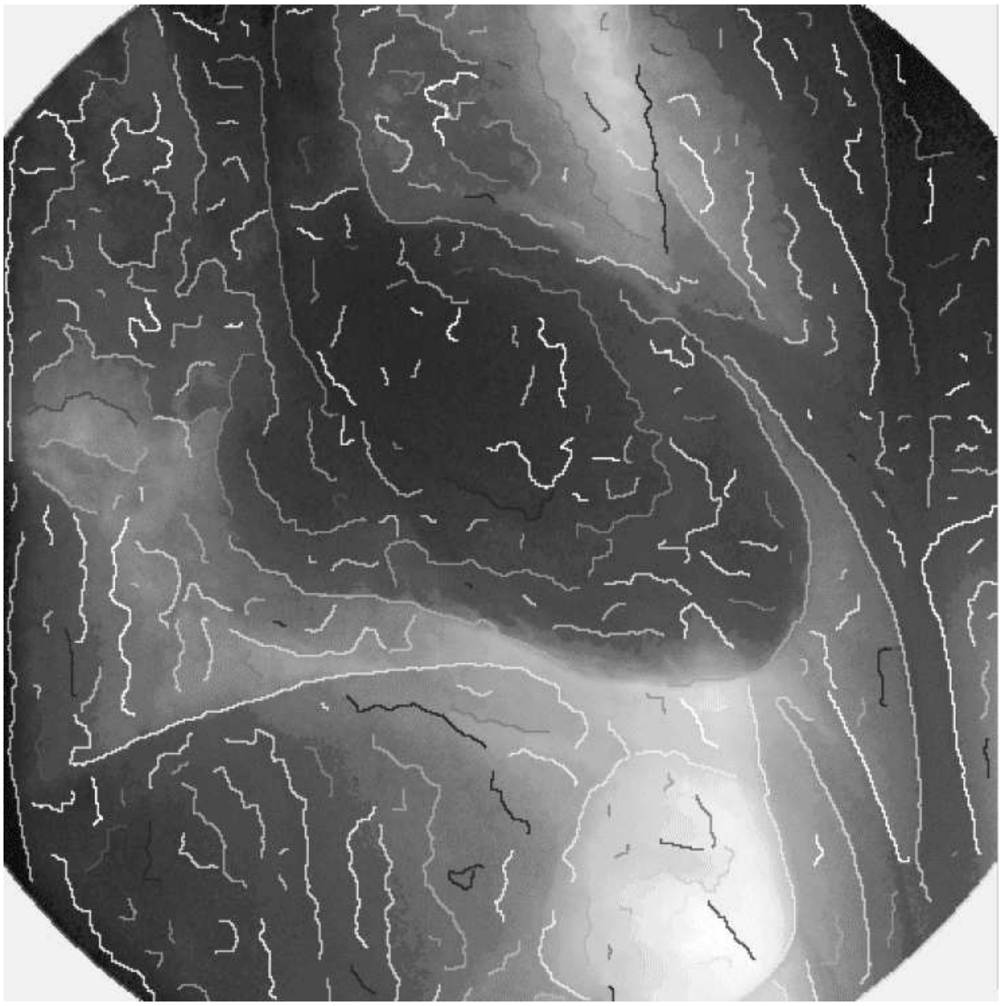

FIG. 8.4: *Contours détectés par utilisation de l'information de distance et de direction.*

progressivement éliminer de l'image tout ce qui ne correspond pas à ce que l'on cherche. Il est aussi possible de choisir des éléments structurants de différentes tailles et ainsi d'intégrer la morphologie mathématique dans un contexte multi-échelle (voir [vdB94] et [Buteau98]).

La morphologie mathématique permet d'utiliser des outils construits spécifiquement pour les tâches à effectuer. Elle sera utilisée dans la section 9.2.2, p. 122 pour « supprimer » une région indésirable de l'image possédant des caractéristiques très particulières. Elle sera également utilisée pour construire une carte de distance et une carte d'indexs (section 9.4.2, p. 144).

Une description plus complète du formalisme ainsi que des techniques de calcul de ce type d'outil est présentée en annexe B.4.

8.2.3 Les méthodes *top-down*

Les méthodes d'extraction de contours présentées jusqu'à présent sont toutes de type *bottom-up*. Elles procèdent de manière « aveugle », sans utiliser d'information sur le contexte ou de connaissances sur la forme des objets recherchés.

Dans certains cas, de l'information *a priori* est introduite, mais sous une forme non explicite, comme la structure de la fonction de coût utilisée dans les méthodes variationnelles ou la forme de l'élément structurant dans les méthodes de morphologie mathématique. Il reste cependant délicat de passer à une description globale de la frontière des objets.

Une méthode utilisant plus d'information sur les objets à détecter consiste à construire et à utiliser un modèle d'objets. L'approche est alors beaucoup plus globale et s'effectue de manière descendante : à partir d'une initialisation de l'objet sur l'image, les informations extraites localement sont utilisées pour mettre en correspondance le modèle d'objet avec la scène présente. Suivant la niveau de connaissance disponible sur l'objet recherché, plusieurs types de modèles ont été développés.

Les modèles rigides

Dans le cas où l'on dispose d'informations très précises sur la forme des objets recherchés, les techniques basées sur les modèles rigides peuvent être utilisées avec profit. Les structures cherchées sont alors définies par une forme générique et mises en évidence à l'aide de méthodes de corrélation ou de mises en correspondance. Une méthode classique consiste à estimer les paramètres d'une transformation rigide (rotation, translation, facteur d'échelle,...) afin de faire correspondre au mieux le modèle avec les informations présentes dans l'image. Cette méthode est difficilement applicable aux images médicales à cause de la très grande variabilité des organes recherchés (taille, forme, position) et de l'image elle-même (équipement d'acquisition, bruit, faible contraste, non reproductibilité totale de l'examen, produit de contraste,...). Pour contourner efficacement ces difficultés, il est préférable d'employer une classe plus riche de modèles : les modèles déformables. Les frontières des objets sont alors représentées par une forme générique qui est déformée afin de détecter et de localiser avec précision les contours. Deux méthodes de modèles déformables sont présentées dans ce qui suit.

Les modèles paramétriques

Une première solution pour prendre en compte les variations de forme des objets cherchés consiste à ne plus les spécifier de manière prédéterminée et rigide, mais à les décrire mathématiquement par un nombre restreint de paramètres. L'expression mathématique des modèles paramétriques est généralement simple ; la caractérisation optimale des paramètres à l'aide des informations présentes dans les images constitue en revanche un problème difficile.

Lorsque les objets à détecter ont une forme relativement simple, le choix d'un modèle peut se limiter à une surface mathématiques paramétrée, ce qui contraint la famille de fonction (Ex : polynômes de Fourier). Un bon compromis entre le nombre de paramètres et la généricité du modèle est trouvé en utilisant des formes de type quadratiques étendues, telles les quadriques, superquadriques puis enfin les hyperquadriques [Vaerman97b]. Ces modèles possèdent les avantages de décrire une très vaste famille de formes (ellipsoïdse, hyperboloïdes, les tores cubes, sphères, et beaucoup d'autres surfaces moins régulières obtenues par composition de formes) avec relativement peu de paramètres et de définir des surfaces continues et différenciables. Ils sont, par exemple, utilisés pour modéliser les membres du corps humain. L'optimisation des paramètres s'effectue de façon classique, à partir de la défintion du fonction de distance, par la méthode de Levenberg-Marquadt (qui nécessite le calcul des dérivées d'ordre un et deux), mais peut également être réalisée par algorithme génétique ou toute autre méthode d'optimisation. Cette modélisation est intéressante et est appliquée à certains types d'images (de volumes) du ventricule, comme présenté dans le § 8.3. Cependant, pour les images rayons X et pour le degré de qualité recherché, ces méthodes nécessitent l'utilisation de trop nombreux paramètres pour estimer de façon robuste les déformation locales. Elles sont, en quelque sorte, de nature trop globale.

La transformation de Hough [Hough62] s'affranchit en partie du bruit et des changements de formes par une démarche originale. Au lieu d'utiliser directement les valeurs des pixels, les configurations les plus probables sont détectées directement dans l'espace des paramètres. Le principe général est le suivant : pour tous les points d'intérêts de l'image, toutes les valeurs (discrétisées) des paramètres qui correspondent à un objet passant par ce point sont accumulés. Cette accumulation d'évidence (de vecteurs compatibles) permet une certaine tolérance aux distortions et donne de bons résultats même si certaines données sont bruitées. En pratique, la transformation de Hough n'est utilisée que pour la recherche de formes très simples (lignes droites, par exemple) dans des scènes peu complexes. Sa mise en œuvre pose en effet des problèmes de combinatoire : une discrétisation fine de l'espace des paramètres conduit rapidement à d'énormes volumes de données. Pour plus de détails, le lecteur curieux pourra se reporter aux références suivantes [Illingwort88][Maître85]. Là encore, l'application de cette méthode au problème de la détection des contours du ventricule gauche n'est pas directement réalisable, compte tenu de la trop grande variabilité des formes de cet objet.

La caractérisation optimale des paramètres peut également être présentée comme la minimisation d'une certaine fonction de coût. Ainsi, [Yuille92] propose de synthétiser les connaissances *a priori* sur la géométrie des structures cherchées sous la forme d'un « gabarit » (*template*) qui reflète les relations entre les divers constituants d'un objet. La fonction de coût à minimiser est alors une énergie qui décrit les interactions entre les images et le gabarit afin de déformer ce dernier de manière à le superposer au mieux sur l'image. Il existe une relation formelle entre gabarits déformables et transformation de Hough lorsque les relations entre les composants du modèle de l'objet deviennent quasi-rigides. Cette méthode ne s'applique pas de façon optimale pour le problème qui nous intéresse puisque l'objet recherché est mono-composant.

Il est également possible d'utiliser des méthodes stochastiques de minimisation. Les paramètres sont alors considérés comme des variables aléatoires, ce qui accroît de façon importante la souplesse des modèles paramétriques. Ce point de vue stochastique est en fait intermédiaire entre les modèles paramétriques et les modèles élastiques présentés dans la section suivante.

Les modèles élastiques

Ces modèles ont été introduits par Kass, Witkin et Terzopoulos [Kass87, Kass88]. Le principe est de représenter les frontières des objets par une courbe ayant des propriétés mécaniques de type membrane élastique. Dans ce qui suit, les termes contours actifs, modèles élastiques et contours déformables seront utilisés de manière équivalente. Pour un panorama assez complet de l'utilisation des modèles déformables dans les images médicales, le lecteur pourra se reporter à [McInerney96] et [Singh98].

Les contours actifs présentent un intérêt réel pour la mise en correspondance de structures entre deux images acquises par des capteurs différents, comme par exemple, les travaux d'Henri Maître ([Maître87b] et son application au recalage d'images satellitaires à des images de cartes topographiques [Maître89]). Il est également possible de les utiliser pour effectuer un suivi temporel de contours dans des séquences d'images, les résultats au temps t servant d'initialisation pour le temps $t+1$. Des termes d'énergie modélisant les transformations peuvent également être ajoutés, en utilisant de l'information *a priori* sur l'objet à suivre. Cependant, le domaine d'application principal des contours actifs est assurément la segmentation.

Les déformations sont dues d'une part aux contraintes internes (tension, courbure de la frontière) et d'autre part aux contraintes externes (attache aux données : influence de l'image). La recherche d'un contour s'assimile donc à la minimisation itérative d'une fonctionnelle définie comme une intégrale le long du chemin emprunté par le contour, $E(v) = \int_0^1 E_{\text{int}}(v(s))ds + \lambda \int_0^1 E_{\text{ext}}(v(s))ds$. Les forces internes E_{int} décrivent, par analogie avec la mécanique, les forces caractéristiques de la membrane, comme la flexion et l'élongation [(34)]. Le terme d'énergie externe E_{ext} découle d'un potentiel scalaire local défini sur l'image observée. Le potentiel le plus simple et le plus utilisé pour la détection des contours est basé sur la norme du gradient. À partir d'une initialisation, les courbes élastiques vont se déplacer de manière à se superposer aux structures pertinentes des images traitées tout en respectant les contraintes internes de la membrane. Cette solution permet de rendre plus fiable la détection des contours puisqu'elle s'affranchit un peu de la localité pour chaîner des points intéressants tout en gardant une forme globale cohérente. Bien entendu, le problème principal est la définition adéquate des énergies en fonction de connaissances sur la régularité des frontières et sur l'information présente dans l'image.

L'étape de minimisation peut être conduite sous l'égide de méthodes variationnelles, à partir de la définition d'équations d'évolution définies à l'aide des équations d'Euler-Lagrange [Bossard94].

(34). on peut également chercher à introduire une dynamique, comme une force de « gonflement » [Cohen91].

On distingue deux grandes catégories de contour actif : les modèles continus et les modèles discrets. Les modèles continus s'appuient sur une représentation analytique du contour, le plus souvent obtenue par approximation de type *spline* reliant les points de contrôle sélectionnés à l'initialisation. C'est dans ce cadre analytique que l'approche variationnelle de la minimisation prend tout son sens. Le modèle discret, quant à lui, associe la courbe initiale à une ligne polygonale dont les sommets sont des points discrets de l'image. Ainsi, la définition d'espaces de recherche pour chacun des points permet d'effectuer la minimisation en une seule passe par programmation dynamique [Bellman62].

8.2.4 La programmation dynamique dans les images

L'algorithme de Viterbi permet de déterminer le chemin de coût minimal (minimum global) dans un graphe orienté, compte tenu de l'espace de recherche considéré. Il faut cependant déterminer une causalité (une orientation) afin d'appliquer le principe de Bellman qui affirme que toute sous-partie d'un chemin optimal est elle-même optimale. C'est cette propriété qui permet la construction du contour global par agrégation. Il existe quelques travaux qui visent à une programmation dynamique bidimensionnelle, mais ceci sous la double contrainte de restreindre les mouvements des points et de recherche de forme circulaire de taille variable [Yamada88].

Dans la plupart des travaux, la causalité est définie en fonction de l'abscisse curviligne le long du contour et l'espace de recherche est considéré perpendiculaire au contour en chacun des points. La programmation dynamique permet de trouver le « meilleur » contour dans une configuration donnée, puis le contour étant modifié, l'espace de recherche l'est également et la programmation dynamique est à nouveau appliquée, et ce, jusqu'à convergence. Cette méthode itérative a été intialement proposée dans [Amini90] pour être ensuite étendue [Geiger95].

L'application de la programmation dynamique pour la reconnaissance dans les images date du début des années 1980.

H. Maître a le premier proposé de combiner la programmation dynamique avec des modèles auto-régressifs simples et d'introduire un état « réfrigérateur » qui permet d'obtenir des chemins interrompus [Maître87b, Maître87a].

L'utilisation combinée de la programmation dynamique et des modèles auto-régressifs a été ensuite utilisée dans [He91] (au sein d'un formalisme HMM) pour classifier des « formes » 2D. L'utilisation des modèles auto-régressifs permet de coder simplement la forme de segments du contour (par des caractéristiques considérées comme localement stationnaires). Les observations du modèle HMM sont des coefficients de modèles auto-régressifs qui sont définis entre les positions constituées d'une séquence de rayon dans une transformation polaire.

Olstad et Torp ont présenté un travail très intéressant visant à introduire des modèles grammaticaux dans une phase d'extraction de contours [Olstad96]. Ce travail cherche à introduire de l'information *a priori* sur la forme du contour à extraire à l'aide de descripteurs simples et locaux, comme la courbure. La recherche du contour optimum contient donc un coût lié à l'enchaînement de points de caractériques différentes. Un objet particulier peut être alors décrit sous une forme syntaxique, comme par exemple, la succession de points presque alignés (de courbure faible, « classe » a) suivie d'un coin à angle droit (angle de courbure proche de 90°, « classe » b) et cela répété trois fois ; le carré est donc décrit sous la forme $a^*ba^*ba^*ba^*$. Cette méthode est très intéressante car l'introduction de connaissance *a priori* sur ce que l'on cherche permet de rendre l'extraction plus robuste au bruit et de discrimimer le bon objet en cas d'occlusion ou de présence d'autres objets.

Il existe de très nombreux travaux sur l'utilisation de méthodes proches des contours actifs pour détecter des contours dans des images. Ainsi le formalisme des *Level Set* [Sethian96] qui considère les contours d'un objet comme des iso-niveaux d'une fonction scalaire échantillonnée sur la grille de l'image. Il est alors possible d'utiliser les équations aux dérivées partielles pour faire évoluer cette fonction (et donc les contours qui sont définis implicitement) de telle sorte qu'un certain niveau de cette fonction converge vers le contour recherché. En outre, de ce formalisme découle le *fast marching* [Cohen96] qui permet de détecter un chemin « minimal » dans une image [Deschamps99] avec une méthode très proche de la programmation dynamique. Ces travaux sont très intéressants, mais en dehors des limites de cette thèse.

8.3 Les applications aux images de ventricules cardiaques

Les images de ventricules sont l'objet de très nombreuses études utilisant des techniques d'extraction de contours très variées. Ces images constituent un problème de référence, tant pour la détection automatique que pour la reconstruction tridimensionelle et l'étude de l'évolution temporelle du mouvement du muscle cardiaque.

8.3.1 Les images de ventriculographie radiologique

Cette technique de visualisation constitue la référence dans l'étude des paramètres morphologiques et cinétiques des ventricules (surtout du gauche), voir § 7.1.4, p. 85. Les travaux sur la détection des contours dans de telles images sont donc très nombreux depuis le milieu des années 80.

Parmi les travaux de ce type, les modèles développés par l'équipe de J.H.C. Reiber de l'hôpital universitaire de Leiden sont particulièrement intéressants [v. Leeuwen86] [van der Zw92] [Dumay92]. Les techniques utilisées sont basées sur la programmation dynamique. Ainsi, [v. Leeuwen86] construit un modèle moyen de contour de ventricule gauche (à partir de 10 patients). Ce modèle est transformé dans l'image à traiter par l'intermédiaire de trois points clés préalablement déterminés par le radiologue et permet de déterminer un premier contour grossier. Ces trois points sont les deux extrémités de la valve aortique et l'apex. Ce modèle permet d'introduire un ordre dans le balayage, partant de l'extrémité antérieure de la valve, passant par l'apex pour terminer sur la partie postérieure de la valve. Le sous-échantillonnage de ce premier contour grossier et la définition de zones de recherche perpendiculaires à ce modèle permettent la construction d'une matrice de coût, à partir du calcul de gradients aux points candidats. Le contour recherché est alors obtenu par programmation dynamique comme celui permettant d'avoir le coût accumulé minimum pour la traversée de cette matrice. Les résultats publiés paraissent prometteurs mais une validation clinique significative manque. Afin d'isoler les contours du ventricule dans ce type d'images ou beaucoup d'autres organes (côtes, diaphragme, colonne vertébrale, ...) sont (partiellement) visibles, [Dumay92, chap. 7] propose une méthode heuristique basée sur un balayage sélectif de l'image. Les contours rencontrés successivement sont étiquettés dans les différents organes en fonction d'information sur la courbure, la position dans l'image, les positions relatives des contours, ...

Ces méthodes, particulièrement la création d'un modèle moyen et la programmation dynamique, paraissent très riches d'enseignement et donnent de la robustesse à la détection.

L'idée de modèle moyen de contour a été étendue dans [Tehrani91]. À partir d'une large base d'images sur lesquelles les contours du ventricule ont été tracés manuellement par un radiologue, un modèle moyen est construit. La moyenne est calculée pour les paramètres d'échelle et de rotation nécessaires pour faire correspondre au mieux les contours (recalés en translation pour commencer au même point). Un autre type de modèle (plus de type paramétrique, voir § 8.2.3) des contours du ventricule gauche a été présenté dans [Groshong92] ; il est fondé sur un codage sous la forme de série de Fourier elliptique. Ce codage permet de décrire un contour fermé sous la forme d'ellipse et, plus des termes sont ajoutés à l'expansion, et plus le contour peut être déformé vers les points caractéristiques de l'image (voir égalemment [Robert95]). La méthode présente l'intérêt de considérer les paramètres (qui sont dérivables) comme des variables aléatoires. Le calcul du modèle le plus approprié est alors effectué par une descente en gradient.
Ces approches basées entièrement sur la mise en correspondance d'un modèle pré-déterminé avec l'image à traiter ne permettent pas d'introduire toute la souplesse désirée. Le caractère pathologique de nombreux cas risque de conduire la méthode à de mauvaises détections alors même que ce sont ces cas-là qui requièrent la plus grande attention.

La programmation dynamique est une autre idée générique intéressante. La première application de cette méthode à des images radiographiques du ventricule gauche a été présentée dans [Pope85]. Le chemin de coût minimal est recherché dans une image issue d'une transformation polaire $(x, y) \mapsto (\rho, \theta)$, à partir du centre du ventricule déterminé de façon semi-automatique. Les « temps » (les ordres pour les points candidats) sont représentés par la variable d'angle θ, alors que les différents candidats à chaque « coup d'horloge » sont rangés par distance au centre ρ. Le coût local pour passer par un point candidat est estimé à partir de l'information de gradient. Les transitions entre point candidats sont construites pour « des temps » (des angles) adjacents et sont affectées de coûts établis en fonction d'une fonction de distance entre les points. Les résultats présentés paraissent prometteurs même s'ils ne sont établis que pour une partie du contour (à l'exception des régions trop proches de la valve aortique). [de Figueir92] reprend la même transformation mais en utilisant un formalisme

Bayésien pour introduire un *a priori* sur la position du contour.

L'utilisation de la programmation dynamique permet de rendre plus robuste l'extraction de contour. Elle permet d'extraire des contours acceptables même pour des images très bruitées. En effet, la recherche d'un chemin de coût minimal reliant des points de gradient fort permet de surmonter les mauvaises détections locales (gradient fort à tort ou absence de détection) pour aboutir à un contour global satisfaisant. Cependant elle nécessite l'introduction d'une causalité, c'est-à-dire d'un ordre de parcours des points candidats au chaînage. La modélisation sous la forme radiale utilisée dans les deux derniers travaux évoqués, restreint fortement la classe de contours envisageable. Elle est en outre trop sensible à la position du centre de la transformation et surtout, la transformation s'effectue de façon non perpendiculaire aux contours car le modèle sous-jacent est le cercle, alors que cet *a priori* est très discutable pour la plupart des images de ventriculographie radiographiques.

8.3.2 Les images obtenues par d'autres modalités

Pour l'analyse des ventricules cardiaques, et bien que l'examen radiographique soit le plus utilisé, il ne possède pas l'exclusivité de la recherche de paramètres pertinents pour le diagnostic et de nombreuses autres modalités d'imagerie médicale sont utilisées (pour une introduction aux différentes modalités, se référer à [Aubert95] et [Acharya98]). Parmi les très nombreux travaux d'extraction de contour du ventricule déja publiés sur différentes modalités, on peut citer :

- 3D CT (*Computed Tomography*) [Clarysse95] ;
- échographie ultrasonore [Hunter93b] [Hunter93a], [Robert95] et en version 3D [Sheehan98] ;
- médecine nucléaire [Qian98] et SPECT (*Single-Photon Emission Computed Tomography*) [Declerck96] ;
- imagerie par résonance magnétique avec [Young98], les travaux effectués à l'université de Pennsylvannie [Park96], ainsi que les études de la dynamique du mouvement du myocarde [Baldy94], [Geiger95], [Donnell98] et [Vaerman97a].

C'est cette dernière modalité qui reçoit en ce moment le plus d'attention dans les communautés scientifique et médicale.

8.3.3 Conclusion sur les travaux présentés

De tous les travaux présentés jusqu'alors deux enseignements généraux peuvent être tirés.
Premièrement, la totalité des travaux adopte une approche de type frontière pour l'extraction des contours. Cette constatation confirme l'intuition exprimée page 101 : la recherche de régions homogènes effectuée par les méthodes de type région est vouée à l'échec pour ce problème.

Deuxièmement, les méthodes *bottom-up* qui visent à la construction du contour uniquement à partir des informations locales de gradient (voir p. 102) ne peuvent pas fonctionner de manière satisfaisante, compte tenu de l'importance du bruit présent dans les images. Ainsi la presque totalité des méthodes présentées utilise un certain type modèle de l'objet. Ainsi [Tehrani91] construit un modèle rigide du ventricule, des modèles paramétriques ont été développés par [Groshong92] (Fourier elliptique) et par [Donnell98], [Park96], et [Vaerman97a] (modèles de super et hyper quadriques), mais la majorité des modèles utilisés sont de type élastiques ([Baldy94], [Clarysse95], [de Figueir92], [Geiger95], [Hunter93b], [Hunter93a], [Pope85]...). Cette approche par modèle peut également s'expliquer simplement pour le traitement d'images médicales : l'observateur possède une bonne idée de la forme et du contexte de l'objet qu'il recherche. C'est la justification principale aux importants travaux de recherche actuellement effectués pour la constitution d'atlas 3D des organes du corps humain.
Parmi les enseignements plus précis que l'on peut tirer des travaux présentés, les recommandations suivantes méritent d'être relevées pour le développement d'un système automatique :

- le modèle de l'objet doit être suffisament souple pour permettre une extraction correcte, même (et surtout) dans les cas pathologiques ([van der Zw92]) ;
- les modèles élastiques paraissent avoir les caractéristiques nécessaires pour cette souplesse ([Pope85]), à condition de bien utiliser une transformation de type curvilinéaire qui n'introduit pas d'*a priori* biaisant sur la forme à détecter ([v. Leeuwen86]) ;
- la souplesse peut en outre être accrue par l'utilisation d'un formalisme probabiliste ([de Figueir92]) ;
- l'introduction de connaissance *a priori* sur l'objet implique l'adjonction d'un nombre important de paramètres et le délicat problème de leur optimisation peut être efficacement pris en charge par une approche issue du domaine apprentissage ([Hunter93b]).

Chapitre 9

Une approche pour l'extraction automatique des contours du ventricule

L'étude critique des méthodes existantes pour la détection d'objet (et plus particulèrement celles utilisées pour l'extraction des contours du ventricule gauche), présentée dans le chapitre précédent, nous a permis d'extraire quelques caractéristiques désirables pour un système automatique d'extraction de contours du ventricule gauche.

Ce chapitre décrit le système proposé pour extraire ce contour. Ce système utilise le plus possible de l'information *a priori* de haut niveau sur l'objet à détecter pour contraindre la recherche du contour le plus probable [Gérard98]. Cette recherche se base sur un modèle hybride chaîne de Markov - réseau de neurones.

La première partie (section 9.1) décrit les classes anatomiques définies. Ces classes ont une signification pour le médecin, et elles vont être ici utiles pour sélectionner les points et surtout contraindre l'ordre dans lequels ils doivent être chaînés pour obtenir le contour de l'objet.

La deuxième partie (section 9.2) présente les prétraitements effectués sur les images afin d'augmenter la dynamique et de supprimer certains éléments perturbateurs.

La classification des points dans les classes définies est effectuée par un réseau de neurones, présenté dans la troisième partie (section 9.3). Cette étape passe bien entendu par une extraction de caractéristiques pertinentes pour la discrimination.

L'étape d'extraction du contour proprement dite, objet de la quatrième partie (section 9.4), est réalisée par programmation dynamique à partir de la définition d'une chaîne de Markov cachée en 2D originale [Gérard99c].

9.1 L'objectif

9.1.1 Solution proposée

À partir de l'étude de la littérature présentée dans le chapitre précédent et plus particulèrement au dernier paragraphe, nous avons fixé les bases d'un système entièrement automatique pour l'extraction du contour du ventricule gauche dans les images angiographiques. La solution retenue procède en deux grandes étapes :

- l'extraction automatique d'un premier contour approché ;
- la détection plus fine par programmation dynamique, autour du contour inital.

La première étape vise à la détection automatique de points clés dans l'image. Ces points sont ensuite chaînés dans un polygone pour construire un modèle approximatif (une ébauche) du ventricule. La détection de ces points repose principalement sur l'utilisation d'un classifieur neuronal et comporte les phases suivantes :

- calcul de mesure locale de gradient afin de ne conserver que les points de gradient fort ;
- calcul d'une pyramide de filtres multi-résolution afin d'obtenir des caractéristiques non uniquement locales et de bâtir ainsi un parallèle avec la vision humaine ;
- estimation par réseaux de neurones de probabilités *a posteriori* pour des classes anatomiques définies préalablement, à partir des caractéristiques multi-résolution ;
- recherche de maximum sur les probabilités obtenues pour localiser des points clés ;
- chaînage des points clés (en ligne droite) pour obtenir un modèle polygonal du ventricule, puis calcul d'un spline passant par ces points, pour un modèle plus lisse.

La seconde étape consiste en l'extraction du contour proprement dite, dans une zone de recherche autour de l'ébauche initiale. Cette étape utilise une phase de programmation dynamique et possède des similarités avec les modèles grammaticaux.

L'introduction de connaissances de haut niveau est très intéressante pour le problème de l'extraction des contours du ventricule gauche. Il est cependant nécessaire d'utiliser des caractéristiques beaucoup plus riches et de plus haut niveau que la simple courbure (comme dans [Olstad96]).
Le principe de la méthode choisie est l'utilisation la plus complète possible d'information *a priori* sur l'objet à détecter, tant dans ses caractéristiques visuelles que dans sa forme. Elle comporte les phases suivantes [Gérard99b] :

- pour chacun des points de gradient candidats, calcul d'un coût « local » de passage à partir des probabilités estimées par le réseau de neurones, ces probabilités injectant des informations sur le contexte « spatial » ;

- codage d'information globale sur la forme à extraire dans un formalisme HMM [Gérard98] ;
- utilisation d'un nouvel algorithme de programmation dynamique [Gérard99a] qui permet l'emploi de toutes les informations probabilistes pour trouver le contour de coût minimum.

On peut déjà noter que l'approche choisie possède un caractère novateur, puisque l'application des chaînes de Markov aux problèmes de traitement d'images n'est pas courante [Aas99] (35).

Toutes ces étapes sont décrites en détail dans ce chapitre.

(35). « *In image analysis, [...] the application of hidden Markov chains has been rare.* »

9.1.2 Les classes définies

15 classes de points ont été définies, comme présentées dans la figure 9.1. Ces classes peuvent être regroupées comme suit :

- les trois points clés : les parties antérieure et postérieure de la valve aortique : *VA*, *VP* et l'apex *AP* ;
- les points des segments antérieurs : de *BA1* à *BA3* et postérieurs : de *BP1* à *BP4* du ventricule ;
- les points de bords non pertinents : l'aorte *CA* et les « pommes aortiques » (les deux parties du sinus de Vasalva) *PA* et *PP* ;
- les points dans les zones de l'image interne au ventricule *CI* et externe *CE*.

La définition des différents segments du ventricule suit les recommendations de l'*American Heart Association* (voir [Austin75]). En effet, la classe *BP2* correspond à la valve mitrale, *BP3* au segment posterobasal, *BP4* au segment diaphragmatique, *BA2* au segment anterobasal et *BA3* au segment anterobasal. Les classes *BA1* et *BP1* correspondent aux extrémités de la valve aortique. Ces classes correspondent à peu près aux segments décrits dans la figure 7.7, page 90, avec infero comme synonyme de postérieur. L'adjonction de classes non directement pertinentes pour la détection de l'objet permet d'accroître la sélectivité de la classification.

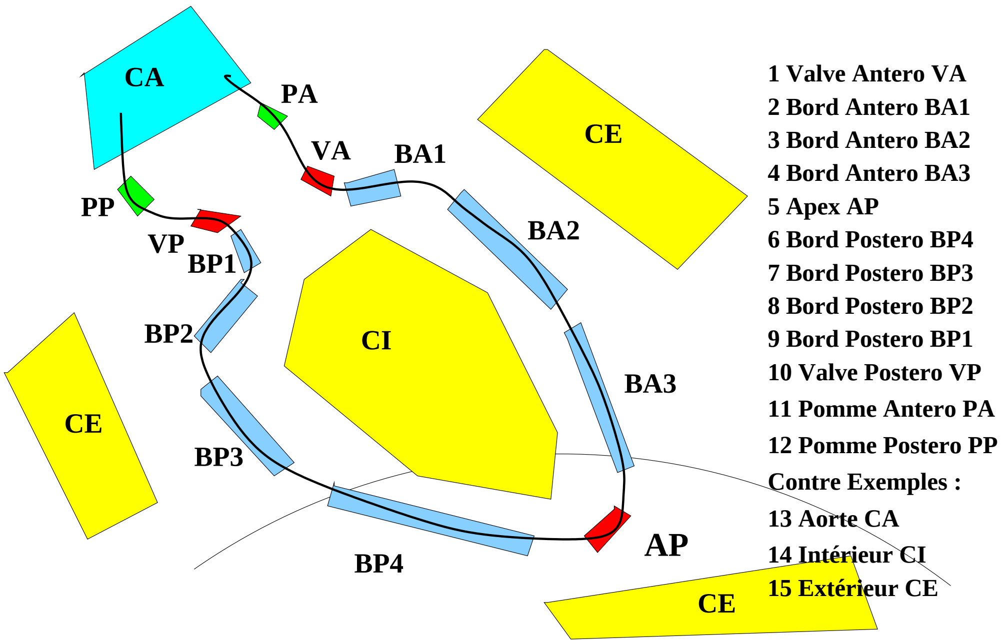

FIG. 9.1: *Les classes anatomiques prédéfinies pour les images de ventriculographie radiographique.*

Le système de classification devra donc être performant sur les ensembles de classes suivants (par ordre de priorité) :

- les 3 points clés : *VA*, *VP* et *AP* ;
- les 7 segments : de *BA1* à *BA3* et de *BP1* à *BP4* ;
- les autres classes : *PA*, *PP*, *CA*, *CI* et *CE*.

Cependant, il est bien évident que par leur définition même, les classes se recouvrent mutuellement et que les frontières les séparant sont mal définies.

9.2 Prétraitements de l'image

La détection de contour s'appuie sur une première phase de détection de bord, c'est-à-dire de calcul de gradient. Ces calculs sont effectués sous la forme de différences entre pixels voisins dans notre cas discret. Ces calculs sont très sensibles au bruit. De plus, comme ils vont être enchaînés, une grande dynamique d'entrée est nécessaire. Or, beaucoup des radiographies numériques utilisées ont une profondeur limitée à 1 octet, c'est-à-dire que chaque pixel a une intensité comprise entre 0 et 255. D'autres ont une profondeur de 2 octets. Le premier travail est donc d'égaliser les différences de profondeur, tout en augmentant la part de la dynamique utile de l'image.

9.2.1 Modification de la dynamique de l'image

La méthode utilisée pour étendre la dynamique repose sur une appoximation linéaire des intensités entre deux bornes. Chacune de ces bornes est calculée à partir de l'histogramme de l'image, c'est-à-dire des nombres de pixels ayant une intensité donnée. Le schéma d'un tel histogramme est présenté dans la figure 9.2. Les bornes sont alors estimées de façon à exclure un certain pourcentage de valeurs extrêmes. Ainsi, α_1 permet de « supprimer » un certain pourcentage des points les plus sombres.

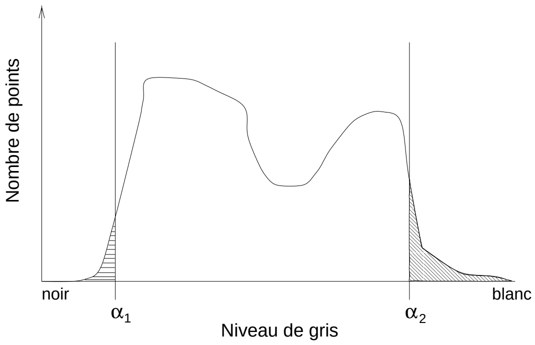

FIG. 9.2: *Principe de la modification d'histogramme.*

La figure 9.3 présente le résultat de cette modification de la dynamique des intensités des pixels pour une image du ventricule gauche. L'imagette du milieu représente les intensités le long d'une ligne horizontale figurée dans les deux images : avant et après la modification.

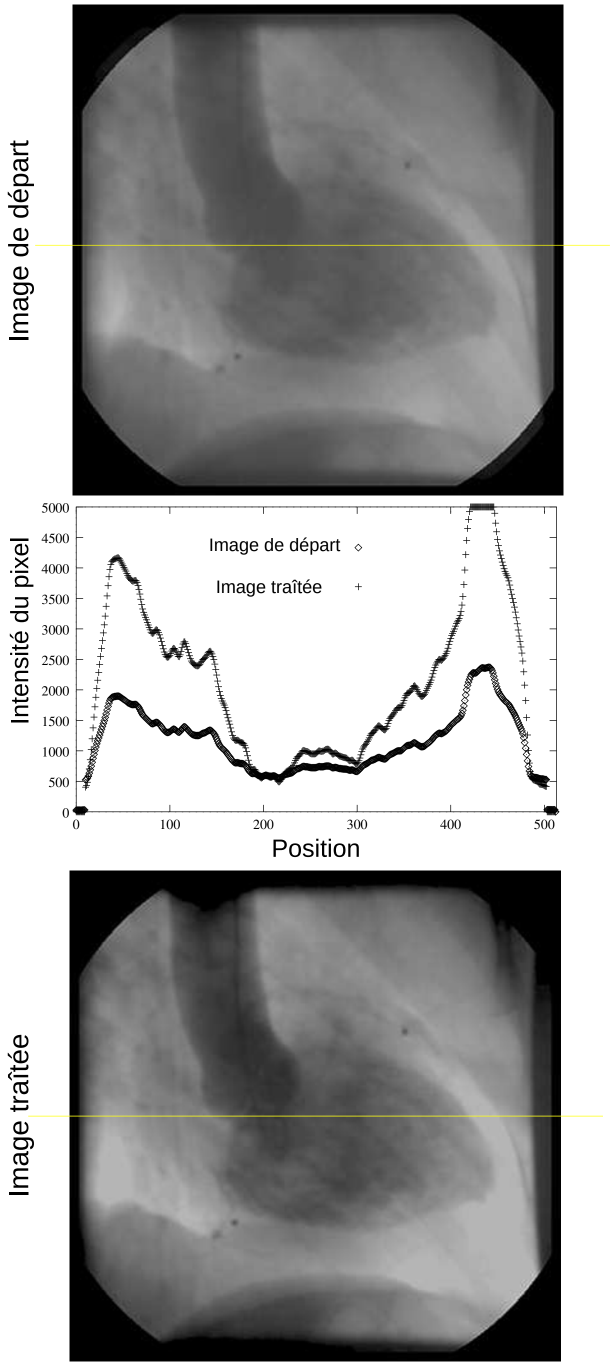

FIG. 9.3: *Résultat de la modification d'histogramme.*

9.2.2 Suppression du cathéter par morphologie mathématique

Comme le cathéter est chargé d'amener le produit de contraste jusqu'à l'organe cible, il est très visible dans les images radiographiques. Ses contours sont donc très nets bien que cette structure n'apporte aucune information pour le diagnostic. On a donc cherché à le supprimer ou du moins à en réduire la différence d'intensité, surtout pour pouvoir comparer les résultats de l'extraction de contour avec les méthodes « classiques » qui n'utilisent que de l'information locale et qui risquent donc d'être perturbées par le cathéter. Le problème est donc d'utiliser un filtre sélectif aux caractéristiques de cathéter. Le cathéter utilisé généralement pour la ventriculographie radiographique est un tube en plastique de 2mm de diamètre (6 French [(36)]). Il apparaît dans l'image comme une bande sombre de largeur constante (environ 7 pixels). Le moyen choisi est alors d'utiliser des filtres discriminants issus de la morphologie mathématique (voir l'annexe B.4 p. 191 pour un aperçu de cette méthode).

Dans le cas du cathéter, on recherche à fermer la partie plus sombre. L'opération à appliquer est alors la fermeture morphologique qui est composée de la succession dilatation puis érosion (telle que schématisée dans la figure B.4(b), p. 193).

La figure 9.4 [(37)] présente le résultat de ce passage avec deux lignes d'intensité (avant et après la fermeture). La position de deux structures de type cathéter est représentée par deux lignes verticales. On peut voir l'effet de la fermeture qui « bouche » les trous d'intensité dus aux cathéters, sans modifier de façon sensible le reste de l'image. En particulier, les gradients intéressants ne sont pas détériorés. C'est tout l'interêt de définir des structures morphologiques (des éléments structurants) adaptées au problème à traiter.

9.2.3 Résultats des prétraitements

Les prétraitements ont permis d'homogénéiser les dynamiques des images et aussi de réduire la confusion possible entre les bords du ventricule et les bords, plus tranchants, du cathéter. Avec un peu d'avance sur le reste du discours, la table 9.1 compare les performances du réseau de neurones (encore à définir) sur une base d'apprentissage (encore à définir) avant et après l'application des prétraitements présentés dans les paragraphes précédents. Elle montre juste la non-dégradation des informations utiles à la tâche de classification. C'est-à-dire que l'approche que nous avons choisie permet de ne pas dépendre de ces prétraitements, contrairement aux méthodes plus classiques qui dépendent entièrement d'informations locales et qui pourraient donc confondre le cathéter avec un bord marqué du ventricule. Cette étape est donc plus utile pour ne pas biaiser la comparaison que nous allons effectuer entre les résultats obtenus par notre méthode et ceux obtenus par la méthode classique.

(36). French (ou unité Française) est l'unité de mesure usuelle des diamètres externes des cathéters : 1 F = 0,33 mm.

(37). pour des raisons de lisibilité des deux images, leur dynamique *d'affichage* ont été modifiées manuellement et différemment.

	Avant	Après
Perf. apprentissage	91,4	92,3
Perf. test	66,6	69,1

TAB. 9.1: *Performances avant et après les opérations de prétraitements.*

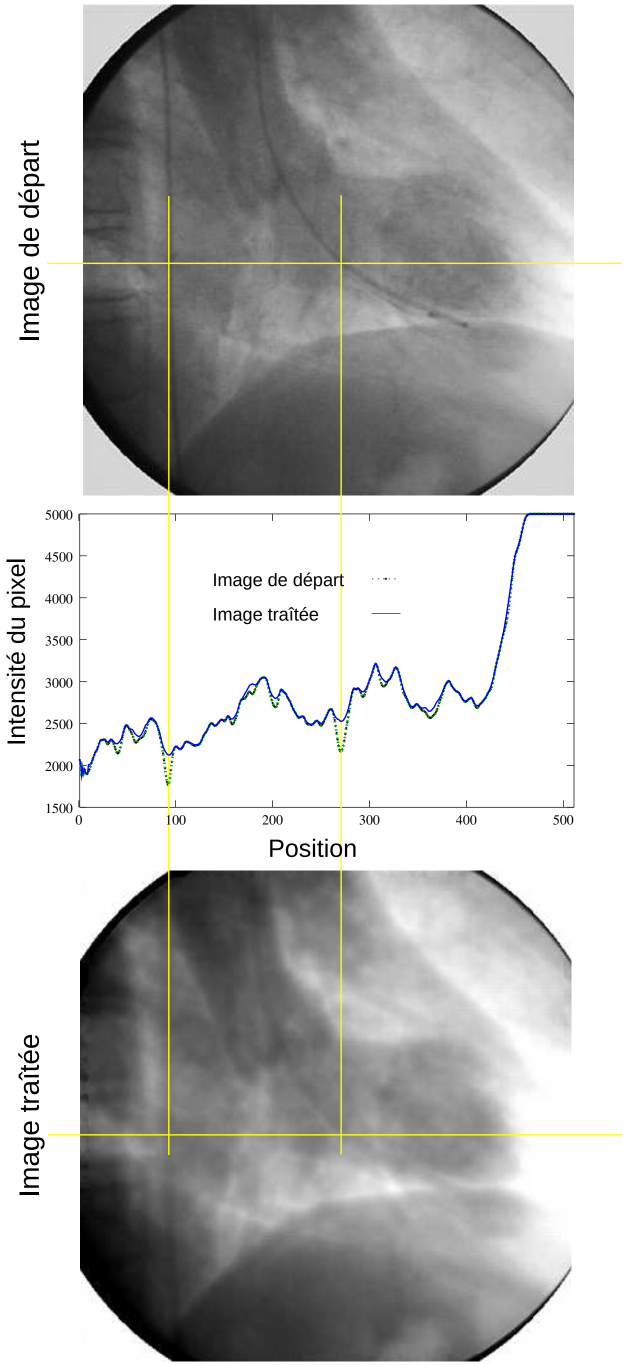

FIG. 9.4: *Résultat du passage des filtres morphologiques pour la suppression du cathéter.*

9.3 Classification des points par réseaux de neurones

9.3.1 Extraction de caractéristiques

L'objectif est ici d'extraire de l'image des mesures discriminantes. La première étape de cette recherche se base sur une comparaison avec le traitement effectué par l'œil humain, très performant pour ces tâches de reconnaissance de formes. La seconde cherche à réduire la quantité de données afin de ne trouver que les caractéristiques les plus discriminantes. L'approche choisie se fonde sur la constitution d'une représentation multi-échelles, multi-résolutions.

Le système visuel humain

Il est estimé que 75% de l'information reçue par un homme est de type visuel [Hall79, p. 1]. La première étape de traitement de cette information visuelle est réalisée au niveau même du capteur : la rétine qui comporte environ 6,5 millions de cônes, 120 millions de bâtonnets pour 1 million de fibres dans le nerf optique [Schmitt97] [Chehikian97]. Les cônes sont les cellules photo-sensibles de la rétine responsables de la vision diurne et de celle des couleurs alors que la vision nocturne est due aux bâtonnets.

La fovéa, au centre de la rétine, est constituée d'une dépression d'environ 1 mm de diamètre (1° de vision angulaire). Cette zone est très importante puisqu'elle constitue le point de fixation de notre regard, là où notre acuité à discerner des petits détails est la plus forte [Hubel88]. Elle reçoit donc un traitement spécial dans l'œil puisque chacun des cônes (environ 34000) présents dans cette région est relié à une cellule ganglionnaire dont l'axone constitue une fibre du nerf optique. Chaque cellule photo-sensible de cette zone a donc une ligne directe vers les centres visuels du cerveau où les opérations plus complexes sont réalisées. Il existe cependant des interactions latérales de type inhibiteur entre cellules voisines. Ces interactions sont responsables de l'antagonisme centre-périphérie des champs récepteurs.

La figure 9.5 représente cet effet par le biais de la réaction d'une cellule ganglionnaire en fonction de la position d'un petit spot lumineux. C'est une fonction de type « chapeau mexicain » qui est généralement représentée par la différence de deux gaussiennes (de rapport de σ de l'ordre de 1,6 [Marr82, p. 62]). La réponse présentée dans la figure 9.5 est celle d'une cellule *on-center*, mais il existe également des cellules *off-center* qui sont excitées pour la périphérie de leur champ récepteur et inhibées au centre.

Lorsque l'on s'éloigne de la fovéa, vers la périphérie de la rétine, il y a au contraire convergence du câblage cellulaire, une fibre du nerf optique pouvant être connectée à des milliers de cellules photo-sensibles qui en constituent le champ récepteur. Il s'y produit un effet de sommation sur un champ récepteur de plus en plus grand. L'image perçue est donc de plus en plus floue.

Le processus primaire de vision humaine présente alors des analogies intéressantes avec les effets de noyaux gaussiens : en effet, les champs réceptifs de la rétine envoient l'intensité lumineuse et ses dérivées (jusqu'à l'ordre 4) à leur cellule ganglionnaire attitrée, dont la réponse dépend de leur diamètre [Hubel88].

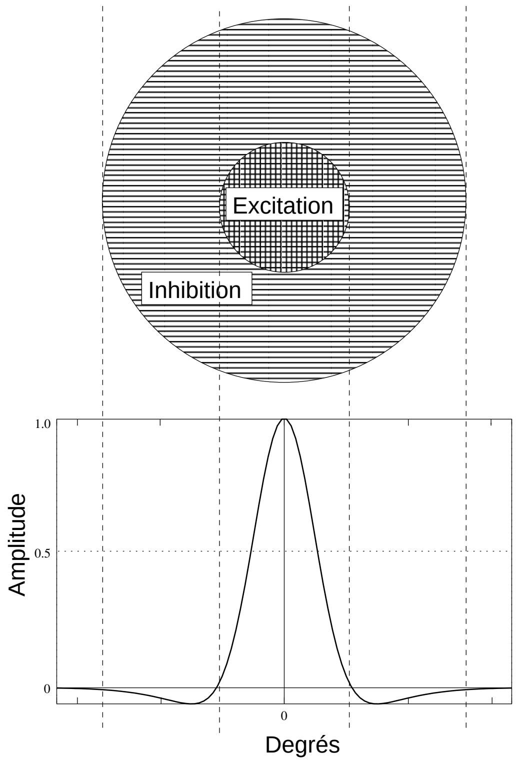

FIG. 9.5: *Réponse d'une cellule ganglionnaire on-center de la région fovéale à un très petit spot lumineux.*

Les espaces multi-échelle linéaires

Les travaux de Witkin [Witkin83], Koenderink *et al.* [Koenderink84] et Lindeberg [Lindeberg90] font référence dans ce domaine. Pour construire une représentation multi-échelles du contenu de l'image, ils ont étudié comment intégrer l'image originale au sein d'une famille $L(x, y, \sigma)$ d'images dérivées à différents niveaux de résolution σ, *l'espace des échelles.*
Le paramètre d'échelle σ est défini comme une quantité scalaire qui correspond à l'échelle d'observation de l'image. Cet ensemble est régi par l'équation linéaire de conduction de la chaleur (ou de diffusion isotrope). Le fait d'augmenter ce paramètre d'échelle correspond à l'uniformisation par diffusion de chaleur, jusqu'à température constante sur toute l'image. Les images de cette série sont obtenues par convolution de l'image originale avec des noyaux gaussiens de taille proportionnelle à σ. En voici l'expression dans le cas de signaux continus à une dimension :
soient un signal $f : \mathbb{R} \to \mathbb{R}$ et une fonction $L : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}$, la famille est définie par :

$$\begin{cases} L(x, 0) &= f(x) \\ L(x, \sigma) &= \displaystyle\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma}\, e^{\frac{-\xi^2}{2\sigma^2}}\, f(x - \xi)\, d\xi \quad \forall \sigma \neq 0 \; . \end{cases} \tag{9.1}$$

De façon équivalente, cette famille peut être déterminée comme solution de l'équation de diffusion :

$$\frac{\partial L}{\partial t} = \frac{\partial^2 L}{\partial x^2} \tag{9.2}$$

avec $t = \sigma^2/2$ et sous la condition initiale $L(x, 0) = f(x)$. C'est-à-dire que la gaussienne est la fonction de Green de l'équation de diffusion isotrope (voir par exemple [ter Haar R96]). Cette famille de solutions possède plusieurs propriétés intéressantes :

- linéarité : permet la superposition de signaux en entrée ;
- isotropie : pas de direction privilégiée ;
- invariance par translation : pas de position privilégiée ;
- invariance de l'échelle : pas d'échelle privilégiée ;
- quand le paramètre d'échelle t augmente (l'image est rendue plus floue), aucun extremum ou passage par zéro supplémentaire ne peut apparaître ;
- causalité : $L(x, \sigma_2)$ dépend exclusivement de $L(x, \sigma_1)$ pour $\sigma_2 > \sigma_1$. Cette contrainte interdit la création de détails qui n'existent pas réellement dans l'image.

Babaud *et al.* [Babaud86] ont d'ailleurs prouvé que la fonction gaussienne était le seul noyau satisfaisant toutes ces conditions, qui garantisse une destruction monotone des détails par filtrage répété de l'image.
On peut aussi montrer [Florack92] que le paramètre d'échelle doit suivre une progression exponentielle pour satisfaire le critère d'invariance, ce qui correspond à la notion intuitive que toutes les échelles jouent le même rôle et que c'est le rapport entre les tailles des différents objets qui compte, plutôt que leurs dimensions absolues.
Bien qu'en principe un seul noyau gaussien à un paramètre suffise pour construire un espace des échelles, cela ne permet pas de décrire complètement la structure de l'image. En fait, les dérivées partielles de ce noyau sont elles aussi solutions de l'équation linéaire de diffusion en deux dimensions [Florack92, Llacer93, ter Haar R96] :

$$\frac{\partial L}{\partial s} = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} \tag{9.3}$$

avec $s = \sigma^2/2$. Ces dérivées partielles s'expriment selon :

$$L_{l,m}(x, y, \sigma) = \frac{\partial^l L(x, y, \sigma)}{\partial x^m \, \partial y^{l-m}}. \tag{9.4}$$

À cause de leur similarité avec les processus primaires de la vision humaine (voir section précédente), les noyaux $L_{l,m}$ sont parfois appelés fonction de réponse visuelle [38].

À partir des dérivées partielles $L_{l,m}(x, y, \sigma)$, il est théoriquement possible de reconstuire l'image d'origine. On pourra trouver en annexe B.3 (p. 189 et suivantes) une étude de cette reconstruction qui montre que l'information contenue dans les dérivées jusqu'à l'ordre 4 est utile à cette reconstruction.

La pyramide choisie

Les deux paragraphes précédents ont montré une certaine analogie entre l'information utilisée par la rétine et celle traitée par les pyramides multi-échelles. Cela est d'autant plus vrai quand on s'intéresse à la dynamique de la formation de l'image dans le cerveau. Au premier coup d'œil, seules les structures les plus marquantes sont détectées, le reste restant relativement flou. Ce n'est que petit à petit, par accomodation et attention que les détails les plus fins peuvent être remarqués. Il y a deux phénomèmes différents (le déplacement de la fovéa, la zone la plus sensible, et l'accomodation) qui aboutissent au même résultat : une « découverte » de l'image.

Une autre similarité concerne l'ordre de dérivation utile. Dans les deux cas, il semble que l'ordre 4 soit la limite pour avoir un bon compromis entre précision et contenu de l'information.

Cette représentation est de relativement haut niveau puisqu'elle utilise de l'information située relativement loin du point courant (et ce, d'autant plus que le σ utilisé dans le filtre passe-bas est grand).

L'espace multi-échelle permet donc de représenter l'information contenue dans une image. En plus de l'image originale L^0, on construit une pyramide de K images de plus en plus floues. Le critère d'invariance impose une progression exponentielle du paramètre d'échelle et on a choisi de les définir selon $\sigma_k = 1,6 * \sigma_{k-1}$ pour $k \in [2, K]$ et avec $\sigma_1 = 2,5$. Cette progression exponentielle est également représentée dans la figure 9.6.

On définit alors

$$L^k = L(x, y, \sigma_k) \tag{9.5}$$

et

$$L^k_{l,m} = \frac{\partial^l}{\partial x^m \, \partial y^{l-m}} \; L^k(x, y, \sigma_k). \tag{9.6}$$

De façon classique et par souci de clarté, on définit également $U^k_x = L^k_{1,1}$, $U^k_y = L^k_{1,0}$, $U^k_{xx} = L^k_{2,2}$, $U^k_{xxy} = U^k_{xyx} = U^k_{yxx} = L^k_{3,2}$, etc...

Les calculs de dérivées sont effectués de façon récursive selon :

$$\begin{cases} L^k_{l,m} & = \frac{\partial}{\partial y} \, L^k_{l-1,m} \quad \text{pour } m = 0 \ldots l-1 \\ L^k_{l,l} & = \frac{\partial}{\partial x} \, L^k_{l-1,l-1} \; . \end{cases} \tag{9.7}$$

(38). Cette formulation souffre tout de même d'un inconvénient de taille par rapport au système visuel humain : le filtrage par un noyau gaussien de grande taille rendra flous les bords d'un objet, alors que l'œil les garde parfaitement nets [Llacer93].

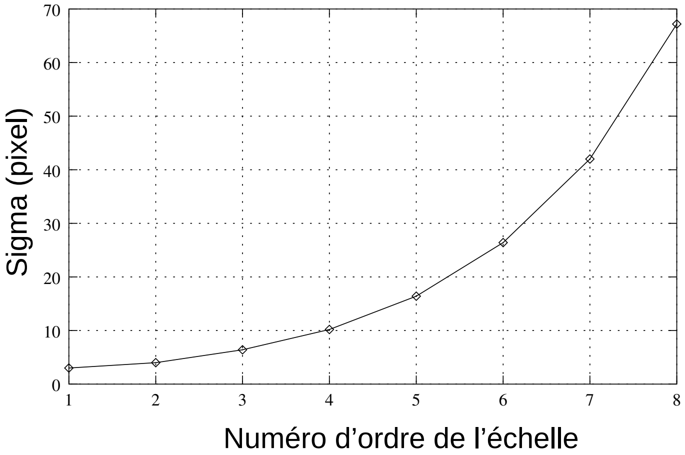

FIG. 9.6: *Progression exponentielle du paramètre d'échelle de la pyramide de filtre utilisée.*

Cependant, afin de réduire le temps de calcul, deux sous-échantillonnages simples (réduction d'un facteur 2 des tailles en x et en y) sont effectués successivement, entre $k = 1$ et 2 et entre $k = 4$ et 5 (39). D'autre part, la première image est en fait filtrée par une gaussienne de $\sigma_1 = 3$ au lieu de $\sigma_1 = 2,5$, ce qui permet de lisser un petit peu plus l'image d'origine afin d'avoir moins de points candidats pour le réseau de neurones. On construit donc une pyramide multi-échelles, multi-résolutions (40). Pour $k = 1$, l'image L^1 a la taille de référence de 513×513 et un $\sigma_1 = 3$. Les images suivantes de la pyramide sont constituées de la façon décrite dans la figure 9.7.

(39). Ces réductions sont plus aisées à réaliser lorsque l'image est de taille impaire.
(40). Une pyramide multi-résolution est également dénommée pyramide gaussienne.

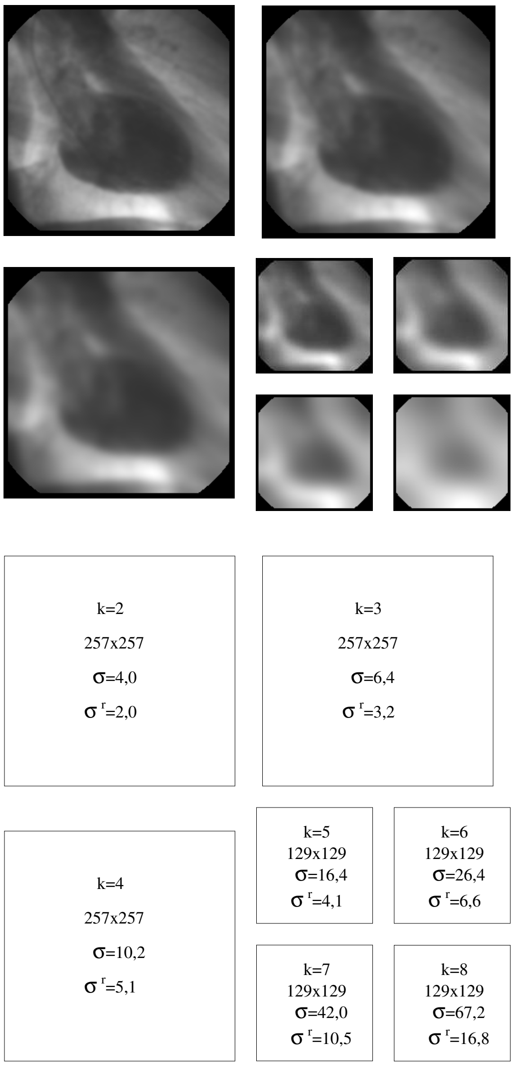

FIG. 9.7: *Pyramide d'images filtrées – taille, σ, σ adapté à leur taille.*

L'information de l'image est donc codée sur une pyramide de filtre comportant 8 résolutions, ainsi que présenté dans la figure 9.7. En outre, pour chacune d'elles, les dérivées jusqu'à l'ordre 4 sont estimées, c'est-à-dire que l'on calcule $L^k_{l,m}$ pour $k = 2, 3, ..., 8$, $l \in [1, 4]$ et $m \leqslant l$ (par (9.6)). Cette information est relativement conforme à l'information visuelle reçue au niveau des cellules ganglionnaires d'une colonne corticale. Pour la première résolution, le filtrage effectué utilise un noyau trop petit pour pouvoir calculer ces dérivées sans introduire trop de bruit. Pour $k = 1$, on ne calcule donc que les dérivées de premier ordre : U^1_x et U^1_y.

Autres informations pertinentes

Le gradient La classification des points de l'image devrait être facilitée par l'introduction d'information sur la nature des gradients locaux. Ainsi, nous avons choisi de caractériser plus finement le point courant en ajoutant de l'information sur le module ainsi que sur la direction du gradient.

Le module est calculé de la manière la plus classique :

$$G = \sqrt{U_x^2 + U_y^2}. \tag{9.8}$$

La direction du gradient est codée sur 16 bits (16 directions de gradient) de manière exclusive. Le bit codant la direction la plus proche de celle du gradient $\phi = \tan^{-1}\left(\frac{U_y}{U_x}\right)$ est mise à 1, les autres étant fixés à 0.

Ces informations de gradient sont calculées pour l'image la plus précise, c'est-à-dire la première image de la pyramide ($U = L^1$).

La courbure Une autre information intéressante pour caractériser le point courant est la courbure qui consiste en la dérivée de la direction du gradient au point considéré. Cette courbure peut également s'écrire de la façon suivante [Maintz96] :

$$\kappa = \frac{2U_xU_yU_{xy} - {U_x}^2U_{xx} - {U_y}^2U_{yy}}{\left({U_x}^2 + {U_y}^2\right)^{3/2}}. \tag{9.9}$$

Cette information devrait être surtout utile pour la discrimination des points clés (apex et extrémités des valves) puisque ce sont les seules classes dont la localisation est basée sur ce critère. À ce propos, [Griffith74] a montré que la position de l'extrémité antérieure de valve aortique pouvait être calculée à partir d'information de courbure. Pour toutes les autres classes de bord, la courbure peut être considérée comme constante compte tenu de la forme relativement linéique de celles-ci. D'un point de vue pratique, nous avons choisi de prendre $U = L^2$, c'est-à-dire de calculer la courbure sur la première image réduite de la pyramide.

Le vecteur de caractéristiques Au total, le vecteur de caractéristiques possède donc 126 éléments qui se décomposent de la manière suivante :

– 1 pour le module du gradient (voir équation (9.8)) ;

- 16 pour la direction du gradient ;
- 1 pour la courbure (voir (9.9)) ;
- 108 pour la pyramide, *i.e.* 3 pour la première résolution puis 15 pour chacune des 7 autres résolutions (voir §9.3.1).

Afin de réduire la diversité de dynamique des différentes images, les valeurs $L^k_{0,0}$ pour $k = 1, .., K-1$ (en fait les valeurs d'intensité des pixels) sont normalisées par la valeur $L^K_{0,0}$, la valeur du pixel dans l'image la plus filtrée. Ce vecteur de caractéristiques qui va donc servir de vecteur d'entrée au classifieur neuronal est calculé en certains points de l'image d'origine. Bien entendu, ce vecteur comporte de très nombreuses redondances et, après une decription des moyens mis en œuvre pour le calcul de ce vecteur, la section 9.3.3 présente les méthodes de sélection qui ont été utilisées pour déterminer les caractéristiques les plus discriminantes.

9.3.2 Calcul du vecteur de caractéristiques

Sélection des points à classifier

Le vecteur de caractéristiques doit être calculé de manière locale, pour certains pixels, dans le but de fournir de l'information à sa classification. Pour des raisons de coût de calcul, il est bien entendu impossible de classifier chacun des pixels de l'image d'origine (de taille 512×512). La première étape consiste donc en la sélection des points pour lesquels le vecteur d'entrée va être calculé et fourni au réseau de neurones. Les points dignes d'intérêt sont ceux dont le gradient est relativement important, puisque l'on est à la recherche de contours. La sélection s'effectue par calcul des gradients puis par suppression des non-maximaux, et enfin par seuillage sur la norme du gradient. La suppression des gradients non-maximaux consiste à supprimer les points pour lesquels le gradient n'est pas maximal selon sa direction, et ce, pour les points voisins, tel que décrit dans la figure 8.3, page 105. Cette procédure permet de sélectionner environ 40.000 points, soit environ 15% des points de l'image. Ce nombre est cependant encore beaucoup trop important et des procédures ont été mises en place afin de le réduire. Ces procédures sont cependant différentes pour la construction de la base d'apprentissage ou pour le fonctionnement en généralisation.

Pour la base d'apprentissage, les images ont été étiquetées manuellement en suivant le contour déterminé par le radiologue, le cas échéant ou le contour le plus probable, dans le cas contraire. Les étiquettes sont des rectangles à l'intérieur desquels les points seront tous déclarés comme appartenant à la même classe. Dans le cas des 3 points clés et de celui de classes correspondant à des zones (intérieur, extérieur, aorte), les rectangles ne sont que régions de classe uniforme alors que dans le cas de classes de contour, ils sont « centrés » sur les parties du contour à détecter. Les exemples sélectionnés vont alors être les points (de gradient maximal) dont la direction du gradient est perpendiculaire au grand axe du rectangle de façon à ne conserver que les points dont l'orientation du gradient est compatible avec le contour. Dans tous les cas, un tirage aléatoire est effectué afin de limiter le nombre maximum d'exemples de chaque classe, ce qui permet d'équilibrer le nombre d'exemples de chacune des classes.

En généralisation, le principe de sélection ne peut évidemment pas utilisé d'information sur les étiquettes. La sélection se fonde donc uniquement sur l'information de gradient. Cette sélection est très simple : elle s'effectue par seuillage du module du gradient. Les gradients trop faibles ne nous intéressent pas. Cela permet de dimimuer le nombre de points à classifier par un facteur proche de 2. Pour aller plus loin, un sous-échantillonnage de l'image d'origine est parfois utilisé. Il consiste à diviser les tailles de l'image par un facteur en puissance de 2 en ne conservant que les gradients les plus importants à chaque réduction.

Calcul des composantes du vecteur d'entrée

Pour chacun des points sélectionnés, les caractéristiques codées dans le vecteur d'entrée sont calculées. Comme plusieurs sous-échantillonnages peuvent (et sont) effectués au sein de la pyramide de filtres, il est nécessaire de recourir à l'approximation bi-linéaire pour calculer les caractéristiques liées aux images réduites. Elle consiste à estimer la valeur d'une caractéristique en un point intermédiaire à la grille éparse comme le barycentre des quatre caractéristiques aux points les plus proches P_1, P_2, P_3, et P_4, affectés des masses respectives $(1-\alpha)(1-\beta)$, $\alpha(1-\beta)$, $(1-\alpha)\beta$ et $\alpha\beta$, voir figure 9.8.

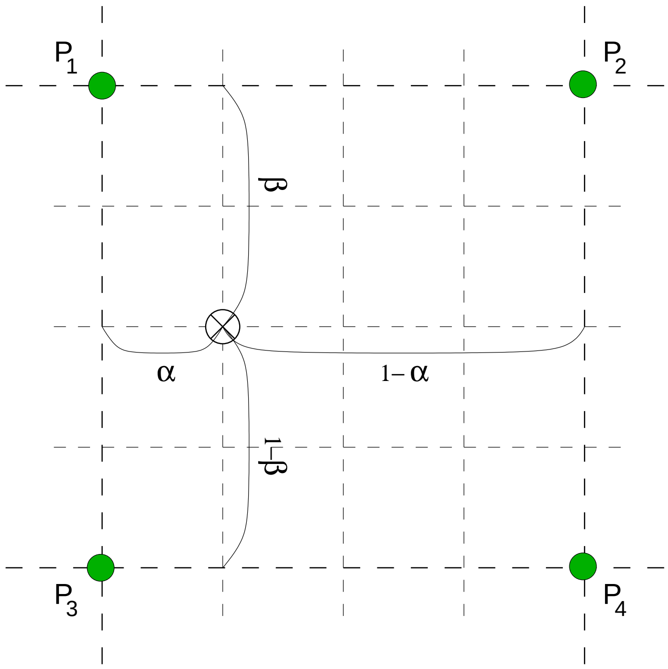

FIG. 9.8: *Principe de l'approximation bi-linéaire.*

9.3.3 Sélection de variables

L'information mutuelle

Cet espace d'entrée est de grande taille, ce qui peut poser des problèmes pour l'apprentissage, étant donné que le nombre élevé de paramètres à estimer qui requiert alors l'utilisation d'une vaste base d'apprentissage. En outre, les performances des systèmes à apprentissage peuvent augmenter lorsque les variables non pertinentes et/ou corrélées disparaissent. Il existe de nombreuses méthodes statistiques visant à sélectionner les variables les plus pertinentes d'un certain ensemble [Leray98, chap. 3]. Parmi celles-ci, on s'intéresse plus particulièrement à l'algorithme MIFS [Battiti94] (*Mutual Information based Feature Selection*). Cette méthode se base sur l'estimation de l'information mutuelle :

$$I(x,y) = \sum_{x,y} P(x,y) \ln\left(\frac{P(x,y)}{P(x)P(y)}\right). \tag{9.10}$$

Cette quantité mesure la réduction de l'incertitude sur x lorsque y est connu. C'est en fait la distance de Kullback-Leibler entre la distribution de probabilité jointe $P(x,y)$ et le produit des distributions marginales $P(x)P(y)$. Quand x et y sont statistiquement indépendants, $P(x,y) = P(x)P(y)$ et donc $I(x,y) = 0$. Plus x et y sont dépendants et plus $I(x,y)$ augmente. D'une façon générale, l'information mutuelle est le plus souvent calculée à l'aide de l'entropie H par : $I(x,y) = H(x) + H(y) - H(x,y)$ où H est l'entropie de Shannon définie par :

$$H(x) = -\sum_{x} P(x) \ln P(x). \tag{9.11}$$

Dans le cas de la sélection de variable, il s'agit de sélectionner itérativement la variable la plus informative par rapport à la tâche en cours (c'est-à-dire les variables y de sortie) et la moins corrélée par rapport aux autres variables déjà sélectionnées. Cette sélection est dite *forward* quand l'opération à l'étape p consiste à adjoindre une nouvelle variable d'entrée x_p à un ensemble existant de variables $\{x\}^{p-1}$. Cette variable est choisie selon

$$x_p = \arg\max_{x_k} I\left(\{x\}^{p-1} \cup x_k,\, y\right). \tag{9.12}$$

Les calculs peuvent être alors longs puisque l'information mutuelle doit être calculée pour toutes les combinaisons possibles (pour toutes les variables x_k possibles), et ce dans un espace de dimension croissante.

Le problème majeur de ces techniques réside dans l'estimation nécessaire des densités $P(x)$, $P(y)$ et de la densité jointe $P(x,y)$. Il semble que la méthode la plus efficace pour ce calcul soit l'estimation non paramétrique et utilisation des noyaux d'Epanechnikov [Leray98, p. 51] (voir le § 2.2 pour une description de cette méthode). Il n'en reste pas moins vrai que ce calcul est très lourd, surtout en grande dimension. Dans notre cas, l'espace d'entrée est de dimension 126 et celui de sortie de dimension 15. Ceci rend ce type d'algorithme totalement inutilisable pour la sélection.

Il est néanmoins possible d'effectuer la première étape de cette sélection *forward*, c'est-à-dire de calculer de façon indépendante l'information mutuelle pour chacune des variables x avec les sorties y. La figure 9.9 présente ces valeurs pour l'espace de sortie Y uniquement composé des 10 classes intéressantes, celles qui représentent le bord du ventricule. Ces valeurs de l'information mu-

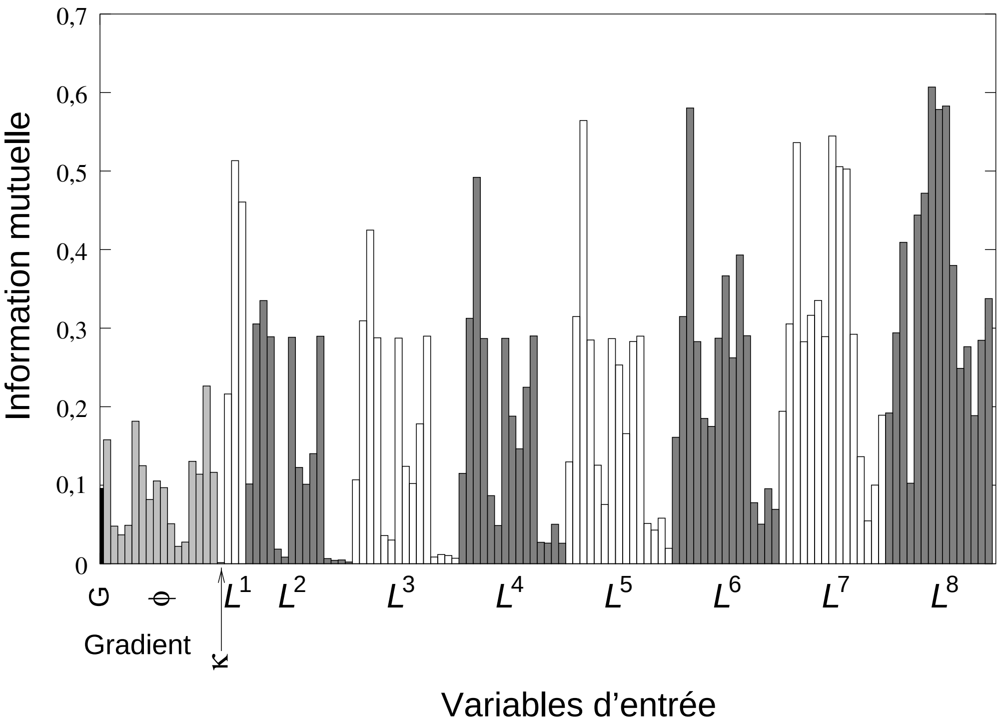

FIG. 9.9: *L'information mutuelle pour les différentes variables d'entrées.*

tuelle peuvent également être regroupées pour chaque ordre de dérivation et pour chacune des résolutions. C'est ce que présente la figure 9.10, dans laquelle les valeurs moyennes de la figure 9.9 sont calculées. Les valeurs de cette représentation doivent être considérées avec précaution puisque les moyennes ne sont pas calculées sur le même nombre d'éléments, passant de 1 à 5 pour les dérivées d'ordre 0 à 4. Plusieurs phénomènes sont remarquables à propos de l'information mutuelle :

- elle s'accroît avec k, *i.e.* avec la taille du noyau de filtrage,
- plus l'ordre de dérivation est élevé et plus cet accroissement est important.

Cette dernière remarque parait légitime puisque la taille du noyau influe directement sur la stabilité des calculs de dérivées, et ceci d'autant plus que l'ordre est grand. Calculer des dérivées à partir d'images peu filtrées revient à introduire du bruit.

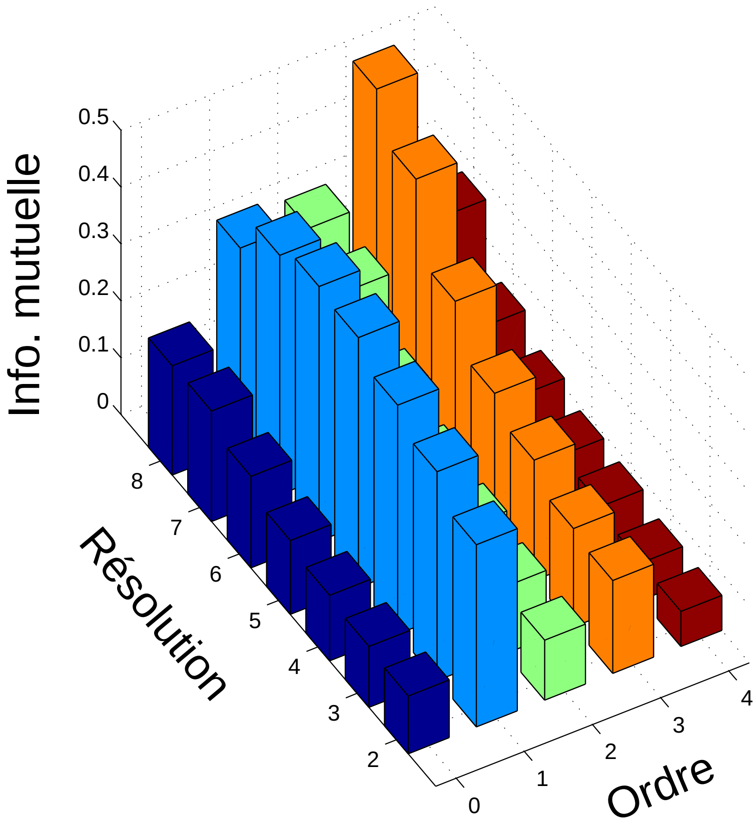

FIG. 9.10: *L'information mutuelle moyenne pour chaque ordre de dérivation et pour chacune des résolutions ; moyennes des valeurs présentées dans la figure 9.9.*

La marge

Notre problème ne se décompose pas facilement variable par variable. La sélection de variables doit plutôt s'effectuer par bloc, d'abord à partir d'une image L^k (ou des paramètres liés au gradient) et ensuite séquentiellement, en allant de plus en plus loin dans l'ordre de dérivation. En effet, les variables sont obtenues itérativement et il est ainsi inutile de supprimer une variable particulière (par exemple $L^2_{2,1} = U^2_{xy}$) si elle est de toute mamière calculée, comme étape intermédiaire, pour une autre variable (pour $L^2_{3,1} = U^2_{xyy}$, par exemple).

Le découpage en 8 blocs a été effectué de la manière suivante : le premier se compose de l'information liée au gradient et à la courbure, ainsi que de la première image de la pyramide, L^1, il comporte donc 21 entrées. Les autres blocs i correspondent aux images de la pyramide L^i, avec tous les ordres de dérivées (jusqu'à 4), donc de dimension 15. Pour chacun de ces blocs, un réseau de neurones comportant une couche cachée de 10 neurones a été entraîné sur la base d'apprentissage comportant 24638 exemples. Les performances, en pourcentage de bonne classification de chacun des blocs, sont présentées dans la table 9.2. Pour ensuite estimer la pertinence de chacun de ces classifieurs vis-à-vis de certains exemples, la marge a été calculée. La marge d'un exemple x_i de classe j^* pour un classifieur calculant la fonction f se calcule selon :

$$M(x_i, j^*) = f_{j^*}(x_i) - \max_{j \neq j^*} f_j(x_i). \quad (9.13)$$

C'est-à-dire que la marge représente la différence entre la sortie du classifieur pour la véritable sortie et son estimation sur les autres sorties. Comme les réseaux utilisés ont des sorties comprises entre -1 et 1 :

$$\forall i, j \quad -1 \leqslant f_j(x_i) \leqslant 1, \quad (9.14)$$

on a

$$\forall i, j \quad -2 \leqslant M(x_i, j^*) \leqslant 2. \quad (9.15)$$

Si la marge est positive, la classification selon la règle de Bayes donnera la bonne réponse. Et plus elle sera positive plus on pourra accorder de confiance au réseau. Pour chacun des exemples d'apprentissage x_i on cherche alors le meilleur classifieur, celui qui donne la meilleure marge. Le tableau 9.2 présente les résultats obtenus (en pourcentage de bonne classification) pour chacun des 8 blocs ainsi que le nombre d'exemples pour lesquels ils ont obtenus la meilleure marge.

Bloc	1	2	3	4	5	6	7	8
Dimension entrée	21	15	15	15	15	15	15	15
Perf. apprentissage	13,5	15,2	14,3	14,3	24,2	47,3	62,7	63,5
Perf. test	17,4	17,3	16,1	16,3	25,1	40,4	46,1	52,9
Nombre d'exemples	0	0	0	4049	1341	3015	4300	11933
Pourcentage (%)	0	0	0	16,4	5,4	12,2	17,5	48,4

TAB. 9.2: *Performances des classifieurs n'utilisant qu'une seule résolution et pourcentage de meilleure marge pour chacun de ses blocs - résolutions.*

Un test « perpendiculaire » a également été effectué. Il consiste à diviser l'espace d'entrée non plus selon les résolutions mais selon les ordres de dérivation. 5 blocs ont été définis. Le premier se compose des 18 éléments de gradient et de courbure, le second de toutes les dérivées d'ordre 0 et 1 et les 3 blocs suivants correspondent aux ordres de dérivées 2, 3 et 4, respectivement et ceci pour les résolutions 2 à $K-1$. Les résultats de l'apprentissage en parallèle de ces 5 classifieurs, chacun avec 10 neurones sur la couche cachée, sont présentés dans la table 9.3. Ce test est bien entendu biaisé vu que les classifieurs n'ont pas tous le même nombre de paramètres. Il est cependant informatif puisque les performances obtenues en n'utilisant que les informations de dérivées d'ordre 4 (le bloc 5) sont assez satisfaisantes. Ceci démontre bien la présence d'information utile à la tâche de classification, même lorsque l'ordre de dérivation est

Bloc	1	2	3	4	5
Dimension entrée	18	24	21	28	35
Perf. apprentissage	16,6	55,2	40,1	54,0	58,8
Perf. test	16,7	41,9	36,3	41,7	45,5

TAB. 9.3: *Performances des classifieurs n'utilisant qu'un seul ordre de dérivée.*

important.

Cette étude a conduit au rejet d'une partie des caractéristiques, toutes celles liées aux résolutions L^2 et L^3, puisque ces résolutions n'apparaissent pas utiles pour la tâche de classification (voir le tableau 9.2). Le nouveau vecteur de caractéristiques se compose désormais de 96 éléments. Les avantages de cette sélection ont été les suivants :

- le nombre de paramètres du réseau est réduit de façon significatif,
- le temps de calcul du vecteur de caractéristiques a lui aussi été réduit surtout que les résolutions supprimées étaient des images de grande taille,
- les performances du réseau de neurones n'ont pas été dégradées :

Dimension entrée	126	96
Perf. apprentissage	89,7	91,4
Perf. test	66,7	66,6

TAB. 9.4: *Performances avant et après sélection de variables.*

On pourrait essayer d'aller plus loin en améliorant les performances du réseau en classification par boosting ou en intégrant un terme lié à la marge dans la fonction de coût du réseau, mais cela risquerait d'empêcher le réseau de bien approximer les probabilités *a posteriori* [Bishop95].

9.3.4 Le réseau de neurones

Le réseau de neurones utilisé est un perceptron à une couche cachée. Il comporte les 96 entrées définies précédemment, 50 unités sur la couche cachée et les 15 neurones de sortie. Les bases de données utilisées comportent 20 images en apprentissage et 23 en test. Le tableau 9.5 présente la matrice de confusion obtenue sur la base de test en poucentage de bonne classification : $M_{ij} = \frac{\tilde{n}_{j|i}}{n_i}$, soit le nombre d'exemples classifiés en classe C_j alors que la classe attendue était C_i, divisé par le nombre d'éléments de la classe C_i (41). L'ordre des classes dans le tableau est celui suivi pour l'enchaînement des points. À titre d'information, les classes PA et PP sont présentées, mais elles ne seront pas prises en compte dans la constitution du contour qui débutera par la classe VA pour finir en VP (voir la figure de la p. 118).

(41). la somme horizontale ne fait pas tout à fait 100% car les résultats pour les classes de contre-exemple CA, CI, et CE ne sont pas intégrés au tableau.

		Classes obtenues										
		PA	VA	BA1	BA2	BA3	AP	BP4	BP3	BP2	BP1	VP	PP
	PA	**76**	19	2	0	0	0	1	0	0	0	0	0
Classes souhaitées	VA	11	**74**	11	0	0	0	0	0	0	0	0	0
	BA1	4	3	**73**	7	0	0	0	1	1	0	0	0
	BA2	0	0	1	**81**	1	0	0	0	0	0	0	0
	BA3	0	0	0	2	**86**	2	3	0	0	0	0	0
	AP	0	1	0	1	23	**69**	1	0	0	0	0	0
	BP4	0	0	0	0	1	5	**87**	1	1	0	0	0
	BP3	0	0	1	0	0	0	32	**43**	5	0	2	2
	BP2	0	0	0	0	0	0	4	19	**47**	11	0	0
	BP1	0	0	0	0	0	3	6	1	17	**58**	6	0
	VP	0	0	0	0	0	0	0	3	4	12	**62**	7
	PP	1	0	0	0	0	0	1	12	7	2	7	**61**

TAB. 9.5: *Matrice de confusion pour la base de test (en pourcentage des classes souhaitées).*

Un point important à remarquer est que les erreurs de classification se produisent le plus fréquemment autour de la diagonale. C'est-à-dire que quand le réseau se trompe dans sa classification, c'est presque toujours au profit d'une classe voisine. Cet état de fait est plutôt normal, car l'objectif du réseau n'est pas d'obtenir une classification parfaite mais d'approcher les probabilités *a posteriori* (voir la discussion du § 2.3.2). Or dans la tâche que l'on a donnée au réseau il est évident que les classes se recouvrent et que les limites entres elles sont très mal définies. Les « mauvaises » performances du réseau pour la classe BP2 par exemple sont donc toutes relatives, puisque cette classe est confondue avec ses voisines (BP3 et BP1).

À titre d'information, les performances de classification du réseau de neurones sur l'ensemble des classes sont de 93 % en apprentissage et 69 % en test.

9.3.5 Apprentissage et résultat

À titre d'illustration, la figure 9.11 présente les cartes de probabilités estimées par le réseau de neurones pour certaines des classes anatomiques (voir la figure 9.1) et pour une image test. Le système développé est également capable de donner comme résultat intermédiaire la position des trois points clés. Cette information est intéressante, car elle est nécessaire à l'ensemble des systèmes existants de détection semi-automatique. Dans cette version, la localisation (en utilisant les règles de cohérence décrites plus bas) des trois points clés est effectuée avec une erreur moyenne de $20,7$ pixels (pour un écart-type de $19,7$) sur 58 images. Cette localisation peut être améliorée par l'utilisation de réseaux de neurones dédiés à une seule classe et fonctionnant dans une zone restreinte autour de la première localisation trouvée. En ce sens, cette partie du travail possède le même objectif que celui poursuivi dans [Makram-Ebe93].

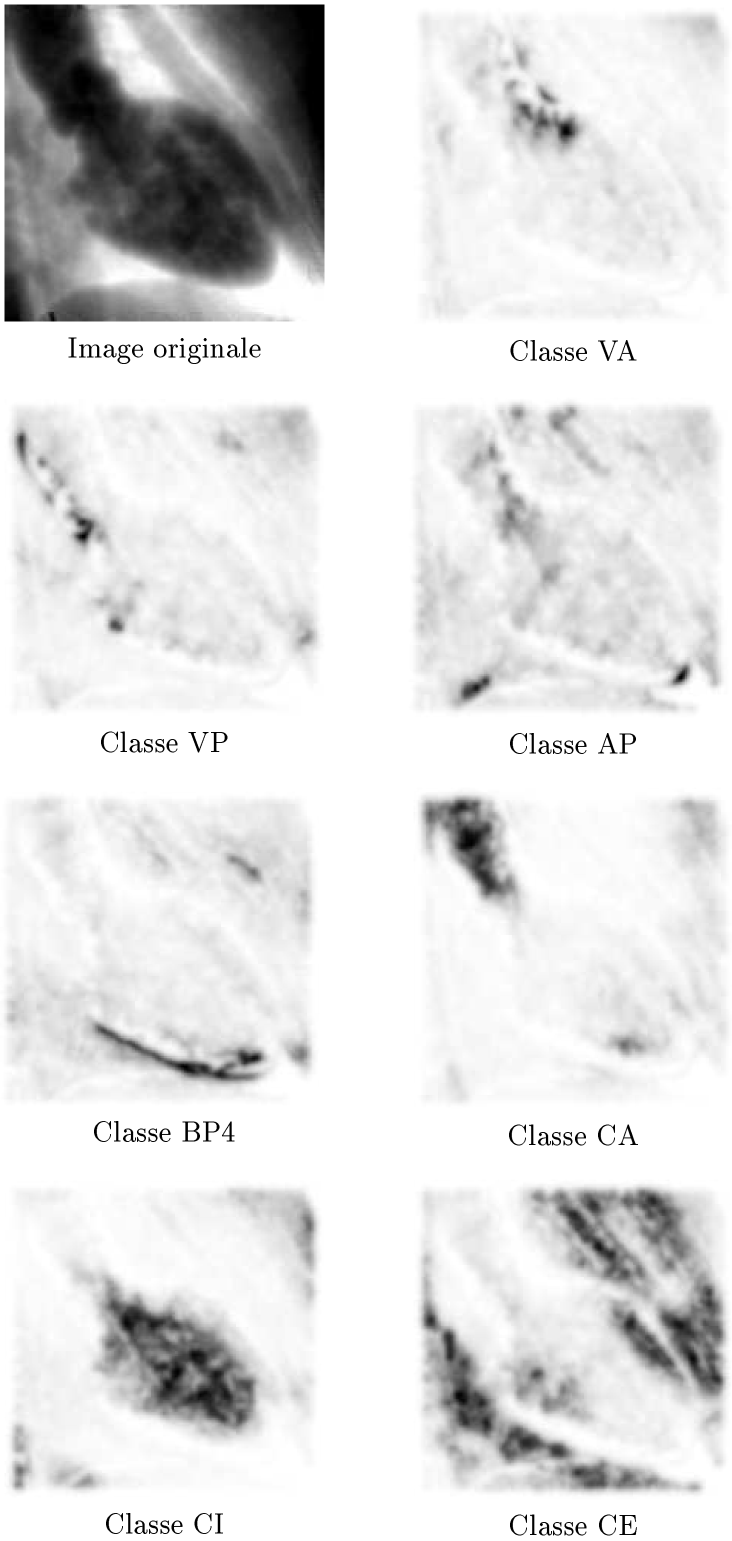

FIG. 9.11: *Exemple de résultat de classification par le réseau de neurones : cartes de probabilités où l'intensité du noir code la valeur de la probabilité.*

9.4 Extraction du contour

9.4.1 Sélection d'une ébauche de contour

À partir de la classification obtenue par le réseau de neurones, il faut maintenant construire un premier modèle de l'objet, une sorte d'ébauche. La méthode suivie est l'introduction d'information *a priori* de haut niveau sur la forme de l'objet afin de sélectionner les objets candidats puis de les trier.

Construction des ébauches plausibles

Le ventricule va être approché par un polygone reliant 5 points importants, dans l'ordre les classes *VA*, *BA2*, *AP*, *BP3* et *VP*.

Les cartes de probabilités La localisation de points importants dans l'image se base sur les probabilités *a posteriori* estimées par le réseau de neurones. Pour chacune des 5 classes citées, une carte de probabilités est construite. Cette carte est une image pour laquelle l'intensité de chaque pixel correspond à la probabilité *a posteriori* de ce pixel d'appartenir à la classe considérée. Afin de limiter le bruit, cette carte est ensuite filtrée par un noyau gaussien de petite taille. On recherche alors la position la plus probable pour la classe considérée, *i.e.* le pixel qui possède la probabilité la plus grande. La carte de probabilités est alors modifiée autour du point trouvé afin de ne pas trouver des maxima trop proches. Cette opération est répétée deux fois pour chacune des 5 cartes de probabilités. Ceci permet de construire $3^5 = 243$ polygones possibles, en reliant les 3 localisations estimées pour chacune des 5 classes à suivre.

Règles de sélection des objets candidats De l'information de haut niveau est introduite afin de sélectionner parmi les 243 polygones construits, ceux qui peuvent représenter notre objet d'étude : le ventricule. Des règles de décision prenant en compte les positions relatives des sommets d'un polygone candidat permettent de rejeter celui-ci, s'il ne peut anatomiquement pas être un ventricule. Ces règles sont schématisées par les flèches dans la figure 9.12. Ainsi la règle ① de cette figure exige que la partie antérieure de la valve aortique soit à droite de sa partie postérieure. L'application de ces règles permet de réduire le nombre de polygones-ébauches candidats à environ 40 (en moyenne et en fonction des images et des performances du réseau de neurones).

Tri des polygones-ébauches candidats

Il s'agit maintenant de trier ces candidats restants. Deux méthodes ont été utilisées. La première compare le candidat à un modèle moyen obtenu sur la base d'apprentissage alors que la seconde utilise les valeurs des probabilités comme critère de tri. L'évaluation des méthodes se fonde sur la distance entre les positions des 3 points clés (*VA*, *AP*, *VP*) détectés dans le polygone classé premier et les positions de ces mêmes points telles que déterminées par le radiologue.

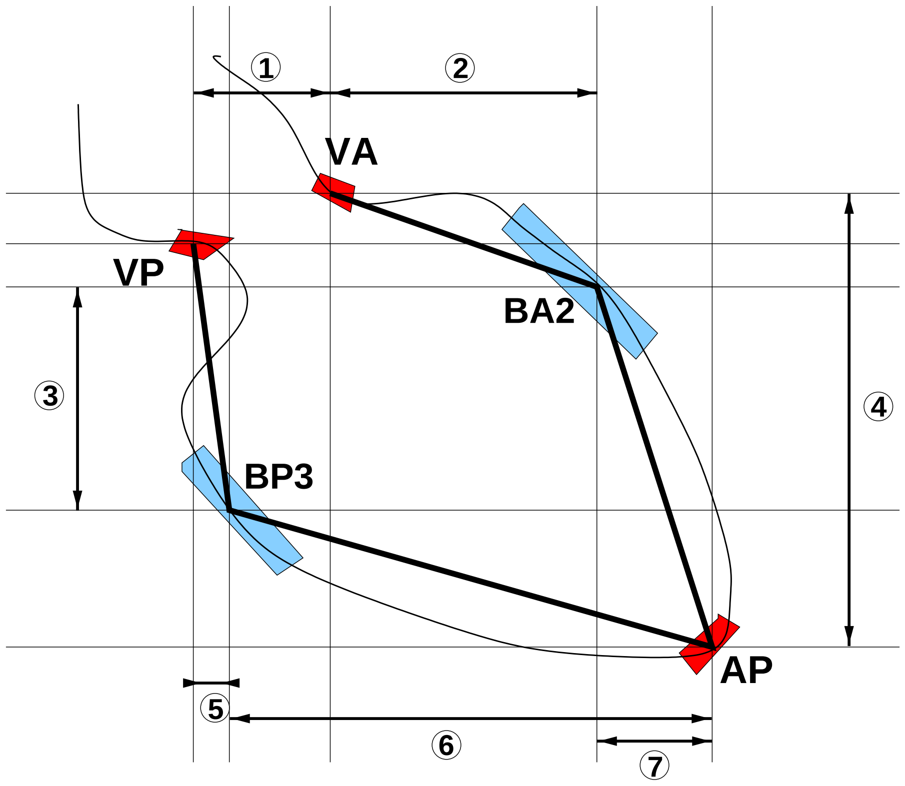

FIG. 9.12: *Description des règles de sélection des objets candidats.*

Tri par comparaison avec un modèle moyen L'idée directrice découle directement du § 9.4.1. Il s'agit d'utiliser à nouveau, et au maximum, l'information que l'on possède sur l'objet à détecter. Le tri des polygones candidats est donc effectué par rapport à une distance de Mahalanobis (voir (2.7), p. 8) au modèle moyen. Il s'agit tout d'abord de construire ce modèle moyen et d'apprécier sa variance. La construction du modèle se fonde sur l'utilisation d'une approche paramétrique d'estimation des paramètres qui pose des hypothèses fortes sur la forme des densités de probabilités *a priori*, ici des distributions gaussiennes. Plusieurs quantités informatives ont été utilisées pour la création du modèle. Elles ont ensuite été évaluées comme critère de tri.

Positions moyennes absolues des points clés Sur les 20 images d'apprentissage de la nouvelle base de données, les points clés sont situés en moyenne aux positions indiquées dans les tables 9.6 et 9.7.

	Valve antéro		Valve postéro		Apex	
Position x, y	192,3	169,2	123,4	226,7	416,8	388,9
Ecart type	38,80	22,81	39,91	25,71	28,72	37,77

TAB. 9.6: *Position moyenne des points clés en fin de diastole.*

	Valve antéro		Valve postéro		Apex	
Position x, y	219,1	199,3	153,3	245,3	407,9	357,0
Ecart type	36,34	24,26	36,65	26,30	27,86	35,83

TAB. 9.7: *Position moyenne des points clés en fin de systole.*

Cela sous l'hypothèse d'une distribution normale des positions, c'est-à-dire distribuées selon :

$$p(x) = \frac{1}{\sqrt{(2\pi)^d \ |\Sigma|}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{9.16}$$

en dimension d. Dans notre cas, les variables sont de dimension $d = 2$ et la densité de probabilité s'exprime par :

$$p(x) = \frac{1}{2\pi \sqrt{|\Sigma|}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{9.17}$$

où $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ représente la position d'un point dans le plan de l'image. Les paramètres $\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}$, matrice de covariance, et $\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$, position moyenne, sont estimés de façon standard. Cette modélisation paramétrique (voir § 2.1) est ici la plus adaptée, compte tenu du faible nombre de données. Les surfaces de probabilités constantes sont les ellipses sur lesquelles la distance de *Mahalanobis* (Δ^2, voir (2.7)) est constante. Les axes principaux de ces ellipses sont les vecteurs propres u de Σ, la variance selon ces axes est mesurée par les valeurs propres correspondantes λ [Bishop95]. La figure 9.13 présente les ellipses à un σ autour des positions moyennes pour les trois points clés. La relative finesse de localisation des points est à noter. Il semble donc que l'information de position absolue des points clés soit suffisament discriminante pour pouvoir trier de façon efficace l'ensemble des polygones candidats. Le tri s'effectue suivant la distance de Mahalanobis des positions des points clés au modèle moyen. Cette distance est calculée en utilisant la matrice de covariance complète (de dimension 6).

Positions moyennes relatives des points clés Mais les positions de points sont trop sensibles aux modalités de l'acquisition des images et une simple translation peut mettre en défaut la méthode. La même méthode de tri peut cependant être employée en utilisant une information plus relative, comme les coordonnées du vecteur reliant les deux extrémités des valves et celles du vecteur reliant le milieu des deux extrémités des valves à l'apex. Les mêmes calculs sont alors effectués en dimension 4. Les résultats de ces deux méthodes de tri par comparaison à un modèle moyen sont présentés dans la table 9.8.

Tri par probabilités *a posteriori* La seconde méthode de tri se fonde exclusivement sur l'information de classe. Les polygones candidats sont triés par

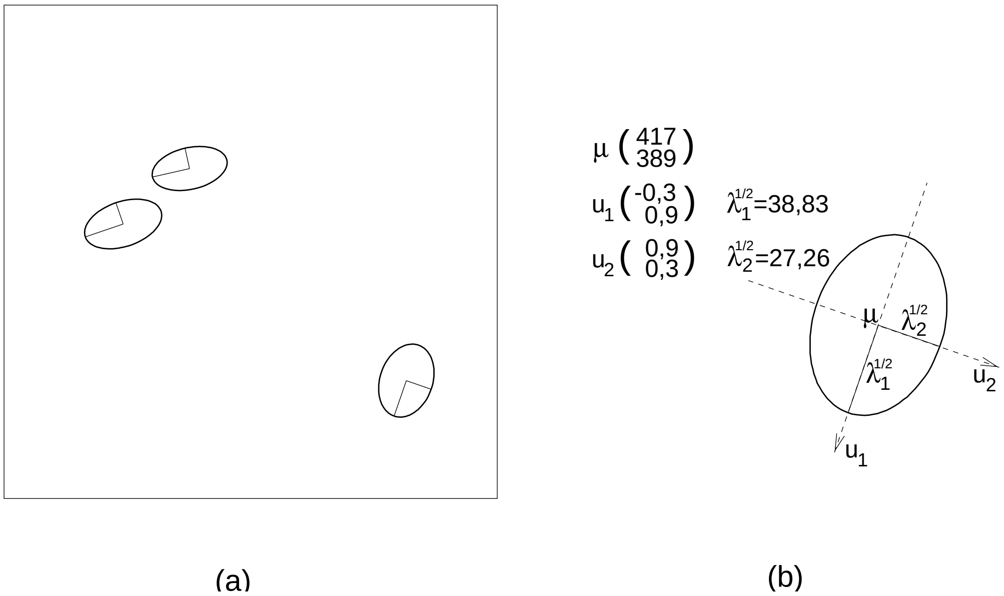

FIG. 9.13: *Position des trois points clés en télé-diastole et leur dispersion moyenne sur la base d'apprentissage comprenant 20 images : (a) position des 3 points clés, (b) détail pour l'apex.*

ordre de leur maximum successif. Le candidat classé premier est donc celui qui relie les points obtenus comme maximum global pour leur classe (si ce candidat n'a pas été rejeté par les règles de la figure 9.12). Le second va relier quatre premiers maxima avec 1 point obtenu comme second maximum, et ainsi de suite...

Cette méthode est très rapide puisque le tri est effectué en même temps que le calcul des maxima successifs. Il comporte cependant un biais envers une classe qui sera toujours la dernière à pouvoir changer d'ordre de maximum. L'implantation a été faite de façon à ce que ce biais défavorise la classe *VP*.

Mesure de la qualité des tris Les méthodes de tri ont été évaluées sur la base de test comprenant 23 images de fin de dyastole et ce, après une réduction d'un facteur 16 de la taille de l'image d'origine, le modèle moyen ayant été construit sur les 20 images de la base d'apprentissage. La table 9.8 compare les performances des différentes méthodes de tri avec le « Meilleur » choix contenu dans la liste des polygones candidats. Les performances sont mesurées sur les 3 points clés (pour lesquels les données du radiologue sont connues) du polygone candidat classé premier et ce pour les 23 images de la base de test en pleine résolution.

Ces résultats montrent que les performances des différentes méthodes de tri sont à peu près équivalentes. Comme les méthodes basées sur un modèle moyen sont à la fois plus complexes à mettre en œuvre et plus sensibles aux images sélectionnées pour l'apprentissage, le choix s'est porté sur la sélection par probabilité qui sera utilisée dans la suite. Cependant, la comparaison avec le meilleur objet

	Meilleur	Probabilité	Modèle moyen Position	Vecteurs
μ	20,4	31,0	29,4	28,4
σ	17,4	21,0	20,4	19,4

TAB. 9.8: *Résultats des méthodes de tri des polygones candidats : moyenne et écart-type des distances en pixel pour les trois points clés.*

parmi les candidats montre qu'il serait souhaitable de construire un critère de tri plus efficace puisque les performances sont très sensiblement dégradées (de $20,4$ à 31 pixels, en moyenne sur les trois points clés). Une amélioration future serait de tenir compte des deux informations à la fois : celle *a priori* codée dans un modèle et celle *a posteriori* contenue dans les probabilités. D'autres informations (angle entre les vecteurs, norme des vecteurs, surface du polygone) auraient pu être utilisées et codées par d'autres méthodes non ou semi-paramétriques telles que les réseaux de neurones.

9.4.2 Transformation causale

L'objectif général est la délinéation du contour du ventricule gauche. Pour ce faire, la méthode proposée consiste à relier entre eux (à enchaîner) des points cohérents. Cette cohérence se fonde sur deux mesures principales : la distance entre ces points et une compatibilité de classes telle qu'estimée par le réseaux de neurones. Il a été défini précédemment (§ 9.1.1, p. 116) que cette phase de liaison peut être efficacement réalisée par programmation dynamique. Cependant, cette étape nécessite la définition d'une « causalité » dans l'image, c'est-à-dire d'une manière de parcourir l'image 2D par un chemin 1D.

On peut penser à de nombreux balayages simples, comme le balayage effectué dans le tube d'un téléviseur et ses rotations, à différentes spirales, etc...
Il existe également des transformations fixes un peu plus complexes comme le parcours de Hilbert-Peano qui permet de parcourir une image 2D par une courbe, tout en respectant le plus possible la topologie. Ce parcours est construit de façon itérative et la figure 9.14 en présente les trois premières itérations.

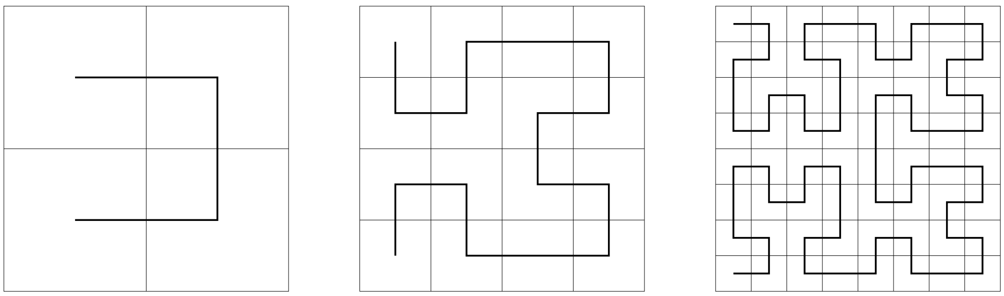

FIG. 9.14: *Construction du parcours de Hilbert-Peano.*

Compte tenu des caractéristiques de l'image utilisée et de la connaissance disponible sur la forme à détecter, tous ces parcours ne présentent pas beaucoup d'intérêt pour notre problème. Un autre type de causalité peut être introduit par transformation de l'image selon une direction bien précise. La plus utilisée (en fait la *seule* utilisée) pour ce type d'image consiste à effectuer une transformation polaire de l'image autour du « centre » du ventricule (voir la discussion sur les travaux existants en page 111) sous la forme $(x, y) \rightarrow (\theta, \rho)$. L'angle θ joue alors le rôle du « temps » et ρ parcourt l'espace des différentes possibilités à chaque instant. Cette transformation a l'avantage principal de définir une causalité cohérente avec la forme du ventricule puisque dans cette transformation, le contour du ventricule ne se replie pas et qu'il est possible (en théorie, au moins) de définir un point de contour pour chaque instant : trouver une valeur de ρ correspondant au contour pour chaque valeur de θ. Il s'agit, certes, d'une transformation très simple et très pratique mais elle repose sur deux hypothèses « dangereuses ». Tout d'abord elle suppose implicitement que la forme du ventricule est étoilée, ce qui n'est pas exact et peut même être complètement erroné pour les cas pathologiques. Enfin, elle nécessite la détermination précise ou tout du moins reproductible de façon précise du « centre » du ventricule.

Il est donc plus intéressant de spécialiser la transformation pour l'image étudiée et, maintenant que l'on dispose d'une première ébauche du contour (seulement composée de 5 points), il est sans doute plus judicieux de l'utiliser. Le principe est d'utiliser une transformation curvilinéaire ($(x, y) \mapsto (u, v)$) définie autour d'une *spline* passant par les 5 points composant l'ébauche. La figure 9.15 présente un exemple de cette transformation. Sur l'image originale en haut, l'ébauche est représentée sous la forme d'un polygone avec des traits en tirets, la *spline* par un trait continu et la zone transformée est comprise dans la bande entourant ce *spline* et délimitée par les traits épais. Cette bande, en forme du ruban élastique, est « dépliée » dans l'image du bas sur laquelle la *spline* devenu ligne centrale est représenté. Les zones noires correspondent aux points pour lesquels la transformation a échoué, soit à cause de l'absence de points (bord de l'image), soit à cause d'une ambiguïtée dans la définition du point le plus proche sur le *spline*. Cette image n'est qu'une illustration puisqu'en réalité la transformation n'est jamais appliquée aux intensités des pixels mais aux vecteurs de probabilités.

Avant d'aller plus loin, il est nécessaire de décrire brièvement la méthode utilisée pour effectuer cette transformation. Afin de construire la table de conversion de (x, y) à (u, v), il est nécessaire d'établir une carte de distance, afin de déterminer à quel point du *spline* correspond chacun des points (x, y), ce qui revient à trouver la nouvelle coordonnée u de chacun des points en prenant l'hypothèse du point à distance minimale. La coordonnée v est alors la distance signée à la ligne centrale ; le signe de v est choisi de telle façon que le repère u, v puisse localement être considéré comme un repère droit. Construire une carte de distance peut être très coûteux en temps et en ressource de calcul, c'est pourquoi nous proposons d'utiliser la morphologie mathématique pour résoudre ce problème (pour une introduction à cette méthode, se reporter à l'annexe B.4). Plus précisément, c'est le théorème de transformation de la pente qui permet de calculer rapidement la distance euclidienne entre chacun des points et son correspondant sur la ligne centrale. Pour ce faire, il suffit d'effectuer une dila-

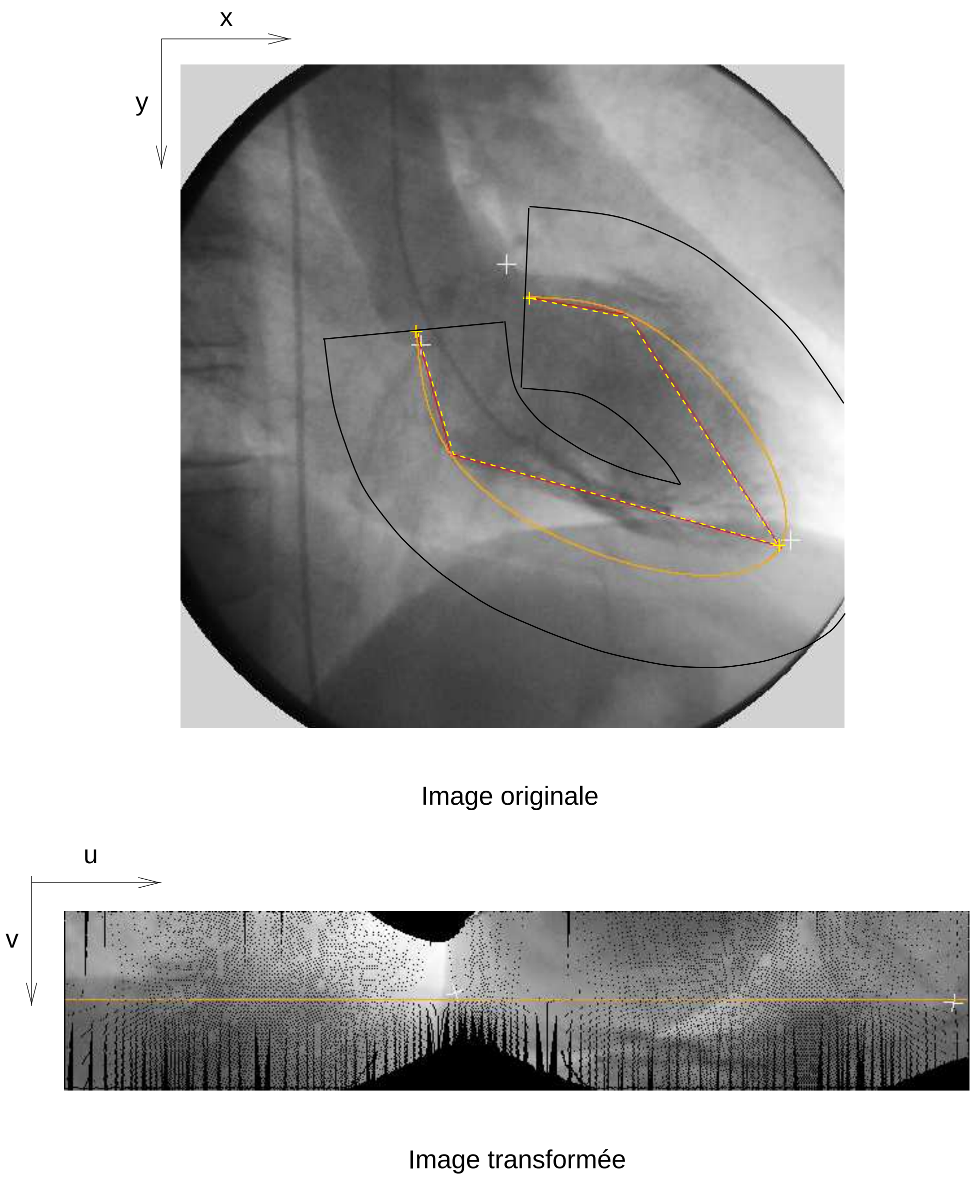

FIG. 9.15: *Exemple de la transformation curvilinéaire.*

tation de la ligne centrale par un élément structurant quadratique (qui permet de calculer la distance euclidienne) et de retenir l'origine du point transformé (ce que permet le « transport » de la pente et qui permet de construire la mise en correspondance). On peut noter que cette méthode est encore accélérée par le fait que les éléments structurants quadratiques se prêtent justement bien à l'implantation rapide et séparable décrite dans [vdB96].

Bien entendu, ce que l'on transforme, ce n'est pas l'image mais les probabilités *a posteriori* des classes telles qu'estimées par le réseau de neurones. Ce qui aboutit à une « image » 3D de dimensions U, V et N (typiquement : 500×160×15). Cependant, la transformation n'est pas tout à fait bijective, et, en particulier, plusieurs points de l'image d'origine peuvent être transformés dans les mêmes coordonnées de l'espace (u, v). En cas de conflit, l'heuristique utilisée est la suivante : le point qui sera transformé sera celui que le réseau classifie le mieux, c.-à-d. celui pour lequel il est le plus sûr de sa classification. Pour cela, il faut définir une mesure de confiance de la classification dans le cas non-supervisé (qui serait comparable à la marge utilisée précédemment dans le cas supervisé). On peut penser à plusieurs types de mesures (voir également [Leray98, chap. 2]) :

- la valeur de la probabilité de sortie maximale,
- la différence entre la sortie maximale du réseau et le deuxième maximum,
- l'inverse de l'entropie du vecteur de sortie,
- la distance de Kullback-Leibler entre le vecteur de sortie et la distribution uniforme $(\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N})$,
- la distance entre le vecteur de sortie et le vecteur de sortie idéal en non-supervisé, soit le vecteur (0,0,...,0,1,0,..,0), le 1 étant placé à l'index de la sortie maximale.

Les conditions à remplir pour une bonne mesure de confiance $\kappa(y)$ avec y le vecteur de sortie du réseau (les estimations des probabilités $P(C_k|x)$) sont les suivantes :

- $\forall y,\ \kappa(y) \in [0, 1]$,
- $\kappa(y) = 0$ quand la sortie du réseau est totalement indifférenciée y=$(\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N})$
- $\kappa(y) = 1$ quand la sortie du réseau est sous la forme y=(0,0,...,0,1,0,..,0).

Ces trois conditions sont assez faciles à remplir, mais on peut en rajouter une quatrième :
$\kappa(y^1) > \kappa(y^2)$ pour $y^1 = (y_m, \frac{1-y_m}{N-1}, ..., \frac{1-y_m}{N-1})$ et $y^2 = (y_m, 1 - y_m, 0, ..., 0)$, avec y^a_m la valeur maximale du vecteur de sortie y^a. Cette contrainte vise à privilégier les grands pics et donc à minimiser les valeurs des probabilités des classes perdantes.

Nous nous sommes attachés à tester particulièrement une mesure de confiance définie comme l'inverse de l'entropie $\kappa = -H(y) = -\sum_{i=1}^{N} \tilde{P}(C_i|x)\ \ln \tilde{P}(C_i|x)$, bien que cette mesure ne vérifie pas la quatrième condition [42]. Quatre vecteurs de sortie avec leur indice correspondant organisés de façon croissante sont ainsi présentés dans la figure 9.16.

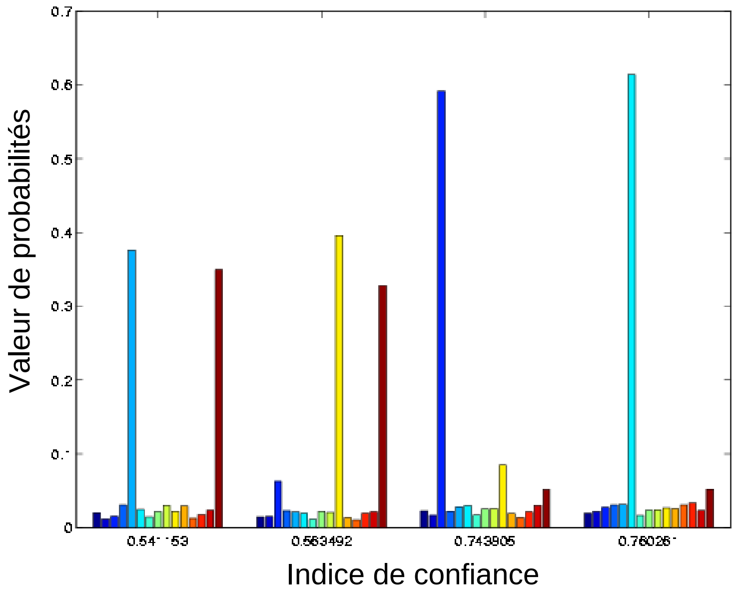

FIG. 9.16: *Mesures de confiance entropique : exemples sur quatre vecteurs de probabilités a posteriori estimées par le réseau de neurones.*

La classification hiérarchique, ici en 1D, a été utilisée pour vérifier le comportement de cet indice de confiance, comme illustré dans la figure 9.17 (avec une mesure de dissimilitude de type « saut minimum » [Saporta90, p. 254]). En sélectionnant différents niveaux pour la distance maximale, on est en mesure de créer des partitions différentes de l'espace des indices. Il s'agit ensuite de vérifier que les vecteurs de sortie qui appartiennent aux mêmes groupes sont bien de même « type ». Ainsi, les quatres vecteurs de la figure 9.16 sont deux représentants de deux classes différentes et il est bien visible qu'ils sont correctement séparés.

À la suite de ces tests « empiriques », nous avons choisi d'utiliser la mesure de confiance entropique, bien qu'elle donne le même « poids » à toute les sorties. En fait, une mesure plus utile pour notre application devrait tenir compte de la non-séparabilité des classes voisines et donc ne pas pénaliser les sorties comportant deux valeurs élevées pour des probabilités voisines (d'où le non-respect du quatrième critère).

Cette mesure est utilisée pour choisir parmi deux vecteurs candidats à la transformation au même endroit ; celui dont la confiance en classification est la plus importante l'emportera. Afin de rendre la matrice encore plus dense, des sous-échantillonnages de l'image des points à sélectionner peuvent être réalisés. Ces sous-échantillonnages sont effectués de manière à conserver les points de

(42). G. Heath, MIT, communication privée.

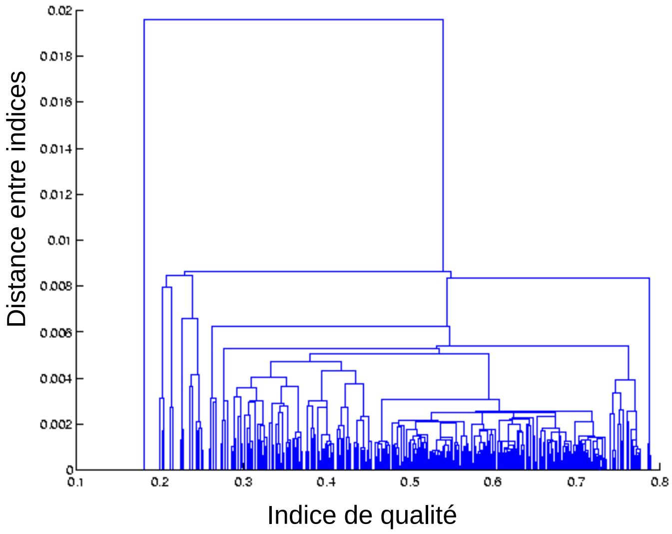

FIG. 9.17: *Classification hiérarchique de mesure de confiance.*

gradient maximum.

La fréquence de ces conflits et le taux de remplissage de la matrice (u, v) sont indiqués dans le tableau 9.9 pour une image typique, à différents niveaux de sous-échantillonnage.

Résolution	U	V	Taux de conflit	Taux de remplissage
1	735	161	4,4	16,2
2	362	81	6,0	36,4
4	186	41	12,4	62,7
8	84	21	15,5	82,7

TAB. 9.9: *Effet du sous-échantillonnage de l'image des points candidats sur le taux de conflit et de remplissage de la matrice transformée.*

Comme on le verra par la suite, un sous-échantillonnage de 4 (c'est-à-dire que seul un point, celui de gradient maximum, est conservé par imagette comprenant $4 * 4 = 16$ points) donne le meilleur résultat.

En outre, cette mesure peut être également utilisée comme valeur d'intensité dans une phase de dilatation visant à rendre l'image transformée la plus dense possible. En effet, la transformation n'étant pas inversible, l'image (u, v) comporte de très nombreux « trous ». Le premier remplissage consiste à chercher par la transformation inverse $(u, v) \mapsto (x, y)$ un point correspondant dans l'image d'origine. Ceci permet de retrouver certains des points non transformés par la transformation directe, mais ne suffit pas. Une phase de dilatation morphologique est alors utilisée pour étendre par recopie les points transformés. Cette dilatation utilise la propriété de la transformée de la pente qui permet de déterminer directement l'origine du point dilaté (voir dans l'annexe B.4, p. 191).

Cette dilatation est réalisée sur l'image (u, v) d'indice de confiance. Les sorties ayant de fortes confiances auront donc plus de propension à se « répandre » sur les positions voisines, voire même à supplanter des sorties de plus faible valeur de confiance.

Cette transformation permet de définir une causalité pour les classes qui doivent donc se ranger dans l'ordre suivant : VA, BA1, BA2, BA3, AP, BP4, BP3, BP2, BP1 et enfin VP (voir la figure de la page 118).

9.4.3 Un modèle de Markov caché en 2D

Motivations

Parmi les travaux en traitement d'images visant à l'extraction de contour, différentes formalisations ont été adoptées (voir le chapitre 8). Dans [Maître87b] un contour (ou plutôt les paramètres d'un recalage entre deux contours) est décrit par un modèle auto-régressif simple. Un formalisme bayésien est introduit dans [de Figueir92] et il permet l'utilisation de probabilités permettant de contrebalancer des valeurs de gradient « anormalemement » faibles. Enfin, [Olstad96] utilise des modèles grammaticaux pour détecter un contour décrit alors comme la succession de valeurs stationnaires de la courbure.
Nous proposons de considérer le contour du ventricule gauche comme une succession de zones anatomiques, c.-à-d. comme une séquence de segments, chacun correspondant à une classe anatomique. En outre, chaque segment est vu comme une séquence de points liés par des contraintes de continuité et de courbure. Les chaînes de Markov cachées permettent d'intégrer ces deux niveaux et ces contraintes dans un seul cadre, bien étudié, en particulier dans le domaine de la reconnaissance de la parole et en reconnaissance de l'écriture en ligne.

Ce que nous proposons, c'est de s'inspirer de ces travaux pour constituer un système hybride composé d'un HMM pour coder la séquence d'état et de réseaux de neurones pour estimer les propriétés d'émission dans chacun des états. Pour ce faire, nous avons besoin de définir un Modèle de Markov en 2D, comme décrit dans le paragraphe suivant et dans [Gérard99a] [Makram-Ebe99].

Ce que nous proposons donc c'est d'étendre le champ d'application des travaux de H. Bourlard [Bourlard94] sur les HMM discriminants pour la reconnaissance de la parole au traitement d'image, problème bi-dimensionnel. Cette formalisation de notre problème de détection de contour permet également l'introduction aisée de connaissance *a priori* de très haut niveau.

Modélisation proposée

L'image est vue comme la réalisation « bruitée » de la situation idéale d'une image ne comportant que le contour recherché comme information. L'objectif est donc de décoder l'image observée pour y retrouver l'information non observable du contour.

On se trouve donc dans une situation analogue à celle de la reconnaissance de la parole, mis à part le fait important que notre vocabulaire ne consiste qu'en un seul « mot ».

Posons les notations suivantes :

- U est la longueur de l'image transformée, c.-à-d. la longueur curviligne de la spline centrale (de l'ordre de 500),
- M est la largeur de la bande d'image transformée, c'est-à-dire le nombre de coordonnées v possibles à chaque instant u (valeur typique : 160),
- N est le nombre de classes anatomiques (15).

Le modèle de Markov proposé se compose des éléments suivants :

- les états Q sont des couples (position,classe) : $Q = (S, C)$,
- les transitions sont à la fois des transitions de positions et de classes,
- la séquence des observations est constituée de la suite d'imagettes : des petites zones de l'image d'origine centrées sur le point courant (et qui sont le réservoir pour calculer les caractéristiques utilisées par le réseau de neurones).

Le « temps » t correspond donc à la coordonnée curvilinéaire u et les positions S à l'autre coordonnée v.

Un état est en fait un couple de coordonnées et de classes. On a donc

$$Q_k = (S_{k_1}, C_{k_2}) \tag{9.18}$$

avec $k \in \mathbb{N}_{M \times N}$, $k_1 \in \mathbb{N}_M$ et $k_2 \in \mathbb{N}_N$, la relation $k \mapsto (k_1, k_2)$ de $\mathbb{N} \rightarrow \mathbb{N}^2$ étant bijective (avec la notation $\mathbb{N}_K = \mathbb{N} \cap [1, K]$). Le modèle de Markov sous-jacent est donc bi-dimensionnel. Quand il n'y aura pas d'ambiguïté possible, on utilisera de façon préférentielle la notation simplifiée $Q = (S, C)$.

Les coordonnées S sont les positions des points sur une colonne de l'image transformée (coordonnée v), alors que les classes sont représentées par C[(43)]. Les probabilités *a posteriori* des classes en chacune des coordonnées $P(C, S|X)$ sont estimées dans notre système par un réseau de neurones.

Les probabilités de transition entre états sont calculées par :

$$\begin{aligned} P(Q^t|Q^{t-1}) &= P(S^t \,|\, S^{t-1}, C^{t-1}) \; P(C^t \,|\, S^t, S^{t-1}, C^{t-1}) \\ &= P(S^t \,|\, S^{t-1}) \; P(C^t \,|\, S^t, C^{t-1}) \\ &= P(S^t \,|\, S^{t-1}) \; P(C^t \,|\, S^t) \; P(C^t \,|\, C^{t-1}). \end{aligned} \tag{9.19}$$

Ceci sous les hypothèses suivantes : indépendance de S^t et de C^{t-1} et modèle d'ordre 1 : $P(C^t|S^t, S^{t-1}, C^{t-1}) = P(C^t|S^t, C^{t-1})$.

La figure 9.18 représente les deux dimensions de notre modèle HMM. L'état central est l'état courant et seules quelques transitions sont représentées. Le modèle choisi pour les classes est de type droite-gauche avec cependant la possibilité (faible) de revenir en arrière d'une classe. En ce qui concerne les positions,

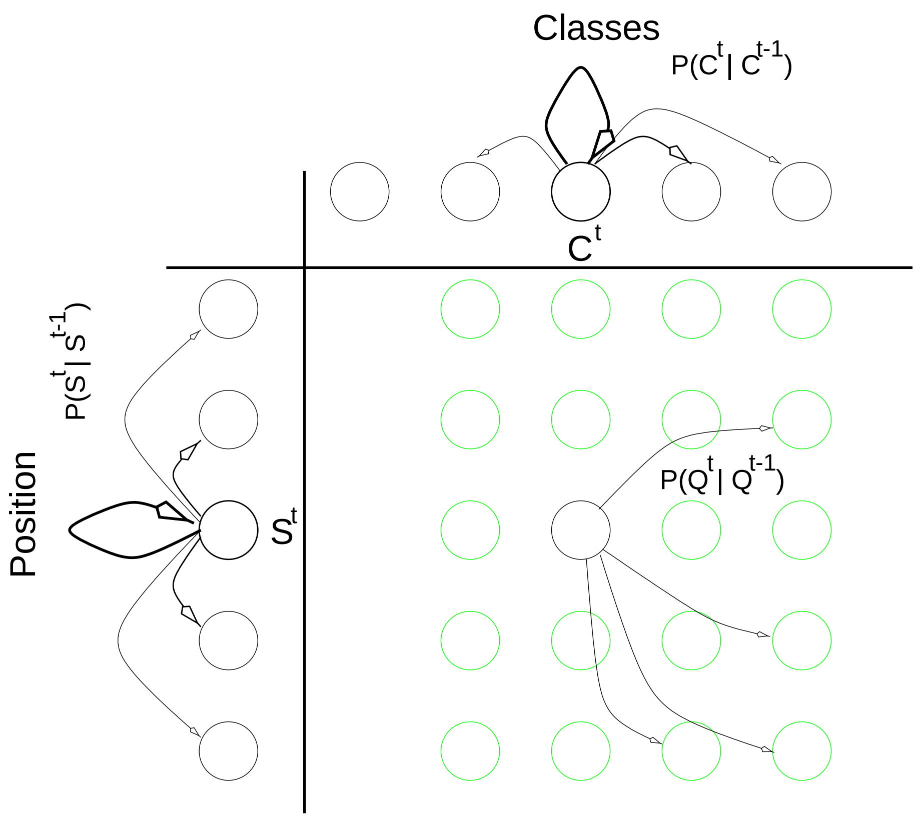

FIG. 9.18: *Le modèle de Markov proposé, en deux dimensions suivant les deux chaînes utilisées.*

la structure du modèle choisi est plus symétrique puisqu'il doit être autant possible de monter que de descendre. C'est-à-dire que les probabilités de transition de classes et celles de coordonnées sont considérées comme étant mutuellement indépendantes. Plus précisément, on considère les indépendances suivantes :

$$P(S^t \mid S^{t-1}, C^{t-1}) = P(S^t \mid S^{t-1}), \tag{9.20}$$

$$P(C^t \mid C^{t-1}, S^t, S^{t-1}) = P(C^t \mid C^{t-1})\ P(C^t|S^t). \tag{9.21}$$

Ce modèle de Markov peut s'exprimer sous la forme d'un modèle graphique, schématisé dans la figure 9.19.

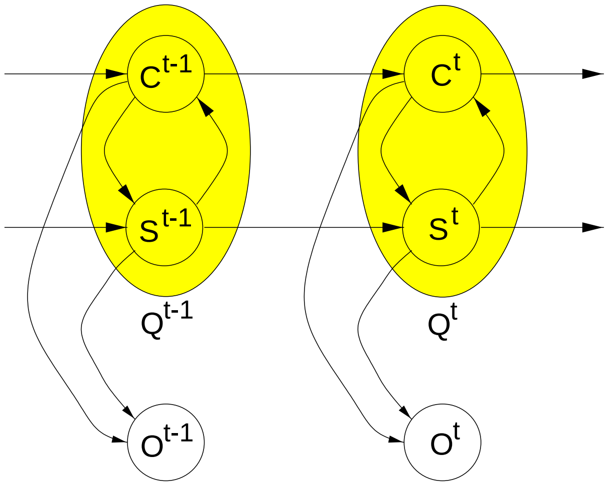

FIG. 9.19: *Le modèle de Markov proposé sous la forme d'un modèle graphique.*

Ces indépendances sont moins fortes que celles prises en hypothèses dans les « Factorial HMM » (voir la figure 3.6, p. 33), c'est-à-dire l'indépendance entre les 2 chaînes, ici S et C :

$$P(S^t|C^t) = P(S^t) \tag{9.22}$$

$$P(C^t|S^t) = P(C^t) \tag{9.23}$$

ce qui correspondrait à supprimer les relations entre S et C. En effet, ces deux chaînes ne sont pas indépendantes dans notre système et la figure 9.20 présente cet état de fait pour trois classes : CE, CI et AP. Dans cette figure, il s'agit en fait des moyennes des probabilités $P(C = x|S)$ sur l'ensemble des t (des coordonnées u du ruban de l'image transformée) et moyennées sur 5 images de la base d'apprentissage. Comme on pouvait s'y attendre, la probabilité d'être dans la classe CE (extérieur du ventricule) est plus importante pour des petites valeurs de v et inversement pour la classe CI (intérieur du ventricule). En ce qui concerne la classe AP, le pic est situé comme attendu au milieu du ruban, mais son intensité est fortement réduite par le calcul de la moyenne sur l'ensemble des positions v ; il est cependant visible.

(43). S et C sont donc en fait des *indices* de position et de classe, resp.

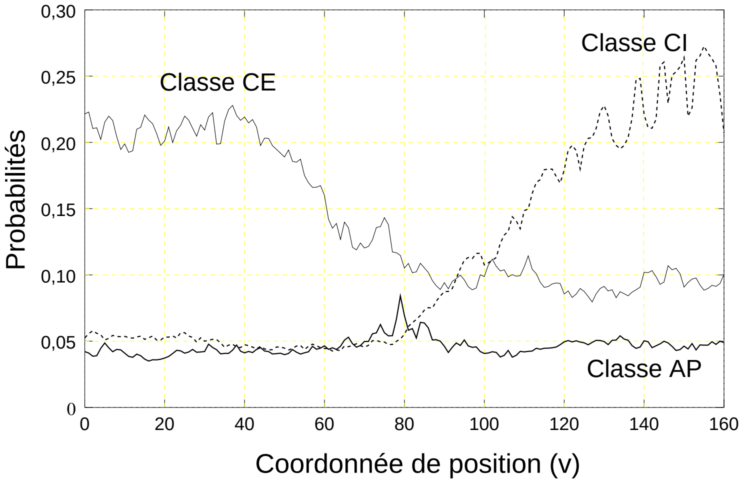

FIG. 9.20: *Les valeurs de probabilités $P(C|S)$ moyennes sur l'ensemble des t pour trois classes : CE, CI et AP.*

Un autre modèle « classique » permet les relations mais dans un sens, les « Hidden Markov Decision Tree » (HMDT, voir la figure 6.5, p. 76).

Les probabilités de transition entre coordonnées sont définies *a priori* de manière à privilégier les positions proches et ce indépendamment du temps t, selon :

$$P(S^t = S_k|S^{t-1} = S_l) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-(S_k - S_l)^2}, \tag{9.24}$$

c'est-à-dire que la distribution $p(S^t = S_k|S^{t-1} = S_l)$ suit une loi normale de moyenne S_l et de variance 1.

Les probabilités d'émission $p(O^t|Q_k^t)$ peuvent être formulées en utilisant le théorème de Bayes de la façon suivante :

$$\begin{aligned} p(O^t|Q^t) &= \frac{P(Q^t|O^t)p(O^t)}{P(Q^t)} \\ &= \frac{P(C^t \mid S^t, O^t)\ P(S^t \mid O^t)}{P(C^t \mid S^t)\ P(S^t)} p(O^t) \end{aligned} \tag{9.25}$$

avec $P(C^t|S^t, O^t)$ estimé par le réseau de neurones et $P(C^t|S^t)$ un *a priori* sur la position verticale des différentes classes. $P(S^t|O^t)$ représente un *a priori* sur la position verticale, supposée uniforme. $p(O^t)$ ne dépend pas des paramètres du modèle et il sera donc ignoré dans la phase de restauration (de décodage).

Reconnaissance

Comme nous l'avons présenté dans la partie I, le décodage s'effectue par maximisation de la probabilité d'une séquence, compte tenu de l'observation.

La séquence d'états la plus probable est alors celle obtenue par :

$$\max_{Q_1^T} p(Q_1^T | O_1^T) \tag{9.26}$$

et comme $P(Q_1^T|O_1^T) = \frac{P(Q_1^T, O_1^T)}{P(O_1^T)}$ la maximisation (9.26) est équivalente à celle de $P(Q_1^T, O_1^T)$. En utilisant les hypothèses standard dans ce type de problème on peut écrire :

$$P(Q_1^T, O_1^T) = P(Q^0) \prod_{t=1}^{T} P(O^t|Q^t)\, P(Q^t|Q^{t-1}). \tag{9.27}$$

Le produit probabilité d'émission - probabilité de transition s'exprime par

$$P(O^t|Q^t)\, P(Q^t|Q^{t-1}) = P(C^t \,|\, S^t, O^t)\ P(S^t \,|\, S^{t-1})\ P(C^t \,|\, C^{t-1}) \frac{P(S^t|O^t)p(O^t)}{P(S^t)} \tag{9.28}$$

La maximisation de (9.27) s'effectue par la minimisation itérative du coût $Z(Q_1^T) = -\log(P(Q_1^T, O_1^T))$ sur toutes les séquences d'états Q_1^T possibles. Le coût pour ajouter un nouveau point Q_k^t à l'instant t à un chemin donné Q_1^{t-1} (se terminant en l'état Q_l^{t-1}) est alors calculé comme

$$Z(Q_k^t, Q_1^{t-1}) = Z(Q_1^{t-1}) - \log P(O^t|Q_k^t) - \log P(Q_k^t|Q^{t-1}). \tag{9.29}$$

En utilisant les expressions (9.19), (9.24) et (9.25), on obtient alors l'expression qui doit être utilisée dans l'algorithme de programmation dynamique soit :

$$\begin{aligned} Z(Q_k^t, Q_1^{t-1}) = Z(Q_1^{t-1}) & \\ & - \log P(C_{k_2}, S_{k_1}|X) \\ & + \log P(C_{k_2}|S_{k_1}) + \log p(S_{k_1}) \\ & - \log P(S_{k_1}^t|S_{l_1}^{t-1}) - \log P(C_{k_2}^t|C_{l_2}^{t-1}) \\ & - \log p(O^t). \end{aligned} \tag{9.30}$$

C'est-à-dire que le coût pour ajouter l'état Q_k^t au bout de la chaîne existante est fonction (dans l'ordre suivi dans (9.30)) :

- du coût accumulé jusqu'alors ;
- de la probabilité *a posteriori* estimée par le réseau de neurones ;
- d'*a priori* sur les positions ;
- de probabilités de transition (tant en positions qu'en classes) ;
- de l'*a priori* – supposé constant – sur les données.

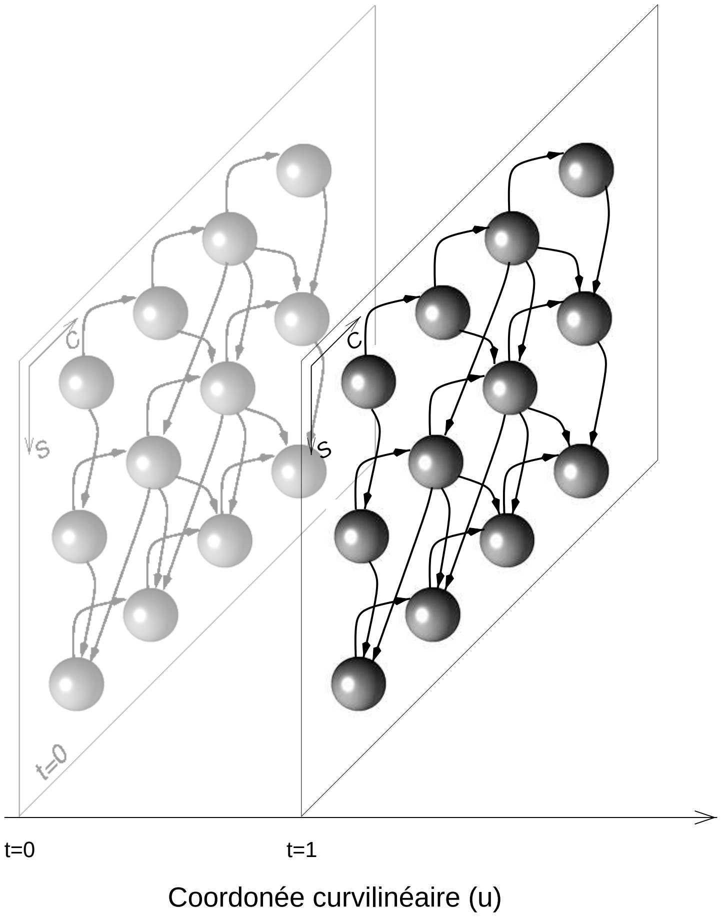

FIG. 9.21: *Le modèle de Markov proposé, présenté « déplié » dans le temps.*

Ce coût pour relier deux états est donc bien défini en deux dimensions, tel que représenté dans la figure 9.21. L'approximation de Viterbi qui consiste à remplacer la somme sur tous les chemins possibles par le calcul uniquement sur le meilleur chemin nous permet de chercher la séquence de probabilité maximale, c'est-à-dire celle de coût minimal. Grâce au principe de Bellman, le passage *forward* revient à sélectionner l'état courant de façon itérative (accumulée) en fonction du chemin passé et des caractéristiques de coût local.

On obtient ainsi la séquence d'états $Q^t = (S^t, C^t)$ la plus probable. Cependant, ce qui nous intéresse principalement, c'est la séquence S^t (la série des positions v du contour à chaque instant) qui donne le contour recherché. Ce qui revient donc à décoder une partie de la séquence d'états en posant une contrainte sur C^t, la séquence de l'autre partie des états.

La séquence C^t définit une classification des points situés sur le contour détecté. Il est bien sûr important de noter que cette classification peut différer de celle donnée localement par le réseau de neurones. En outre, on peut vérifier que la séquence des probabilités des classes telles qu'estimées par le réseau suit bien les contraintes qu'on lui a imposées. C'est ce que représente la figure 9.22 pour une image de test ; les valeurs estimées des probabilités *a posteriori* $\tilde{P}(C^t|S^t, O^t)$ sont représentées (pour les classes de bord) le long du contour détecté. Et ce que l'on observe est bien conforme à ce que l'on attendait : les classes telles qu'estimées par le réseau de neurones s'enchaînent les unes à la suite des autres.

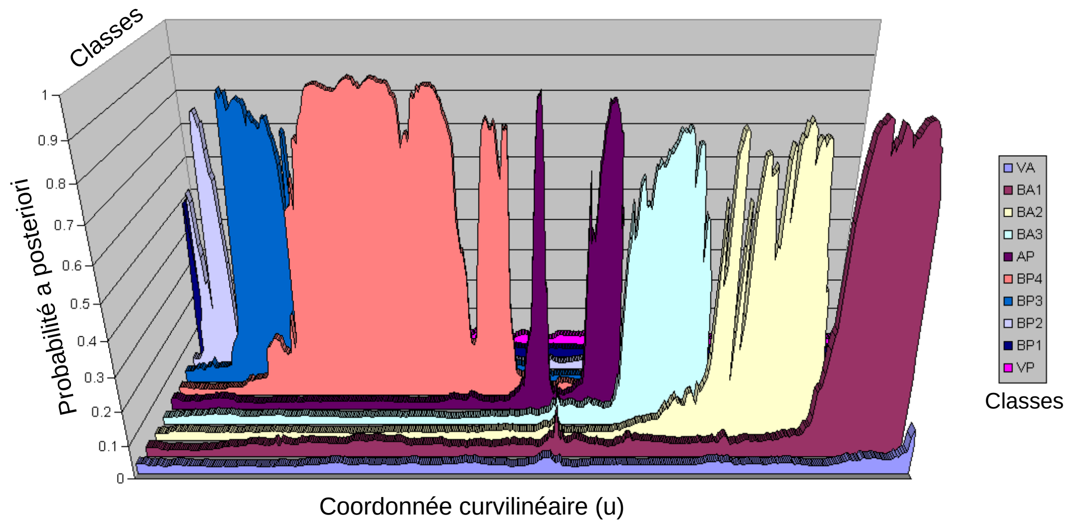

FIG. 9.22: *Les probabilités a posteriori estimées par le réseau de neurones, le long d'un contour détecté.*

Estimation des paramètres

Les probabilités d'émission sont déduites des estimations de probabilités effectuées par le réseau de neurones (voir (9.25)). Les paramètres décrivant ces probabilités sont donc ceux du réseau. La phase de reconnaissance pourrait être utilisée pour entraîner à nouveau le réseau dans les zones où il ne donne pas de très bons résultats. Ainsi, la probabilité pour la classe centrale de la figure 9.22 (qui correspond à la classe AP) le long du chemin décodé passe par une valeur très basse. C'est-à-dire que le réseau de neurones n'a pas été très performant dans cette zone. Mais, compte tenu du faible volume de données ansi ajouté, ce nouvel apprentissage n'aurait que fort peu de conséquences. Il n'est donc pas utilisé de façon automatique, mais la détection de ces zones « de mauvaise performance » a servi à aider à l'amélioration du réseau de neurones, dans le choix des caractéristiques en entrée mais aussi dans celui des exemples d'apprentissage à sélectionner.
Le calcul des probabilités d'émission nécessite cependant la prise en compte de $P(C^t|S^t)$ (voir (9.25)), c.-à-d. d'un *a priori* sur la valeur des probabilités pour les différentes valeurs de la coordonnée v. Pour certaines classes, comme CE (la classe représentant l'extérieur du ventricule) et CA (l'intérieur), la forme de cet *a priori* est relativement simple à deviner : une relativement forte (resp. fabile) probabilité pour les v petits et diminuant (resp. augmentant) pour les v plus grands.

Les probabilités de transition définies dans (9.19) et (9.24), sont principalement celles liées aux classes. Tout comme pour la reconnaissance de la parole, les classes se doivent d'évoluer dans le sens du temps, c'est-à-dire selon un schéma gauche-droite. Cependant, compte tenu des recouvrements existant entre les classes, la possibilité d'un retour en arrière d'une classe ne doit pas être écartée. Les probabilités de transition d'une classe C_1 à une classe C_2 ont été estimées par la fréquence d'occurence d'une telle transition le long des contours déterminés par les médecins et ceci pour les images de la base d'apprentissage. La probabilité de transition entre les classes $P(S^t|S^{t-1})$ est de forme gaussienne de telle sorte que la probabilité la plus forte soit celle correspondant à rester sur la même position entre les temps t et $t-1$ et qu'elle décroisse quand on s'éloigne de la position précédente.

Interprétation en termes de traitement d'image

Du point de vue du traitement d'image, notre méthode de détection de contour s'apparente à une méthode de programmation dynamique dans laquelle de l'information *a priori* de haut niveau est introduite [Gérard98]. Cette information est de *haut niveau* pour deux raisons principales :

- elle concerne un *a priori* sur la forme globale de l'objet à détecter (l'enchaînement des classes anatomiques) ;
- la description des points est composée de probabilités obtenues à partir d'une phase de classification basée sur une pyramide de filtres, donc pas

uniquement de caractéristiques très locales comme le gradient ou la courbure.

Le principe de notre extraction de contour est donc d'utiliser l'algorithme de Viterbi, mais avec l'objectif de trouver la ligne courbe de coût minimal dans un espace 3D. L'équation de décodage proposée (9.30) peut se réecrire sous la forme suivante

$$\begin{aligned} Z(v_1, c_1, v_2, c_2) = Z^*(v_1, c_1) & \quad \text{coût accumulé} \\ + \alpha\, Z(v_2, c_2) & \quad \text{coût local} \\ + \beta\, D(v_1, v_2) & \quad \text{distance de position} \\ + \gamma\, D_t(c_1, c_2) & \quad \text{« distance » de classe} \end{aligned} \tag{9.31}$$

qui représente le coût pour ajouter un point de coordonnées (v_2, c_2) à un chemin aboutissant jusqu'alors au point (v_1, c_1) (voir également la figure 9.21). C'est-à-dire que l'on recherche les valeurs de position et de classe au temps t qui minimisent le coût Z :

$$Z^*(v, c) = \min_{v_2, c_2} Z(v_1, c_1, v_2, c_2). \tag{9.32}$$

Les coûts utilisés dans cette équation sont l'opposé du logarithme des probabilités correspondantes et les termes α, β, et γ représentent les poids donnés aux différentes parties du calcul du coût (dont les valeurs sont déterminées empiriquement).

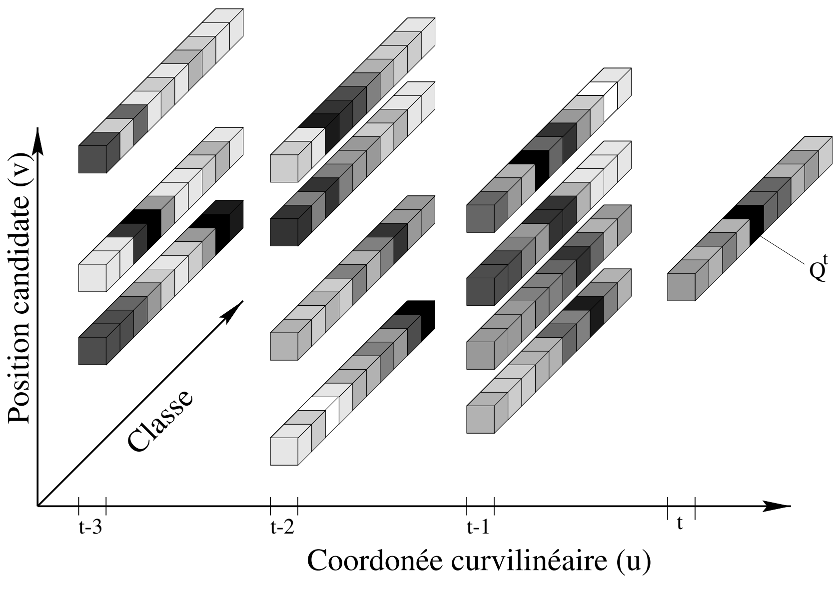

FIG. 9.23: *Le passage forward de l'algorithme de programmation dynamique utilisé.*

La figure 9.23 qui est une autre représentation de la figure 9.21, représente cette situation. À partir de l'état courant $Q^t = (S^t, C^t)$, au temps t, à la position

S^t et la classe C^t, il s'agit de trouver un état précédent dont le coût d'appariement est minimal. Cette recherche s'effectue pour les états envisageables, c.-à-d. tous les couples (position,classe). Une méthode concurrente aurait été de considérer le coût d'appariement comme une distance entre « histogrammes » de classification, les états étant alors uniquement constitués des positions v à chaque instant u.

En fait, pour s'assurer que l'algorithme de programmation dynamique ne reste pas bloqué dans un endroit très peu dense de l'image 3D, l'implantation effectuée autorise la recherche d'un point précédent (pour le passage *forward* de l'algorithme de Vitterbi) dans le temps $t-2$ mais en augmentant artificiellement le coût de tels sauts. En cas de blocage éventuel, un secteur de recherche étendu est utilisé qui autorise un saut jusqu'à $t-10$ et une variation de position également plus étendue, au prix d'un coût de distance pénalisant. Cette méthode évite le recours à l'état « refrigérateur » proposé dans [Maître87b, Maître87a].

En outre, afin de « guider » encore un peu plus la recherche par programmation dynamique, la sélection des points de départ et d'arrivée pour le passage *forward* est biaisée vers les points d'intérêt, c.-à-d. les points classifiés en VA et VP respectivement. En effet, *a priori* la programmation dynamique doit être entièrement en 2D avec toutes les positions pour $t = u = 0$ envisageables comme points de départ pour le *forward* et la recherche du chemin de coût minimum pour le *backward* doit se faire parmi toutes les positions $t = u = U$. En pratique, les coûts de départ sont modifiés afin de privilégier les points classifiés VA, la première classe de notre contour. Pour la recherche du point final, à partir duquel le passage *backward* va reconstruire le chemin de coût minimal, on ne considère pas uniquement les points de la dernière colonne, mais également ceux de quelques colonnes précédentes (en rajoutant un coût bien supérieur à celui de distance, afin de ne pas trouver de chemin tronqué).

Chapitre 10

Résultats, conclusion et discussion

Le système proposé dans le chapitre précédent pour extraire les contours du ventricule gauche dans des radiographies numériques est maintenant testé. L'absence d'une mesure objective et fiable de la qualité du contour trouvé nous a conduit à présenter les résultats sur de nombreuses images et à comparer la méthode proposée avec une méthode de programmation dynamique plus classique. L'apport de l'information de plus haut niveau y est indéniable. Cependant il reste beaucoup à faire pour poursuivre la validation du système, l'améliorer et étendre son champ d'application.

10.1 Résultats

Afin de valider l'introduction d'information de haut niveau pour l'extraction de contour, les résultats obtenus sont comparés avec une méthode de programmation dynamique plus classique qui ne se base que sur l'information locale de gradient. Pour établir cette comparaison, la même transformation causale est appliquée, c.-à-d. que les deux algorithmes travaillent exactement dans les mêmes conditions.

Dans les images de résultat présentées par la suite, les conventions graphiques suivantes sont adoptées :

- le contour dessiné par le médecin est représenté par un trait gris fin reliant les croix claires symbolisant les positions véritables des trois points clés ;
- l'ébauche, sous la forme du polygone reliant les 5 points détectés automatiquement par le réseau de neurones est indiqué par un trait alternant tirets clairs et foncés ;
- le contour obtenu par programmation dynamique « classique » n'utilisant que de l'information locale de gradient est figuré en blanc ;
- enfin, le contour extrait automatiquement en utilisant l'information de classes est représenté par un trait noir épais.

Bien entendu, les résultats présentés ont été obtenus pour des images télédiastoliques n'appartenant pas à la base d'apprentissage (c'est-à-dire au sein d'une série ne comportant que 26 images),

- les informations données par le médecin (contour et points clés) n'ont pas été utilisées,
- les contours extraits par les deux méthodes sont basés sur la même transformation : celle issue de l'ébauche.

10.1.1 Exemples de résultat

Le premier résultat présenté (figure 10.1) montre que le contour obtenu par programmation dynamique classique ne s'écarte pas assez de la ligne centrale de l'image transformée (le *spline* passant par les points du polygone). De plus ce contour semble être attiré par des points de gradient relativement fort n'appartenant pas aux limites du ventricule (en particulier pour la partie postérieure). Cette remarque est également valable pour le deuxième résultat (figure 10.2), cette fois-ci pour le coté antérieur.

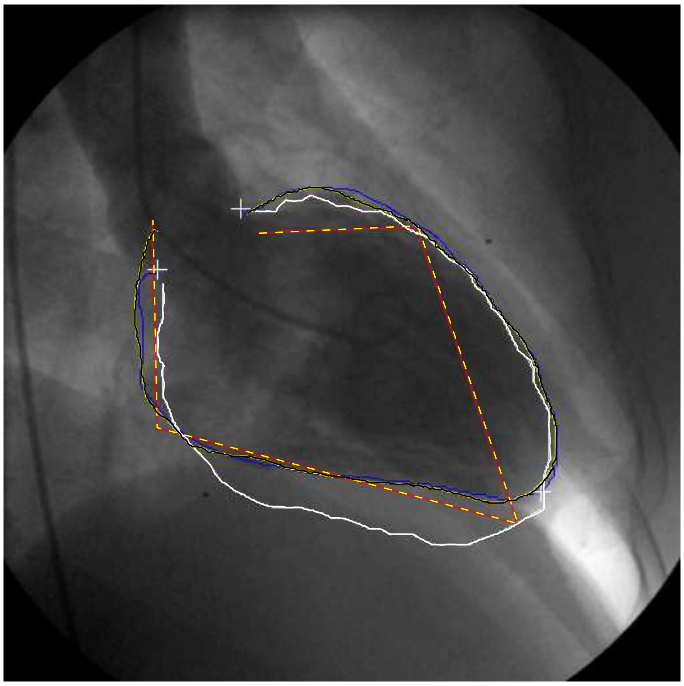

FIG. 10.1: *Résultat 1 : la programmation dynamique classique ne s'écarte pas assez de l'ébauche du contour et se trompe sur la partie postérieure.*

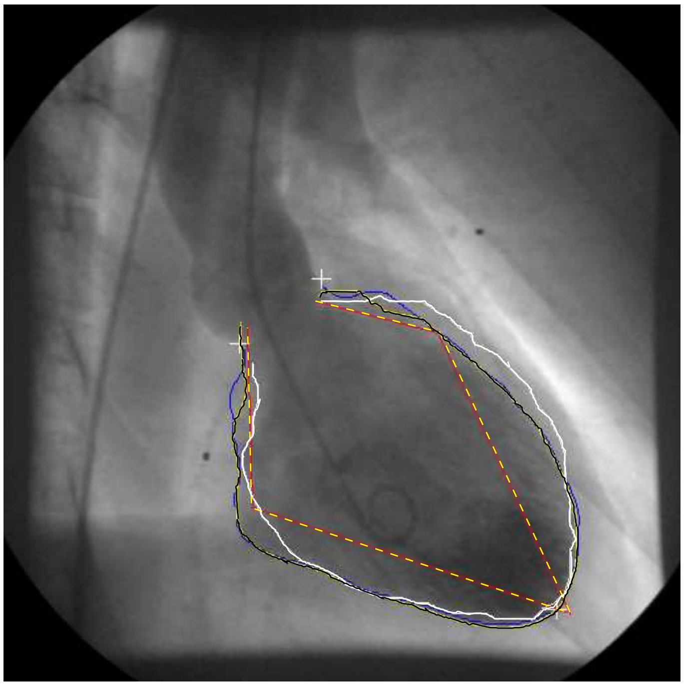

FIG. 10.2: *Résultat 2 : la programmation dynamique classique se trompe sur la partie antérieure.*

Un autre problème de la programmation dynamique dite classique est dans la force de distance qui, comme elle n'est pas contrebalancée par une autre force, a tendance à « rogner » les formes du contour, surtout en l'absence de gradient important, comme présenté pour les résultat trois et quatre (les figures 10.3 et 10.4) qui montrent que les parties postérieures et surtout apicales sont nettement raccourcies par le contour blanc.

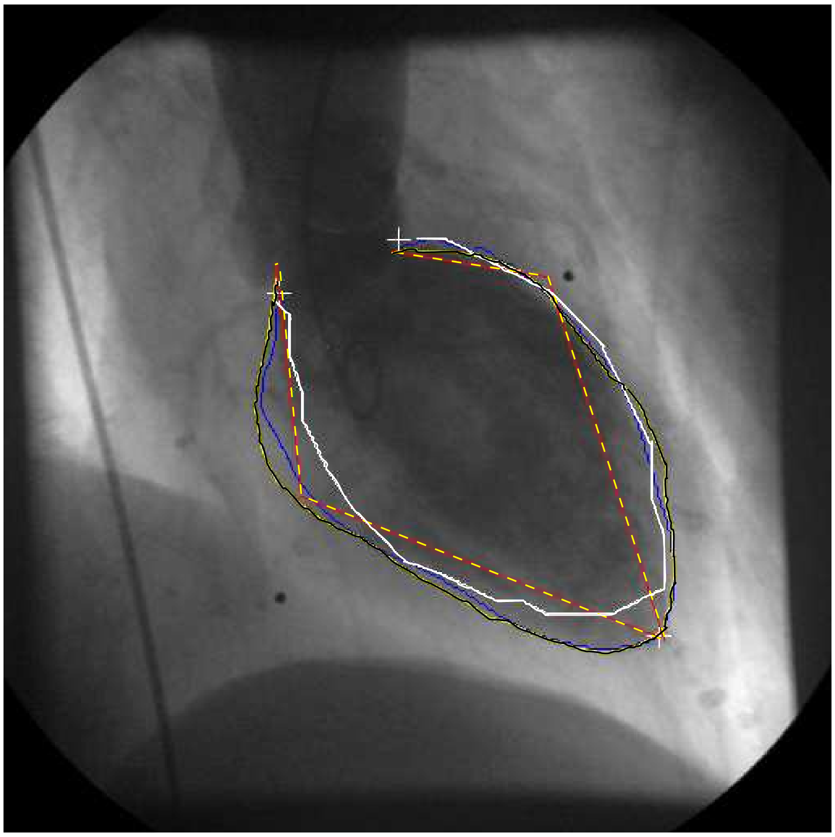

FIG. 10.3: *Résultat 3 : la programmation dynamique classique a tendance à « raccourcir » le contour.*

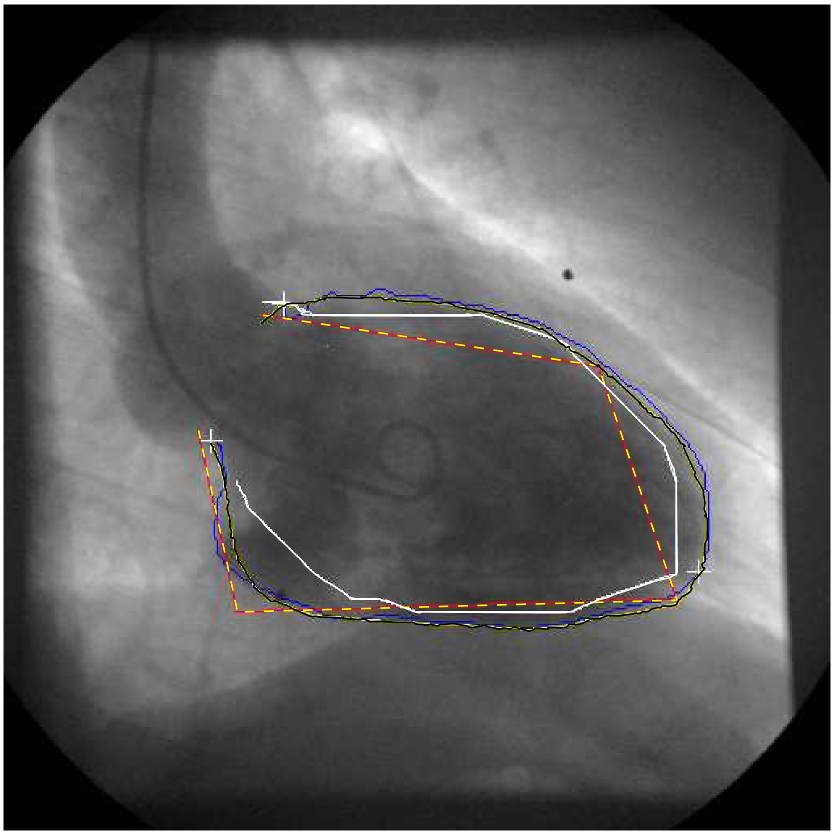

FIG. 10.4: *Résultat 4 : la programmation dynamique classique a tendance à « raccourcir » le contour.*

Pour les quatres résultats présentés jusqu'à présent, la détection automatique des points clés avait donné de très bons résultats, ce qui facilitait la détection car le contour recherché restait proche du milieu de l'image transformée. Cette situation « idéale » n'est malheureusement pas la seule à se présenter et les résulats suivants montrent les performances de l'extraction du contour dans des cas où la première phase de détection automatique des points clés ne donne pas entièrement satisfaction.

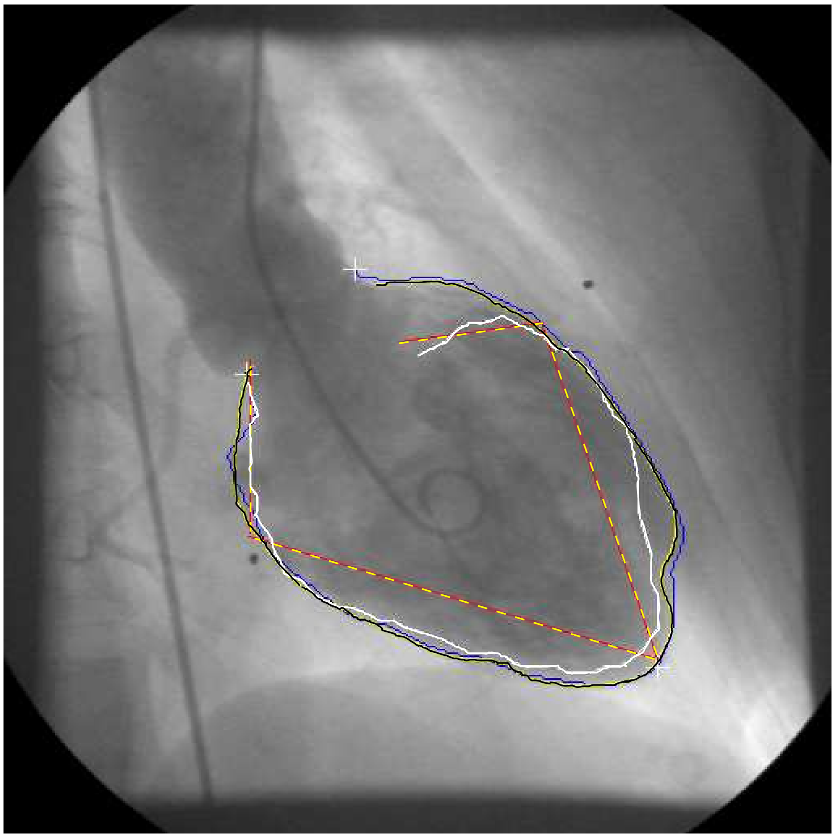

FIG. 10.5: *Résultat 5 : une erreur de détection de points clés, corrigée par l'extracteur de contour proposé, et pas par la programmation dynamique classique.*

Dans le cinquième résultat (figure 10.5), le premier point clé correspondant à la classe VA est manifestement mal détecté par le réseau de neurones. Cependant, l'information des classes permet à l'extracteur proposé de « retrouver » le contour correct, contrairement à la détection n'utilisant que l'information de gradient. Cette même situation est présentée pour le sixième résultat (figure 10.6) pour une image appartenant à une autre base de données pour laquelle le médecin n'avait indiqué que la position des points clés et pas le contour.

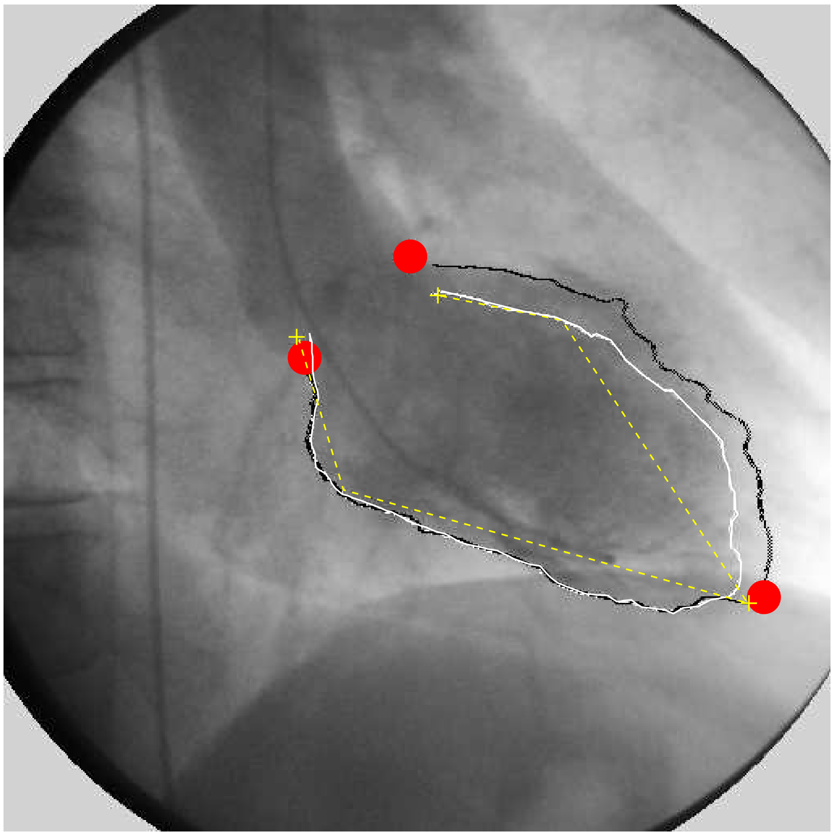

FIG. 10.6: *Résultat 6 : une erreur de détection de points clés, corrigée par l'extracteur de contour proposé, et pas par la programmation dynamique classique. Note : le médecin n'a pas dessiné le véritable contour.*

Dans certains cas, cependant, la détection des points clés « manque » les positions recherchées d'une distance trop importante pour permettre à la phase d'extraction de contour de retrouver le contour correct qui ne figure pas dans la zone d'intérêt autour de l'ébauche. Dans ce cas, l'approche proposée permet de déterminer un contour correct tant que cela demeure possible comme présenté dans le résultat n° 7 (figure 10.7). C'est un peu le même cas pour le huitième résultat (figure 10.8) obtenue pour une image présentant relativement peu de contraste.

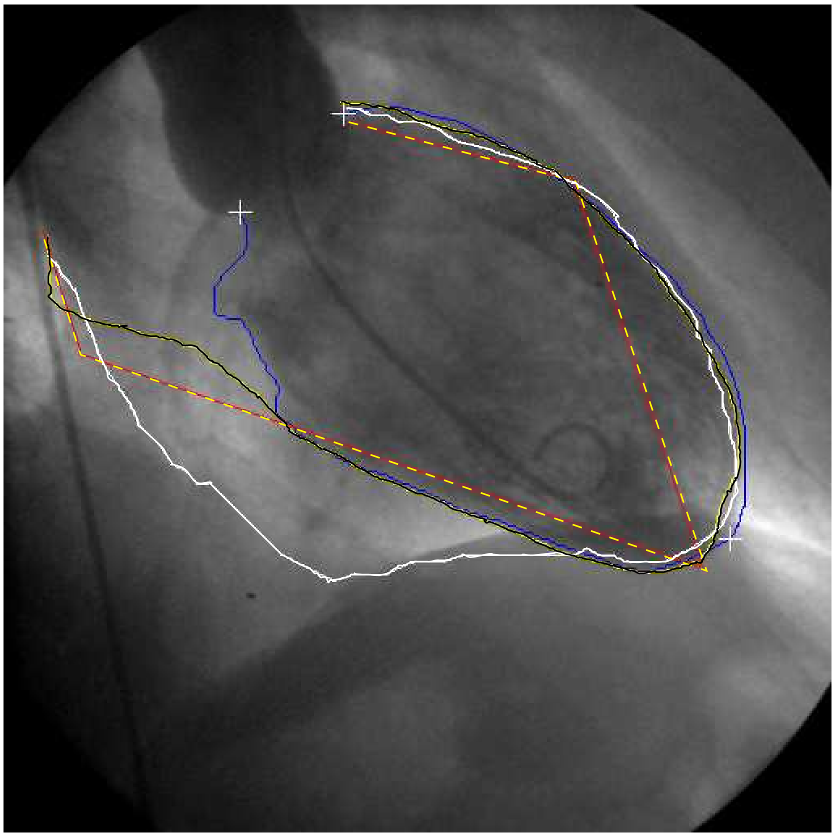

FIG. 10.7: *Résultat 7 : une grave erreur de détection d'un point clé empêche l'extracteur de bien fonctionner : le contour n'est que partiellement correct.*

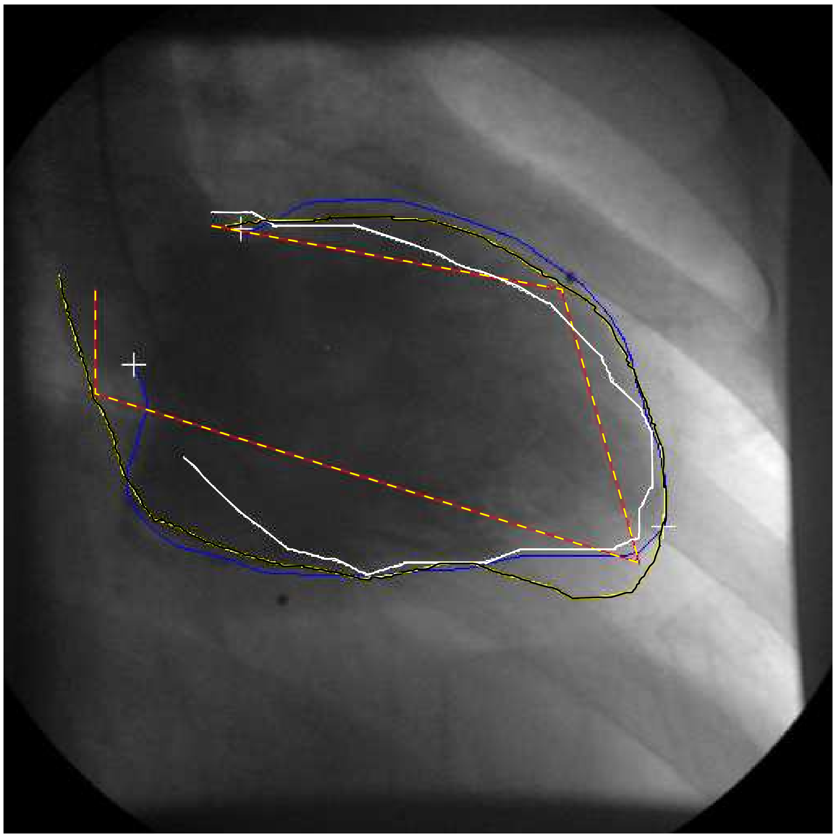

FIG. 10.8: *Résultat 8 : une erreur de détection de points clés empêche complètement l'extracteur de bien fonctionner.*

Cependant, dans le cas d'image difficile à interpréter conduisant à un mauvaise première détection, une méthode semi-automatique peut être employée. Elle consiste à demander à l'utilisateur-médecin de désigner les trois points clés de l'image : les deux extrémités de la valve aortique et l'apex (les classes VA, VP et AP représentées par des disques sur les images de résultat). Cette localisation manuelle donne les trois points de référence et les positions des deux autres composants du polygone sont établies à partir d'un modèle moyen calculé sur la base d'apprentissage. Ce protocole a été choisi afin de ne pas trop s'écarter de la pratique courante du radiologue qui est habitué à localiser les trois points clés.

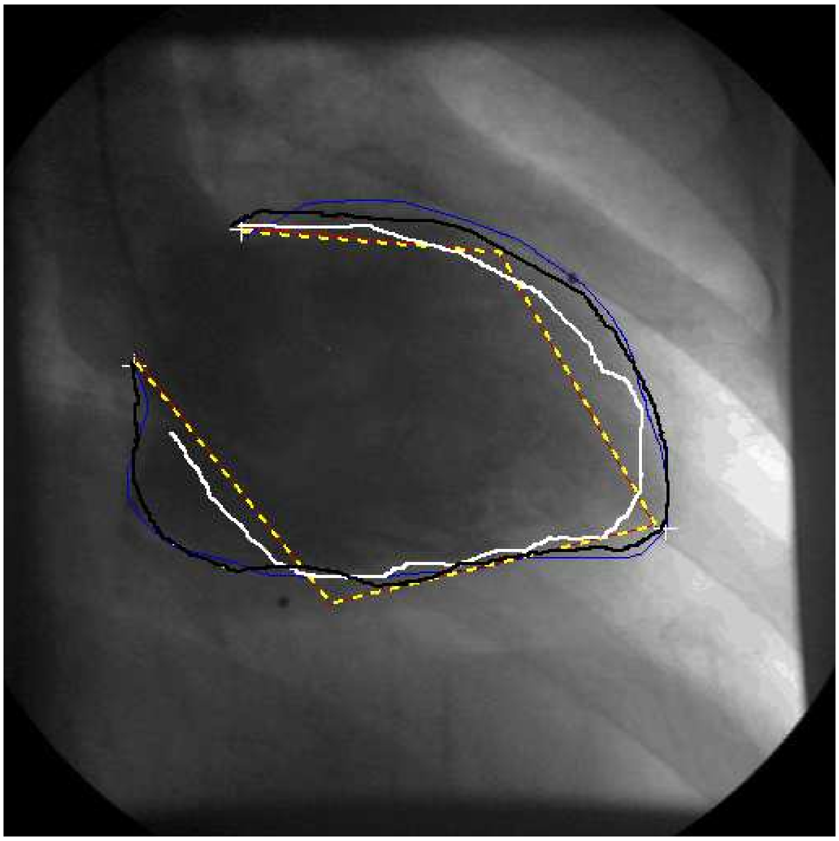

FIG. 10.9: *Résultat 9 : extraction semi-automatique du contour pour l'image précédente, résultat correct.*

Cette localisation permet donc de spécifier la transformation causale et par là, la région d'intérêt dans l'image. La méthode d'extraction de contour est alors appliquée. Cette méthode est appliquée pour la même image que celle de la figure 10.8 et donne alors de bien meilleurs résultats (fig. 10.9).

La figure 10.10 présente un autre résultat obtenu semi-automatiquement pour une image très difficile car presque sans aucun produit de contraste et comportant des éléments indésirables : des agrafes qui aident à la cicatrisation à la suite d'une intervention chiruchicale et qui apparaissent comme des morceaux de fils noirs dans l'image. L'utilisation de l'information de classe (le contour en noir) permet de déterminer un contour beaucoup plus proche de celui indiqué par le médecin (en gris fin) que le contour trouvé par la méthode de programmation dynamique plus classique, sans les informations de classes (en blanc). La détection reste cependant erronée aux abords des valves, ce qui est bien compréhensible, compte tenu de l'absence totale de produit de contraste dans ces zones. Mais elle pourrait facilement être améliorée en imposant le passage par les points déterminés par le médecin, alors qu'ici ces points n'ont servi qu'à spécifier la transformation causale.

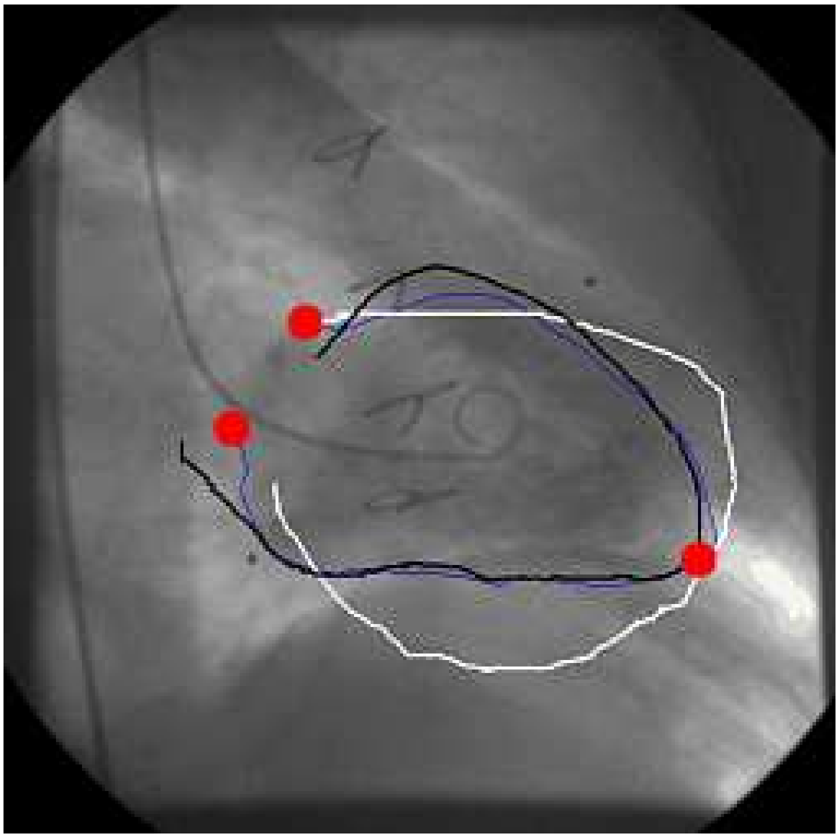

FIG. 10.10: *Résultat 10 : extraction semi-automatique sur une image difficile car étant très peu contrastée : le contour est assez correct, tout du moins relativement loin des valves, alors que la programmation dynamique classique échoue.*

10.1.2 La validation

L'analyse quantitative des résultats est rendue difficile par l'absence de méthode standard de comparaison. Dans la plupart des cas, les performances des algorithmes d'extraction de contours ne sont mesurées que de manière qualitatives, à partir d'une appréciation visuelle subjective. Les analyses objectives se basent sur la différences entre les contours détectés par la méthode d'extraction automatique et les contours tracés manuellement par un radiologiste. Les méthodes quantitatives peuvent se décomposer en deux groupes : celles qui se basent sur les résultats de mesures dérivées comme les volumes des ventricules ou la fraction d'éjection et celles qui calculent des différences entre les contours eux-mêmes. Dans l'état actuel des choses, seul le premier groupe de méthode est utilisée.(44)

Parmi les méthodes de comparaison de contours, une approche raisonnable consiste à effectuer une comparaison point à point entre les contours détectés et ceux dessinés par le médecin. Une mesure possible est la déviation moyenne en terme de pixels exprimée par la somme des différences en valeur absolue normalisée par le nombre de points [Pope85]. Une autre méthode effectue un comptage du pourcentage de points du contour détectés présents dans une bande de largeur fixe autour du contour de référence. Une méthode comparable peut facilement être utilisée dans notre cas par application de l'algorithme de F. Sheehan ([Sheehan86], et voir sa description dans l'annexe 7.4) qui permet de mettre en correspondance deux contours et ainsi de calculer une distance entre eux.

Un des problèmes importants pour avoir une mesure fiable et objective est l'obtention de contour de référence. En effet, la variabilité des contours tracés manuellement est très grande, ainsi que celle des mesures issues de ces contours : pour une détection manuelle, le volume est surestimé en moyenne de 32,8 ml (pour un volume moyen de 150 ml !) et cela dans le cas très favorable d'images parfaitement contrôlées du ventricule seul, à l'exclusion de tout autre organe [Beier96]. La seule validation envisageable est alors celle des médecins et une campagne de validation clinique doit être envisagée.

10.2 Conclusion

On a vu que face à un problème de détection de contour complexe, les outils les plus adaptés sont issus des méthodes « *top-down* » de l'approche frontière. C'est-à-dire que la recherche du contour s'effectue à partir d'un modèle souple déterminé par les connaissances possédées *a priori* sur la forme de l'objet à détecter. L'objectif de ce travail a été de poursuivre ce raisonnement et d'introduire de l'information de haut niveau dans l'ensemble des phases de l'extraction. L'information est introduite par deux moyens. Tout d'abord, les coûts de sélection des points en vue de les chaîner sont calculés à partir d'information multi-échelles. Ensuite, la recherche se doit d'être souple afin de capter les variabilités morphologiques. Mais cette souplesse implique généralement une augmentation

(44). « *Tracing variability studies generally examined the endpoint parameter, eg volume or ejection fraction. This of course provides no information on the accuracy of the border detection.* » F. Sheehan, communication privée.

importante du nombre de paramètres libres. Dans notre cas, cette augmentation est limitée par l'utilisation d'une méthode basée sur l'apprentissage. Le réseau de neurones utilisé donne en plus des résultats intermédiaires intéressants pour le médecin avec une classification selon des critères anatomiques et la détection de points clés du contour.

En outre, les temps de calcul sont raisonnables car la totalité de l'algorithme s'exécute en moins de 11 s sur une station Ultra 1 de SUN (avec un sous-échantillonnage de 4 de l'image des points candidats : ceux de gradient fort), comme présenté dans le tableau 10.1.

Station	1	2	4	8
Ultra 1	31,3	17,5	10,9	7,2
Ultra 60	13,9	8,1	5,0	3,8

TAB. 10.1: *Temps de calcul en seconde pour la totalité de l'algorithme, pour différents niveaux de sous-échantillonnage de l'image des gradients et pour deux stations de travails différentes.*

10.3 Perspectives

10.3.1 Extension du système existant

L'extraction du contour proposé se fonde sur une ébauche du contour du ventricule déterminée automatiquement par le réseau de neurones. En fait, le système actuel calcule plusieurs ébauches, les trie pour ensuite n'utiliser que celle classée première. Comme on l'a vu précédemment, cette détection n'est pas toujours bonne et peut même être si mauvaise que l'étape d'extraction qui suit ne puisse pas trouver le contour correct. L'étape de sélection des polygones pourrait être nettement améliorée par comparaison plus stricte à un modèle moyen, et surtout, compte tenu de la faible variabilité des positions des points clés (voir la figure 9.13, p. 143), cette information de position devrait être utilisée dans la phase de sélection et de tri. Mais elle pourrait également être avantageusement utilisée comme information d'entrée pour le réseau de neurones sous le forme des coordonnées (x, y) du point courant, cette information n'étant que le reflet de la pratique du radiologue qui cherche toujours à positionner le ventricule de la même manière, au centre de l'image.

En outre, il faudrait une mesure *a posteriori* du contour déterminé pour pouvoir être capable de rejeter ce contour et ensuite tester les autres ébauches construites. Ceci permettrait d'extraire plusieurs contours de la même image et des les classer par « ressemblance » avec un modèle moyen. Muni de cette mesure on serait également capable de comparer le contour extrait à celui obtenu « en remontant dans le temps », c.-à-d. au contour issu d'une phase de programmation à l'envers, dans les u décroissant (aprés une modification de la matrice de coûts de transition).

Le système actuel pourrait être nettement amélioré par l'utilisation d'une plus large base de données et surtout par l'inclusion dans la base d'apprentissage des images spécifiques au site clinique pour tenir compte à la fois des paramètres d'acquisition du système rayons-X en place et des habitudes du médecin pour la détection du contour.

L'étude d'un système semi-automatique doit être poursuivie afin dans un permier temps d'imposer le passage par les points clés déterminés par le médecin, alors que jusqu'ici ces points ne servent qu'à spécifier la « transformation causale ». Dans un futur système, il serait également souhaitable de donner au médecin la possibilité de corriger le contour obtenu, soit en indiquant des points de passage forcés, soit en redessinant tout ou partie du contour. Cette extension est facilement réalisable dans le système actuel en biaisant pour le diminuer très fortement le coût local de passage dans les points désignés par le médecin.

Un système totalement équivalent a été construit pour les images en télésystole avec un réseau de neurones dédié. Cependant, de par leur nature même, ces images possèdent un contraste très faible particulièrement dans la région apicale vu qu'elles sont obtenues à la fin de la contraction qui expulse le sang et le produit de contraste hors du ventricule. Les résultats obtenus actuellement ne sont donc pas satisfaisanst, mais pourraient être améliorés en combinant la détection statique existante avec une phase de suivi dynamique.

10.3.2 Proposition de système concurrent

Le système proposé repose sur la définition des classes anatomiques. Cependant, de par leur nature même, ces classes se recouvrent entre elles et une formalisation plus riche devrait tenir compte de cet état de fait. Une solution envisageable serait de travailler avec des classes floues, plus à même de décrire les recouvrements existants. Dans ce cas, les outils de la logique floue peuvent être utilisés, sans changement fondamental de la démarche et on peut également se servir d'un formalisme combiné : réseau de neurones - logique floue [d'Alché-92, d'Alché-94].

L'idée directrice du système proposé est de décoder l'image bruitée observée pour essayer de retrouver l'image « idéale » et non observable qui ne contient que le contour. Suivant la même modélisation, cette méthode devrait être comparée à une méthode issue du domaine de traitement de données et qui chercherait le contour comme une sorte de « moyenne » à partir « d'évidences » récollectées dans l'image. À cet effet, un test préliminaire a été effectué par B. Kégl du *Computer Science Department* de l'université Concordia de Montréal. Il s'agit de rechercher dans une image ne comportant que les points classifiés comme appartenant à une classe de bord, la courbe principale [Kègl98] ; c.-à-d. la courbe lisse qui passe par le « milieu » d'un nuage de points en dimension d [Kègl98] (qui minimise la distance à tous les points du nuage). Il s'agit d'une généralisation des composantes principales, ici sous la forme d'un polygone. Pour le test montré dans la figure 10.11, l'algorithme a été modifié afin de tenir compte de la probabilité de chacun des points de faire partie du bord (définie comme la somme des probabilités pour les classes VA, BA1, BA2, BA3, AP, BP4, BP3, BP2, BP1, VP).

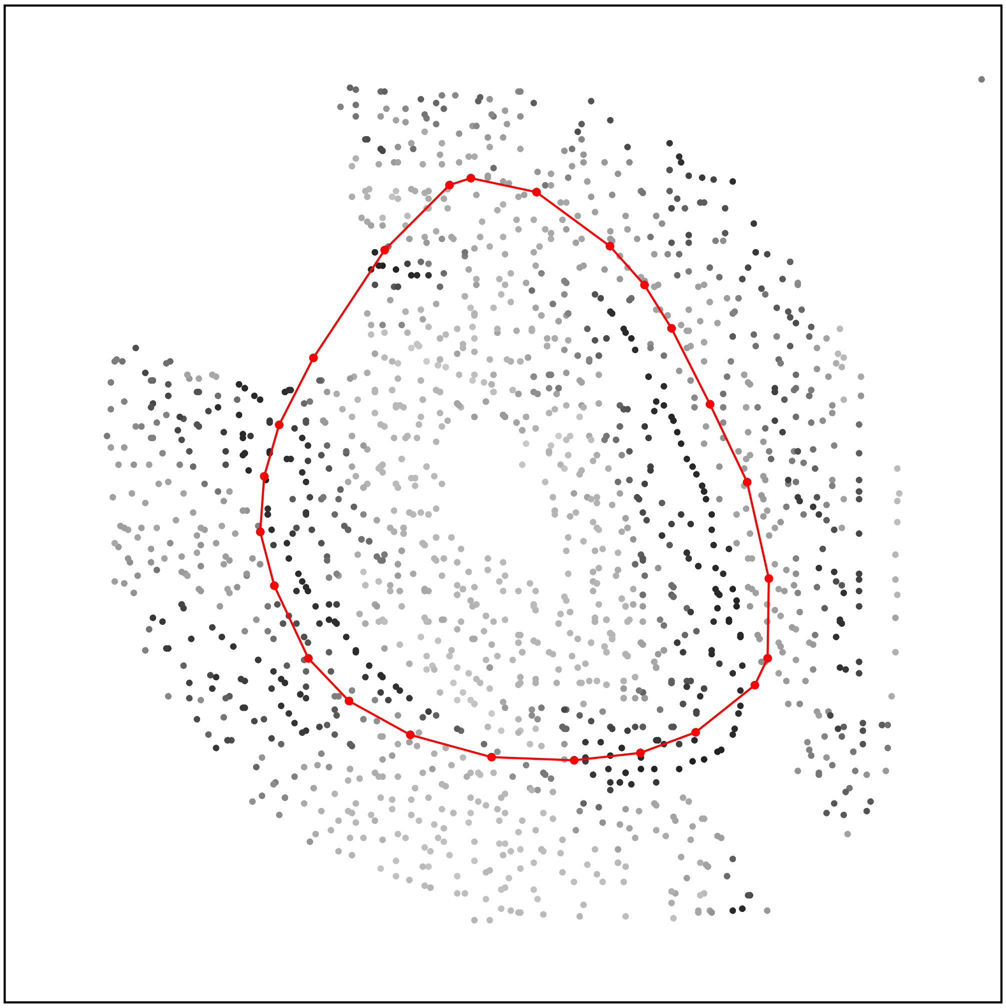

FIG. 10.11: *Le résultat d'une détection par courbe principale.*

10.3.3 Suivi dynamique

Une fois un contour extrait, il est intéressant d'être capable de le suivre au cours du cycle cardiaque afin de qualifier les mouvements des différentes parties du myocarde. La première méthode consisterait à effectuer une recherche par contour actif à partir du contour trouvé dans l'image précédente. En poursuivant l'idée générale de ce travail, une méthode plus riche serait de différencier les *a priori* sur les mouvements en fonction des morceaux du contour, c.-à-d. des classes anatomiques. Ceci permetrait d'introduire l'information existante sur la dynamique du ventricule gauche telle que : la région apicale possède un mouvement global de relèvement vers le haut et devient plus pointue vers la fin de la phase systolique, la position relative des valves n'évolue pratiquement pas au cours du cycle, ...

Pour introduire ces connaissances, il faut introduire un formalisme probabiliste de la dynamique et celui développé dans [Isard96] pour la poursuite de contours dans des séquences d'images paraît contenir la richesse suffisante à notre application.

Une autre source d'information pourrait être mise à profit : les mesures simultanées de l'électrocardiogramme qui constituent une source de renseignements sur le contexte. Il est alors intéressant de noter que l'approche générale développée dans la partie II pour la prévision de fin de décharge de batteries s'applique tout à fait et l'analogie suivante peut être développée d'un système qui serait une chaîne de Markov cachée dont les paramètres dépendent du temps et ce en fonction de variables contextuelles telles que l'ECG.

10.3.4 Applications à d'autres images

La méthode proposée est relativement générique pour pouvoir être transposée pour la détection d'autres objets dans des images de types différents. Pour effectuer cette transposition, il suffit de modifier la définition des classes, d'entraîner un classifieur à partir d'une base d'images étiquetées, et de proposer des règles statistiques de haut niveau pour générer l'ébauche et pour décrire l'enchaînement des classes.
Les contraintes de la méthode proposée : la relative lourdeur des calculs et le besoin d'une certaine cohérence dans le contexte et dans la forme (et la pose de la forme) à trouver semblent néanmoins restreindre son extension à la détection d'objets bien connus dans des images relativement bien controlées. En dehors de la détection d'organes dans des images médicales, on peut par exemple penser à des tâches de vision industrielle comme la surveillance dans le placement automatique de composants sur des plaques de circuit imprimé.

Mais la généricité du modèle s'exprime surtout dans l'intégration de connaissances *a priori* de haut niveau dans chaque étape d'un système d'extraction de contours. Tout d'abord, l'intérêt d'utiliser une programmation dynamique moins myope est démontré. C'est-à-dire que l'extracteur de contour doit être capable de prendre en compte de l'information spatiale autour du point considéré pour le chaînage. Ensuite, et surtout, ce chaînage doit être guidé par le plus possible d'information possédée *a priori* sur la forme globale du contour à détecter.

Chapitre 11

Conclusion

L'approche générale que nous avons suivie est basée sur une modélisation du phénomème physique sous-jacent. Modéliser permet de mieux comprendre les mécanismes mis en œuvre en se rapprochant des conditions réelles et permet l'introduction aisée et transparente d'information de haut niveau.

Après avoir montré que les deux approches standard pour le traitement des séquences, *ie* les filtres temporels issus du domaine de l'automatique et les chaînes de Markov cachées, n'étaient que l'expression du même type de modélisation d'un système dynamique, chacune des deux approches a été utilisée de façon originale pour résoudre deux problèmes industriels.

Dans le premier problème de prévision de décharge de batteries, l'approche proposée est fondée sur des algorithmes dotés de la capacité d'apprentissage à partir d'exemples. Plus précisément le système construit se compose de deux réseaux de neurones organisés de façon hiérarchique. Le premier réseau est un modèle simple de courbe de décharge, en attendant de pouvoir disposer d'un modèle plus explicite aux yeux des experts. Le deuxième réseau à la tâche de déterminer les paramètres du premier réseau compte tenu d'informations contextuelles sur l'état courant de la batterie. La très grande diversité dans le comportement des batteries est maîtrisée par la capacité incrémentale de ce second réseau, qui prend en charge l'adaptation en ligne du « sélectionneur » de paramètres compte tenu du comportement de la batterie en cours d'utilisation. Les résultats obtenus permettent d'avoir une précision moyenne de l'ordre de 6 minutes pour un événement pouvant intervenir dans un intervalle de temps étendu sur 10 heures.

Le système développé et les idées qui ont présidé à sa création sont très généraux et peuvent sans grande difficulté être transposés à d'autres problématiques. En effet, le problème traité est celui de la modélisation d'un processus dynamique dont les paramètres internes évoluent eux-mêmes au cours du temps. Ce type de problème est très courant dans les domaines de l'économie, de la finance, mais également pour des tâches de classification qui peuvent être rendues adaptables à des phénomènes évoluant dans le temps, comme la réponse des capteurs d'information. Il devient alors également envisageable de pouvoir s'adapter au plus versatile de tous les phénomènes : l'utilisateur lui-même. Cette voie de recherche est relativement neuve mais reçoit actuellement l'attention de très nombreux chercheurs.

Ce travail a conduit à la publication de deux articles, l'un dans une conférence internationale [Gérard97], l'autre dans une revue [Gérard99c] et au dépôt d'un brevet international [Patillon98]. Le travail complémentaire d'implantation, de validation et d'intégration au produit final est maintenant à la charge de l'entreprise américaine FLUKE, commanditaire de ce travail.

Le deuxième problème traité relève plus de la reconnaissance des formes avec la constitution d'un système de détection automatique du contour dans des images médicales. L'idée directrice est l'introduction d'information de haut niveau à toutes les étapes constitutives de l'extraction. Le modèle développé comporte deux niveaux. Le premier niveau est constitué par un classifieur multiéchelle chargé de repérer certaines parties (classes) de l'anatomie du ventricule et d'évaluer en chaque point de l'image les probabilités *a posteriori* d'appartenance à ces différentes classes. Ces probabilités servent ensuite de base à l'établissement d'un coût de chaînage des points, le contour étant vu comme une séquence de ces points, mais elles sont également utilisées pour guider la programmation dynamique vers un modèle de transition codé dans une chaîne de Markov cachée. L'introduction d'information *a priori* permet à l'extracteur de contour de mieux discerner le véritable contour dans l'image bruitée. Les premiers résultats obtenus sont prometteurs même si une validation clinique reste encore à faire.

Ce travail a donné lieu à deux présentations lors de conférences internationales [Gérard98, Gérard99a] et au dépôt d'un brevet français en cours d'extension à l'étranger [Makram-Ebe99]. Des discussions sont actuellement en cours pour intégrer ce système à la station de travail conçue par Philips Medical Systems et destinée à l'aide au diagnostic.

La suite de ce travail de thèse pourra prendre la forme de l'intégration des deux approches afin d'utiliser de l'information dynamique dans l'extracteur de contour, afin de permettre le suivi dynamique des contours du ventricule gauche dans la séquence d'images décrivant le cycle cardiaque. Là encore, de l'information de haut niveau devra être introduite sous la forme de probabilités des mouvements des différentes parties de la paroi afin de constituer un système robuste, l'idée étant que l'ajout d'information dynamique va aider la détection du contour, ce qui d'ailleurs correspond à la pratique du cardiologue qui commence par visionner la séquence cardiaque avant de dessiner les contours dans les images fixes. En outre, un système parallèle à celui mis en place pour la prévision des fins de décharge pourra être utilisé afin de rendre le système capable de s'adapter en cours d'examen aux particularités du patient.

Annexe A

L'électrochimie des batteries Nickel-Cadmium

A.1 Les principes

Le fonctionnement d'une batterie rechargeable Nickel-Cadmium (NiCd) est schématisé dans la figure A.1. On peut y distinguer les deux électrodes :

- l'électrode positive Ni constituée du couple électrochimique $NiOOH/Ni(OH)_2$;
- l'électrode négative Cd constituée du couple électrochimique $Cd/Cd(OH)_2$;

qui sont noyées dans une solution de KOH concentré. Afin d'éviter tout contact électrique entre les deux électrodes, un séparateur isolant (mais poreux) les sépare.

Les réactions électrochimiques principales qui ont lieu pendant la charge et la décharge sont les suivantes :

- dans l'électrode Nickel :

$$\mathrm{Ni(OH)_2 + OH^-} \underset{\text{décharge}}{\overset{\text{charge}}{\rightleftharpoons}} \mathrm{NiOOH + H_2O + e^-} \quad \text{(-0,52 V)} \qquad \text{(A.1)}$$

- dans l'électrode de cadmium :

$$\mathrm{Cd(OH)_2 + 2e^-} \underset{\text{décharge}}{\overset{\text{charge}}{\rightleftharpoons}} \mathrm{Cd + 2OH^-} \quad \text{(-0,808 V).} \qquad \text{(A.2)}$$

Elles sont reproduites dans la figure A.1 avec des flèches pleines pour le fonctionnement en décharge et creuses pour la charge.

Les potentiels d'oxydo-réduction (dans un milieu basique) des couples présents aux électrodes de Nickel et de Cadmium valent $0,520$ V et $-0,808$ V, respectivement. Ce qui donne un potentiel standard d'oxydo-réduction du système complet de $1,328$ V.

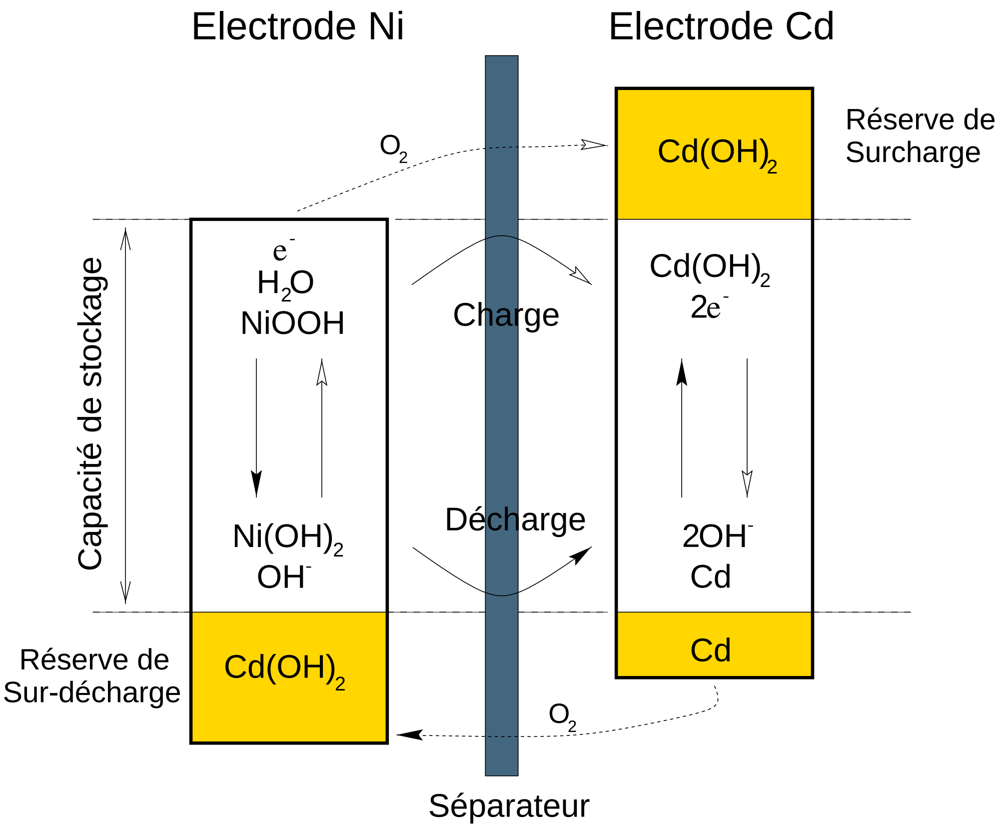

FIG. A.1: *Représentation schématique du fonctionnement d'une batterie NiCd*

A.2 Les systèmes de protection

Outre les réactions principales, des réactions secondaires de décomposition de l'électrolyte ont lieu, plus particulièrement pendant les phases de surcharge et de sur-décharge. La première de ces réactions annexes est le développement d'oxygène sous forme gazeuse :

$$4OH^- \rightleftharpoons 2H_2\,O + O_2^{(g)} + 4e^- \qquad (0{,}4011\ V) \qquad (A.3)$$

La seconde concerne l'hydrogène :

$$2H_2O + 2e^- \rightleftharpoons 2OH^- + H_2{}^{(g)} \qquad (\text{-}0.828\ V) \qquad (A.4)$$

Il faut protéger la batterie contre une augmentation de la pression trop importante due à la production de gaz. Pour ce faire, un espace libre est ménagé à l'intérieur de la batterie afin de permettre au gaz produit (et plus particulièrement à l'oxygène) d'entrer en contact avec l'autre électrode où il sera décomposé.

Plus précisément, les systèmes de protection sont présentés dans la figure A.1. Le principe est de faire de l'électrode de nickel, l'électrode limitante et d'ajouter à chacune des électrodes des réserves de « combustible ».

A.2.1 En cas de surcharge

Durant la charge, $Ni(OH)_2$ est oxydé en NiOOH et simultanément, à l'électrode cadmium, du $Cd(OH)_2$ est réduit en Cd métallique. Au fur et à mesure de la charge, la quantité de $Ni(OH)_2$ diminuant, sa vitesse d'oxydation baisse à tel point que la réaction de production d'oxygène (A.3) devient prédominante. Lorsque l'électrode nickel (limitante) atteint sa pleine capacité, tout le courant appliqué est consommé dans la production d'oxygène et on peut observer une hausse de la tension, ainsi que de la pression et de la température interne. À ce moment là, l'électrode de cadmium n'est pas totalement chargée. La réserve de $Cd(OH)_2$ qu'elle contient est alors réduite et sa capacité augmente. La vitesse de décompostion de l'oxygène au niveau de l'électrode de cadmium dépend de la pression à l'intérieur de la batterie et après un certain temps, un palier stable est atteint : tout l'oxygène produit par l'électrode de nickel est décomposé par l'électrode de cadmium. Des valeurs constantes de pression et de température sont alors observées.

A.2.2 En cas de sur-décharge

Au cours de la décharge, les réactions inverses se produisent avec la réduction de NiOOH en $Ni(OH)_2$ et l'oxydation du Cd en $Cd(OH)_2$. En fin de décharge, lorsque l'électrode de nickel est totalement déchargée, d'autres couples électrochimiques entrent en jeu, si l'utilisateur continue à décharger la batterie. En principe, la production d'hydrogène (A.4) devrait avoir lieu a l'électrode de nickel. Cependant, comme la réaction de consommation d'hydrogène qui devrait avoir lieu à l'électrode de cadmium est particulièrement lente, le cycle production-décomposition ne peut être stable, contrairement à celui de l'oxygène. Pour cette raison, du $Cd(OH)_2$ est ajouté à l'électrode de nickel. En cas de sur-décharge, la réduction de ce $Cd(OH)_2$ en Cd prédomine la réaction de production d'hydrogène. Comme au même moment du Cd est oxydé à l'électrode de cadmium, la tension totale de la batterie est alors proche de 0 V. Lorsque l'électrode de cadmium est elle-même vide, la tension change de sens et le phénomène inverse à celui de la surcharge se produit : de l'oxygène est produit par l'électrode de Cd et est décomposé par l'électrode de nickel avec augmentation de la pression et de la température à l'intérieur de la batterie.

Annexe B

Détection d'objets : outils en analyse d'images

L'objectif de cette annexe est de présenter certaines méthodes de traitement d'images. Elle comporte certaines redites vis-à-vis du corps principal du document. Cependant elle contient également des détails techniques et certains liens entre les méthodes que nous avons jugés intéressants de présenter.

B.1 Les approches par régions

Le principe général de ces méthodes dites de « segmentation » est de partitionner l'image en des zones homogènes (au sens d'un certain critère) appelées régions (ou classes). La segmentation part de l'hypothèse fondatrice que l'image à traiter est la réalisation statistique d'une réalité que l'on cherche à découvrir. Dans ce domaine, les travaux en statistique de C. Graffigne ([Graffigne93] et [Cocquerez95, p. 378-388]) et de l'équipe de W. Pieczynski [Pieczynski94, Benmiloud95, Peng95, Giordana97, Salzenstei98] sont particulièrement complets.

Le premier type d'algorithme est purement local, c.-à-d. qu'il n'utilise pas d'information générale sur la forme ; il est sans doute plus adapté aux images texturées (c'est-à-dire aux images présentant des zones constituées de motifs répétés). Le second type de méthode est d'ordre global, ce qui requiert la classification de chaque pixel à partir de toute l'image, c'est-à-dire en exploitant la totalité de l'information disponible.
Il existe de très nombreux travaux sur ce type de segmentation que l'on peut globablement classer dans trois catégories suivant l'importance donnée aux connaissances *a priori* utilisées dans la méthode :

- non supervisé : aucune information *a priori* n'est utilisée, seul le nombre maximal de régions est fixé par avance ;

- semi-supervisé : l'estimation *a priori* des paramètres des échantillons de régions n'est pas nécessaire, mais un certain nombre de seuils doivent néanmoins être choisis ;

– supervisé : le nombre de classes est fixé *a priori* et les paramètres caractérisant chaque classe (région) peuvent être appris par avance sur des fenêtres de petites tailles caractéristiques de ces textures.

Le cas non supervisé est généralement traité par les méthodes de champ de Markov caché et les techniques employées sont alors très proches de celles utilisées pour les chaînes de Markov cachées (voir § 3.7, page 28). En fait, les HMMs représentent un compromis entre les méthodes locales (de type champs de Markov) et globales : les observations sont composées d'une série de sections stationnaires.

B.1.1 Les champs de Markov cachés

Les champs Markoviens cachés constituent la méthode de référence pour le cas non-supervisé dans un cadre global.

L'image à interpréter $y = (y_s)_{s\in\mathcal{S}}$ est une version bruitée de l'image inobservable des classes $x = (x_s)_{s\in\mathcal{S}}$, elle-même réalisation (configuration) d'une variable aléatoire $X = (X_s)_{s\in\mathcal{S}}$, où $\mathcal{S}$ est l'ensemble des sites (avec, le plus souvent, confusion entre les sites et les pixels). Les variables aléatoires X et Y sont appelées champs aléatoires. Chaque X_s est à valeur dans l'ensemble fini et discret de classes $\Omega = \{\omega_1, \dots, \omega_K\}$, alors que chaque Y_s prend ses valeurs dans $\mathbb{R}$. Le problème de la segmentation est alors d'estimer la réalisation inobservable x du champ X à partir de l'image observée y, elle-même réalisation du champ Y.

Un système de voisinage $\mathcal{V}$ est constitué par des parties de $\mathcal{S}$ et le champ aléatoire X est un champ markovien associé à ce système de voisinage si et seulement si

– $P(X = x) > 0$, $\forall x \in \Omega$ et,

– $P\left(X_s = x_s | X_r = x_r, r \in \mathcal{S} - \{s\}\right) = P\left(X_s = x_s | X_r = x_r, r \in v_s\right)$: c'est-à-dire que la connaissance d'un voisinage v_s du pixel s est suffisante pour calculer la loi marginale en chaque site s.

Un sous-ensemble c de $\mathcal{S}$ est appelé une clique relative au système de voisinages $\mathcal{S}$ si c est un singleton ou si deux pixels quelconques de c sont voisins (au sens de $\mathcal{V}$). Soit $\mathcal{C}$ l'ensemble des cliques définies sur $\mathcal{V}$, le théorème de Hammersley-Clifford [Besag74] démontre qu'un champ aléatoire X est un champ markovien associé au système de voisinages $\mathcal{V}$ si et seulement si sa distribution de probabilité $P(x)$ est une mesure de Gibbs définie par

$$\forall x,\ P(X = x) = \frac{1}{Z}\,\mathrm{e}^{-U(x)}, \tag{B.1}$$

où U est un fonction appelée énergie, définie comme une somme de potentiels déterminés en toutes les cliques auxquelles x appartient. Z est généralement appelée fonction de partition, c'est en fait la constante de normalisation de la probabilité P et elle est définie par $Z = \sum_x \mathrm{e}^{-U(x)}$.
La segmentation consiste alors à estimer x, l'image d'étiquettes, connaissant l'image observée y, c'est-à-dire une version bruitée (généralement par un bruit

blanc Gaussien) $P(x|y)$. Cette probabilité *a posteriori* est calculée grâce au théorème de Bayes

$$P(x|y) = \frac{P(y|x)\,P(x)}{P(y)} \tag{B.2}$$

où $P(y)$ est un *a priori* supposé constant et où $P(y|x)$ modélise le comportement des observations à l'intérieur d'une région homogène, par exemple une texture.

La segmentation (restauration de x) qui consiste à estimer les variables cachées peut être effectuer par des méthodes bayésiennes comme celles déjà présentées pour le cas des systèmes à états (voir le chapitre 3, p. 15) :

- l'estimateur du MAP qui cherche $\hat{x}$ qui maximise $P(X = .|Y = y)$ (voir § 3.5.2 p. 20), comme introduit initialement dans [Geman84] ;

- l'estimateur du MPM qui maximise de façon plus locale, selon $\forall\ s \in\ \mathcal{S}$, $\hat{x}_s\ =\ \arg\ \max\ P(x_s|y)$ (voir § 3.5.1 p. 20), comme introduit initialement dans [Marroquin87].

Ces différentes techniques nécessitent le calcul de densités de probabilités. Ces calculs étant le plus souvent impossibles explicitement, les densités de probabilités sont souvent estimées à l'aide d'échantillonneur, comme celui de Gibbs (voir le § 2.4.1) ou celui de Metropolis-Hastings (§ 4 page 14). En outre, après l'ajout d'un paramètre traditionnellement appelé température dans la loi de Gibbs, la segmentation peut être obtenue par une méthode d'optimisation. Cette optimisation peut être déterministe comme l'ICM (*Iterated Conditional Mode*) [Besag86], lequel, bien que ne garantissant pas la convergence vers le minimum global, est très rapide. Il s'agit, en effet, d'un simulateur de Gibbs à température nulle, c'est-à-dire que l'algorithme converge vers un minimum local de l'énergie *a posteriori* puisqu'un seul puits de potentiel est visité, contrairement au recuit simulé. L'optimisation peut également être réalisée par des méthodes stochastiques comme le recuit simulé qui consiste en un échantillonneur de Gibbs combiné avec une décroissance de la température.
Il existe également une technique basée sur les chaînes de Markov [Benmiloud95] qui utilise un parcours de type Hilbert-Peano (voir sa définition dans le § 9.4.2, figure 9.14, p. 144) pour transformer les processus aléatoires bi-dimensionnels (les champs aléatoires) en processus mono-dimensionnels. ce qui permet de réduire considérablement le nombre de paramètres puisque l'on considère le processus comme stationnaire sur de relativement grandes régions de l'image.

Parallèlement, l'estimation de paramètres de ces champs Markoviens (ceux des fonctions de potentiels $U(x)$ et de texture-bruit, pour le passage de x à y) peut être réalisée de la même manière que dans le cas des chaînes de Markov (voir également § 3.7, p. 28), avec, principalement, les méthodes EM [Peng95] et ICE [Pieczynski94]. De nombreux estimateurs sont classiquement utilisés

- l'estimateur du maximum de vraisemblance qui choisit $\hat{\theta}(x)$ qui maximise $P_{\theta_n}(X = x)$,

- l'estimateur du maximum de pseudo-vraisemblance qui cherche à maximiser $PL_\theta(x) = \prod_{s\in\mathcal{S}} P_\theta(X_s = x_s|X_r = x_r,\, r \in \mathcal{S} - \{s\})$.

B.1.2 Exemple de croissance de régions

Les méthodes structurales introduisent de l'information *a priori* sur les caractéristiques des objets à détecter. On peut citer les méthodes à base de classification locale des pixels à partir du niveau de gris ou d'autres attributs locaux. Chaque pixel est classifié, puis il faut extraire les composantes connexes de pixels de même classe pour obtenir une partition de l'image en zones homogènes. Cette méthode a l'inconvénient majeur de n'exiger aucune cohérence spatiale : chaque pixel est classé indépendamment de ses voisins. De plus, un post-traitement est souvent nécessaire pour supprimer les petites régions.
Une classe de méthodes plus riche est composée des techniques basées sur la croissance de régions. À partir de germes disséminés dans l'image, ce type de méthode cherche à regrouper de façon itérative les pixels voisins selon le double critère d'homogénéité et de connexité. Le prédicat d'homogénéité est basé sur la variance ou les attributs de texture. On peut citer l'algorithme de Chassery-Garbay [Chassery84] qui est basé sur la corrélation d'histogrammes locaux (utilisés comme descripteurs de textures).

Afin de segmenter l'image en régions homogènes, il est également possible d'utiliser des techniques de fusion, qui permettent l'agrégation de petites zones homogènes en des régions plus grandes. Cette étape est réalisée à partir du graphe d'adjacence des zones, et d'un critère d'uniformité.

Dans ce cadre, une approche originale a été étudiée au LEP [Philip97, Létang98]. Elle combine les avantages des méthodes de ligne de partage des eaux et de segmentation par croissance de régions cohérentes. L'idée de ces lignes de partage des eaux est de considérer l'image comme une surface dans un espace de dimension 3 et d'utiliser la terminologie de la géographie. Une ligne de partage des eaux est définie comme étant une crête formant la limite entre deux bassins versants. De façon intuitive, deux points de la surface appartiennent au même bassin versant si, lorsque de l'eau tombe en ces points, elle coule vers le même minimum de la surface. Cette détection de ces lignes de partage des eaux peut être effectué par morphologie mathématique, mais l'implantation choisie dans ce travail repose sur [Vincent91]. Après une première pré-segmentation de ce type sur l'image des modules du gradient, un graphe d'adjacence des bassins versants obtenus (les classes) est construit. Cette étape est suivie par une phase de croissance de régions cohérentes. Cette cohérence est mesurée sur l'ensemble des classes C par une énergie E que l'on va chercher à minimiser et définie par

$$E = \sum_{c \in C} \sigma^2(c) + \lambda f(K)\, l(K), \tag{B.3}$$

où $\sigma^2(c)$ est la variance de la classe c, K est l'ensemble des frontières inter-classes, $l(K)$ sa longueur et $f(K)$ une fonction particulière. Le paramètre λ contrôle les poids respectifs des deux parties de l'énergie. Plus λ sera grand et plus le nombre de classes après minimisation sera petit. Cette fonction d'énergie n'est pas convexe et, tout comme dans de nombreux problèmes de minimisation, elle possède de nombreux minima locaux. La fusion est donc réalisée à la manière d'un recuit simulé, à partir de la pré-segmentation en bassins versants et pour une valeur de λ initialisée à λ_0. Pour chaque classe de la segmentation courante, un test est effectué pour mesurer la validité de la fusion avec toutes ses classes voisines ; si le test est concluant, les classes sont fusionnées et l'arbre d'adjacence mis à jour. Cette étape est réitérée jusqu'à convergence. Si le nombre de classes obtenues est encore trop important, la valeur de λ est augmentée et l'étape de fusion recommence. Dans l'implantation décrite par M. Philip dans [Philip97], la fonction $f(K)$ utilise les informations de gradient afin de décourager tout regroupement de régions séparées par un gradient ayant un module important :

$$f(K) = \int_K 1 - \left(\frac{\| g \|}{\| g_0 \|} \right)^{\alpha}, \tag{B.4}$$

où $\| g_0 \|$ est le maximum du module du gradient sur toute l'image, $\| g \|$ le module du gradient local et α un paramètre fixé de façon empirique à des valeurs comprises entre $0,5$ et 1. Un résultat obtenu par cette méthode est proposée dans la figure B.1.

B.2 Les approches par frontières : complément sur le filtrage

Le principe général des méthodes frontières est de détecter les variations brusques d'intensité, c.-à-d. les parties hautes fréquences de l'image, les frontières entre des zones homogènes dans l'image. La discussion suivante complète l'introduction aux filtres linéaires présentée dans 8.2.2, p. 102 et suivantes.

Les filtres linéaires à Réponse Impulsionnelle Finie (RIF) ont l'avantage de pouvoir être mis facilement en œuvre par convolution numérique. Malheureusement, le fait que la réponse impulsionnelle soit finie limite les possibilités de ces filtres. En outre, l'approximation d'un fitre à réponse impulsionnelle infinie (RII) par un filtre RIF n'est pas toujours possible ou peut alors conduire à un nombre de coefficient rédhibitoire pour le temps de calcul. La détermination des filtres RII est basée sur le découpage de la fonction de transfert du filtre (exprimée par la transformation en z) en partie causale (valeurs positives de la variable spatiale) et en partie anticausale (valeurs négatives de la variable spatiale). Le filtre dérivateur de Deriche est l'exemple le plus connu de filtre linéaire RII. L'annexe C présente une implantation du filtrage passe-bas sous la forme d'un filtre récursif 1D.

Les filtres présentés jusqu'à présent partagent, outre le fait d'être linéaires, la caractéristique d'être définis indépendamment du contexte de leur utilisation, c'est-à-dire de l'image. Il existe cependant d'autres types de filtrage qui prennent en compte l'image elle-même pour déterminer les coefficients du filtre. On peut citer le cas du filtrage optimal qui consiste à minimiser un critère quantitatif de qualité sur l'image résultat, les deux exemples classiques de ce type de critère sont le rapport signal à bruit (filtre de Wiener) et la détection de contours (filtres de Canny-Deriche, Shen).
Les techniques du filtrage linéaire ont l'avantage d'être simples d'étude et d'implantation. Le choix d'un opérateur de filtrage résulte d'un compromis entre effet passe-bas, pour la réduction de bruit, et passe-haut pour le rehaussement

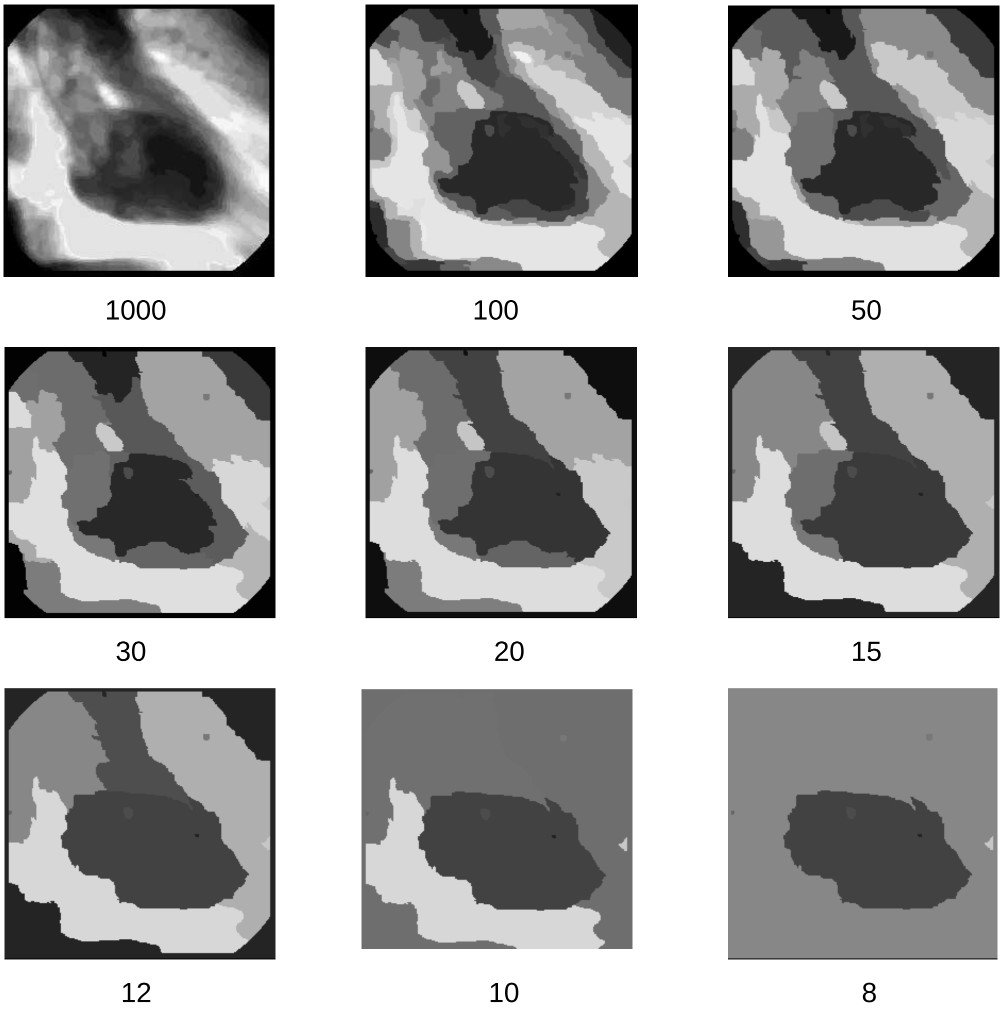

FIG. B.1: *Exemple de résultat obtenu par la méthode d'agrégation des classes sur une image test avec un nombre de classes diminuant.*

de contraste. Pour aller au delà de ce compromis, il faut passer au filtrage linéaire adaptatif (dont le comportement entrée/sortie est globalement non linéaire) ou à des opérateurs dont la structure est intrinsèquement non linéaire.
Les familles d'opérateurs non linéaires les plus courantes sont :

- le filtrage morphologique qui repose sur la notion de forme (voir § B.4) ;
- le filtrage homomorphique qui travaille dans une image transformée par une fonction non-linéaire puis la transformation inverse est appliquée (par exemple : moyennes non linéaires, filtrage du bruit additif et convolutif) ;
- le filtrage polynomial (ou de Volterra) qui permet d'introduire conjointement des effets linéaires et non linéaires (par des termes de réponse multi-impulsionnelle), en calculant l'image filtrée sous la forme d'un polynôme de l'image d'origine (exemple : *unsharp masking*, rehaussement de

contraste) ;

– le filtrage d'ordre qui calcule une fonction des statistiques d'ordre (exemple : le filtre médian).

B.3 Reconstruction de l'image d'origine à partir de sa représentation multi-échelle

La discussion suivante complète la remarque de la page 127 : « à partir des dérivées partielles $L_{l,m}(x, y, \sigma)$, il est théoriquement possible de reconstuire l'image d'origine ».
Plus précisément, l'évaluation locale de ces différentielles, autour d'un point $P(x_p, y_p)$ permet une reconstruction correcte sur un rayon d'environ $1,2\sigma$ autour de ce point P. Pour ce faire, on peut utiliser les produits de polynômes d'Hermite $H_m\left(\frac{x-x_p}{\sqrt{2}\,\sigma}\right) H_n\left(\frac{y-y_p}{\sqrt{2}\,\sigma}\right)$ (avec $n = l - m$, ordre de dérivation dans la direction y) qui forme une base orthogonale avec la fonction fenêtre

$$\Psi_\sigma^p(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x - x_p)^2 + (y - y_p)^2}{2\sigma^2}\right). \tag{B.5}$$

Ainsi l'image originale $L(x, y)$ peut être reconstruite par une somme infinie :

$$L(x, y) = \sum_{m\geqslant 0,\, n\geqslant 0} V_{m,n}\, H_m\left(\frac{x - x_p}{\sqrt{2}\,\sigma}\right) H_n\left(\frac{y - y_p}{\sqrt{2}\,\sigma}\right) \tag{B.6}$$

où les coefficients $V_{m,n}$ sont de la forme

$$V_{m,n} = \gamma_{m,n} L_{l,m}, \tag{B.7}$$

et

$$\gamma_{m,n} = \frac{1}{m!\, n!}\left(\frac{\sigma}{\sqrt{2}}\right)^{m+n}. \tag{B.8}$$

Ce résultat permet de définir une corrélation croisée et fenetrée entre deux images $f(x, y)$ et $g(x, y)$ par

$$C_p(f, g) = \int_R \Psi_\sigma^p(x, y)\; f(x, y)\; g(x, y)\; dxdy = \sum_{m\geqslant 0,\, n\geqslant 0} \frac{F_{m,n}^\sigma\, G_{m,n}^\sigma}{m!\, n!}, \tag{B.9}$$

avec

$$F_{m,n}^\sigma = \sigma^{m+n}\left(\frac{\partial^{m+n}}{\partial x^m\, \partial y^n}\right) F \tag{B.10}$$

représentant les informations de dérivée normalisées pour l'image f filtrée par une gaussienne de taille σ et donnant l'image F^σ, autour du point P, et de même pour $G_{m,n}^\sigma$ vis-à-vis de l'image G^σ obtenue par filtrage de g. Ainsi, la corrélation

croisée (fenêtrée) des images f et g peut être calculée comme un produit scalaire (pondéré) entre leurs vecteurs de dérivées. En d'autres termes, un produit scalaire de deux vecteurs de dérivées est toujours équivalent à une corrélation croisée des images correspondantes autour du point d'intérêt (x_p, y_p).
L'équation (B.9) permet d'estimer la corrélation croisée $C_p(f, g)$ comme l'application d'une combinaison linéaire d'opérateurs différentiels à l'image filtrée G. Le calcul des dérivées partielles permet de définir une convolution discrète appliquée à l'image G^σ. L'information utile aux tâche de reconnaissance de formes est généralement concentrée dans quelques opérateurs différentiels seulement, de telle sorte que la convolution discrète ne comporte que quelques paramètres non nuls.
Afin de tester cette décomposition, un test de reconstruction a été effectué. Il s'agit de reconstruire une image $f(x, y)$ autour d'un point de référence $P(x_p, y_p)$. En fait, la reconstruction va s'effectuer sur l'image filtrée F au point $P'(x', y')$ de telle sorte que

$$F(x', y') = f(x', y') \exp\left(-\frac{(x_p - x')^2 + (y_p - y')^2}{2\,\sigma^2}\right) \tag{B.11}$$

sera reconstruit par

$$\sum_{m \geqslant 0,\, n \geqslant 0} \frac{F^\sigma_{m,n}}{m!\, n!}\, \sigma^l \left(\frac{\partial^l}{\partial x^m\, \partial y^n}\right) D \tag{B.12}$$

où les termes différentiels sont évalués au point $P(x_p, y_p)$ et où

$$D(x', y') = \exp\left(-\frac{(x_p - x')^2 + (y_p - y')^2}{2\,\sigma^2}\right) \tag{B.13}$$

est la réponse basse fréquence d'une fonction delta d'intensité $2\pi\sigma^2$. L'erreur de reconstruction au point $P'(x', y')$ s'estime par la différence entre $F(x', y')$ (B.11) et sa reconstruction définie dans (B.12). L'erreur de reconstruction totale s'exprime comme l'erreur quadratique moyenne sur la fenêtre gaussienne centrée sur le point P.
Ce mécanisme de reconstruction a été appliqué à une image de test présentée dans la figure B.2 qui a été obtenue après un filtrage de $\sigma = 25$ pixels.

La somme infinie de (B.12) a été estimée aux ordres de dérivation $l = m + n$ de 0 à 8 pour les points de référence numérotés de 1 à 4 dans l'image B.2. L'erreur de reconstruction est normalisée afin de tenir compte des différences d'information présentes dans les différentes régions d'interêt. Cette erreur résultante est présentée dans la figure B.3. On peut y voir que, contrairement à l'intuition mathématique, l'ajout de nouveaux termes différentiels ne diminue l'erreur de reconstruction que jusqu'à un certain ordre. Après l'ordre 4, l'ajout de termes d'ordre supérieur accroît l'erreur et ceci de façon très rapide. Ceci est dû à la discrétisation imposée pour le calcul numérique qui conduit à des erreurs importantes pour des ordres élevés de dérivation : « trop dériver revient à ajouter du bruit ». Tout comme les expériences en vision humaine, il semble donc que la dérivation jusqu'à l'ordre 4 permette l'introduction d'informations utiles et qu'en outre, cette information soit calculable.

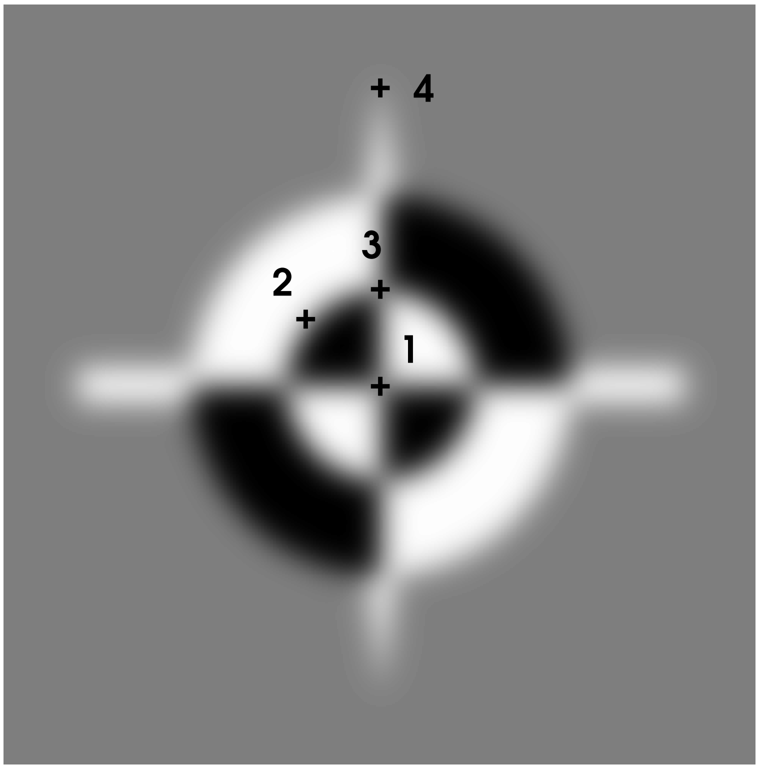

FIG. B.2: *Image de test pour la reconstruction par utilisation de l'information multi-échelle.*

B.4 La Morphologie mathématique

L'idée de base est de comparer l'image avec une sonde que l'on maîtrise parfaitement, l'élément structurant. L'originalité de cette méthode par rapport aux techniques de filtrage linéaire est d'examiner les relations booléennes telles que l'inclusion ou l'intersection existant entre l'image et l'élément structurant. Dans ce cadre, les images sont assimilées à des ensembles et la morphologie mathématique est connue pour ses relations ensemblistes d'inclusion et d'intersection qui sont à la base des opérateurs morphologiques de base que sont l'érosion et la dilatation. Une image à niveaux de gris est ainsi assimilée au sous-graphe de sa fonction « intensité », c.-à-d. à l'ensemble des points situés « sous » la surface définie par $(x, y, I(x, y))$ représentant l'image. Les transformations morphologiques se présentent sous la forme d'un édifice constitué de primitives (dilatation et érosion) et de principes de construction de primitives d'ordre supérieur. Résoudre un problème d'analyse d'images revient alors à choisir des primitives et à les assembler pour progressivement éliminer de l'image tout ce qui ne correspond pas à ce que l'on cherche. Il est aussi possible de choisir des éléments structurants de différentes tailles et ainsi d'intégrer la morphologie mathématique dans un contexte multi-échelle (voir [vdB94] et [Buteau98]). Pour une introduction à la morphologie mathématique, on pourra se reporter à l'annexe C de [Cocquerez95], rédigée par F. Prêteux.

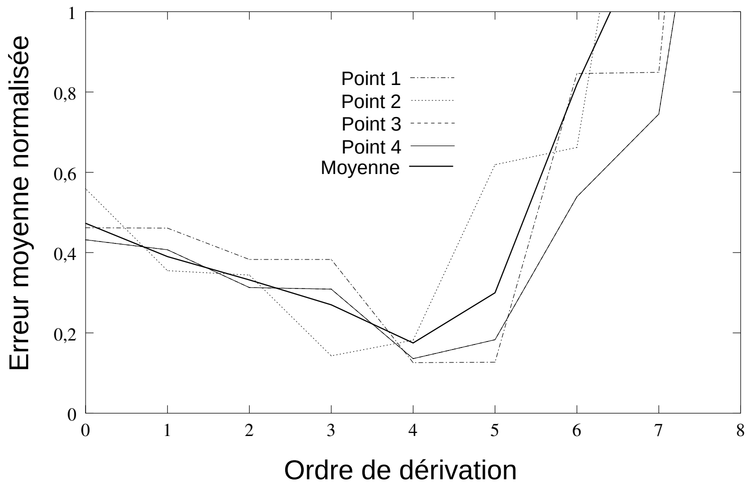

FIG. B.3: *Erreur de reconstruction à différents ordres pour les points de la figure B.2.*

Les opérations de morphologie mathématique peuvent être implantées de façon rapide en utilisant les propriétés de la transformation de la pente (ou *slope transform*). Cette transformation s'appuie sur la notion de dilatation tangentielle qui décrit le contact de surfaces dérivables et généralise la dilatation classique.

Partant de la définition classique de la dilatation de deux fonctions réelles f et g :

$$(f \oplus g)(x) = \sup_u [f(u) + g(x-u)] \tag{B.14}$$

Van den Boomgaard [Dorst94] montre que sous l'effet de la dilatation de f et g convexes, un segment de droite infinitésimal est transporté de façon parallèle de $(\overline{u}, f(\overline{u}))$ à $(\overline{x}, f(\overline{u}) + g(\overline{x} - \overline{u}))$, avec $\overline{x}$ donné implicitement par la relation $\nabla f(\overline{u}) = \nabla g(\overline{x} - \overline{u})$. D'où l'on déduit que :

$$\nabla(f \oplus g)(\overline{x}) - \nabla f(\overline{u}) = \nabla g(\overline{x} - \overline{u}) \ . \tag{B.15}$$

Ceci constitue le théorème de la pente, qui définit la dilatation $(f \oplus g)$ comme une opération locale, dont le résultat ne dépend que des propriétés locales de la fonction f et de la fonction structurante g.

La dilatation tangentielle, notée $\widetilde{\oplus}$, est construite en appliquant ce théorème sur tous les points du domaine de f. Les fonctions structurantes g doivent être strictement concaves ou convexes. La fonction f, en revanche, peut contenir des portions concaves et des portions convexes. Dans le cas où elle est strictement concave, la dilatation tangentielle se construit de façon équivalente à la dilatation classique. Sinon, la relation entre les deux est la suivante : $\sup(f \widetilde{\oplus} g)(x) = (f \oplus g)(x)$. La figure B.4(a) (tirée de [Dorst94] et reproduite dans [Buteau98, p. 133]) présente l'application du théorème de la pente. Il est

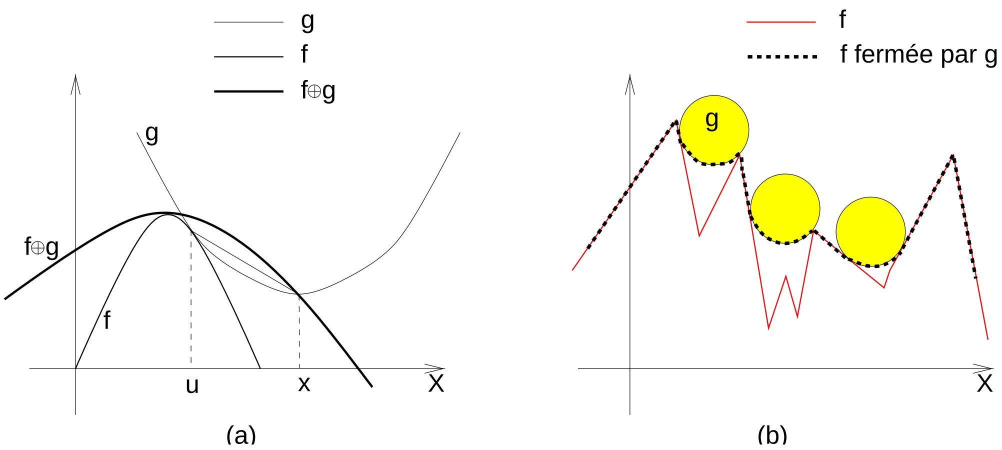

FIG. B.4: *(a) application du théorème de la pente à une fonction convexe, (b) schéma de la fermeture.*

ensuite possible de définir selon le même principe l'opération complémentaire : l'érosion tangentielle. L'application successive d'une dilatation puis d'une érosion par le même élément structurant est appelée fermeture et a pour effet de combler les trous, les chenaux étroits et de connecter les objets proches. Cette opération est schématisée dans la figure B.4(b) et ce mécanisme a été utilisé dans le § 9.2.2, p. 122 pour « supprimer » de l'image la bande sombre représentant la cathéter.

Ce théorème possède une autre caractéristique très intéressante : il est très facile de connaître l'origine du point courant par la transformation morphologique. En fait, on sait précisément que le point courant d'abscisse x provient du point d'abscisse u et on a donc une mesure de la distance impliquée par la transformation. Ceci permet la constitution de cartes d'index codant l'origine de chacun des points transformés (cette propriété a été utilisée dans la section 9.4.2). En outre, si la fonction structurante g est quadratique, le calcul de la transformée de la pente permet également d'enregistrer une carte de distance euclidienne de façon très rapide. En effet, les éléments structurants quadratiques sont strictement séparables et peuvent donc être appliqués sans perte dans la direction horizontale indépendamment de la direction verticale.

L'implantation utilisée ici utilise l'information de la valeur de la pente pour réduire l'espace de recherche du $\sup_u$ dans (B.14). Cette méthode (décrite dans [vdB96]) permet d'accoitre de façon substantielle la vitesse d'application du filtre.

Annexe C

Implantation du filtrage passe-bas sous la forme d'un filtre récursif 1D

Afin de pouvoir observer l'image à différents niveaux de détail, fixés par les échelles du domaine d'étude, celle-ci est rendue plus ou moins floue par des filtres passe-bas gaussiens. Les filtre gaussiens sont des approximations commodes des filtres passe-bas parfaits qui sont de la forme $\sin(x)/x$ (dans le domaine spatial) et donc très complexes à mettre en œuvre.
Les noyaux utilisés sont conçus à partir d'une technique de filtrage à une dimension, inspirée de l'étude de Deriche [Deriche90] et étudiée au LEP [Philip97] [Létang98].
Un filtre 1D élémentaire est doté d'un noyau symétrique positif, dont la hauteur décroît de façon exponentielle, lorsqu'on s'éloigne de son centre. Le schéma synoptique de ce filtre est représenté dans la figure C.1 et son expression est de la forme suivante :

$$\tilde{S}(n) = (1-\alpha)\cdot S(n) + \alpha \cdot S(n-1) \quad \text{avec } 0 < \alpha < 1. \tag{C.1}$$

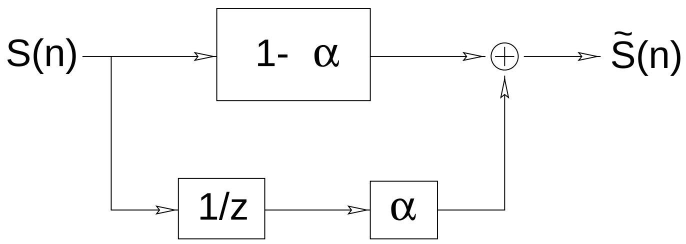

Fig. C.1: *Schéma synoptique du filtre récursif.*

Une itération de ce filtre consiste en fait à deux passage l'un causal (C.1) et l'autre anti-causal, ce qui correspond à un filtre total de fonction de transfert

$$H(z) = \frac{z\,(1-\alpha)^2}{(z-\alpha)\,(1-\alpha z)}. \tag{C.2}$$

Ce filtre vérifie bien $H(z) = H(z-1)$, de telle sorte que sa réponse impulsionnelle est bien symétrique. σ^2, le moment d'ordre 2 peut alors est estimé en fonction de α :

$$\sigma = \sqrt{\frac{2\alpha}{(1-\alpha)^2}}\ , \tag{C.3}$$

mais c'est plutôt la relation inverse qui est importante, c'est-à-dire l'estimation de α en fontion de l'écart type σ de la gaussienne à approximer :

$$\alpha = 1 + \frac{1}{{\sigma_0}^2} - \sqrt{\left(1 + \frac{1}{{\sigma_0}^2}\right)^2 - 1} \tag{C.4}$$

où σ_0 dépend également du nombre d'itération N, du pas de discrétisation p (=distance entre deux pixels consécutifs), ainsi que de la dimension de l'espace d suivant :

$$\sigma_0 = \frac{\sigma}{p}\sqrt{\frac{d}{N}}\ . \tag{C.5}$$

Après un certain nombre d'itérations, ce filtre offre une bonne approximation d'un filtrage par un noyau gaussien, en vertu du théorème central limite appliqué à une somme de variables, de moyenne nulle, indépendantes et identiquement distribuées. En pratique, il s'avère que deux ou trois itérations suffisent. La figure C.2 présente l'application d'un tel filtre passe-bas à un signal d'entrée 1D de type Dirac. Pour cette figure les paramètres suivants ont été utilisés : $\sigma = 10$, $N = 1, 2, 3, 10$ et comme on est en 1D, on a $p = 1$, $d = 1$.

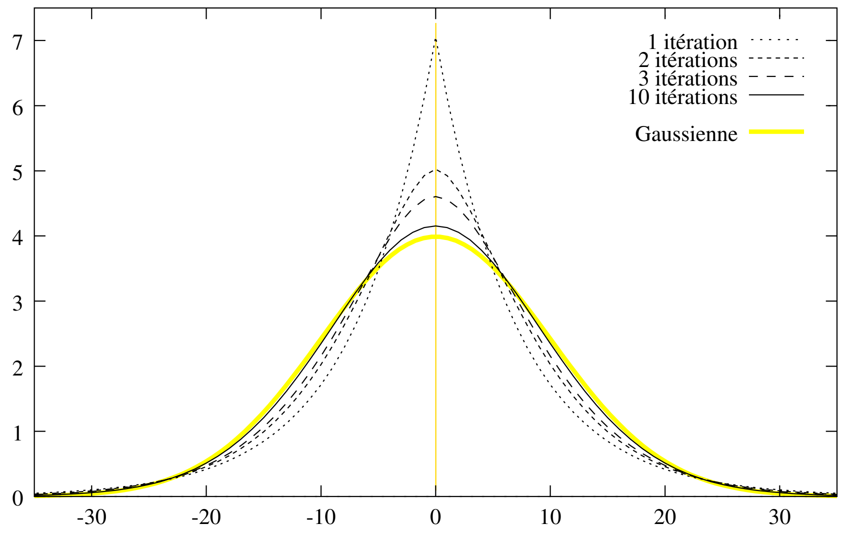

FIG. C.2: *Réponse impulsionnelle du filtre récursif après quelques itérations. Comparaison avec la réponse du noyau gaussien.*

La séparabilité du noyau Gaussien nous permet de travailler indépendamment suivant les lignes et les colonnes pour construire facilement des filtres 2D

isotropes, par combinaison de deux noyaux 1D appropriés, un dans la direction désirée et l'autre dans la direction perpendiculaire.
La mise en œuvre de tels filtres est simple et peu coûteuse. La principale difficulté est de conserver les propriétés des filtres récursifs, lorsqu'on applique à un nombre d'échantillons fini. Une solution intéressante pour l'implantation est de combiner des balayages successifs de l'image avec des directions de parcours différents. En fait les balayages utilisés sont décris dans la figure C.3 ; il s'agit de passages verticaux, horizontaux auxquels sont adjointes les diagonales principales.

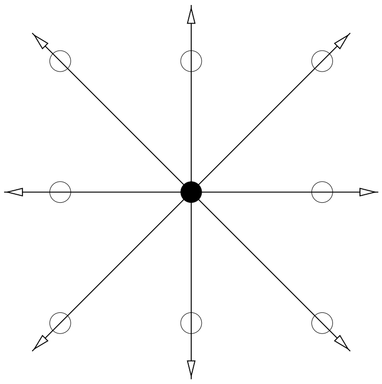

FIG. C.3: *Les balayages de l'image pour le calcul de filtre récursif 2D.*

La nature isotrope du filtre obtenu est montrée dans la fig. C.4. Les équations (C.1) et (C.5) sont assez générales pour pouvoir être utilisées dans n'importe quelle direction dans un espace de dimension quelconque.

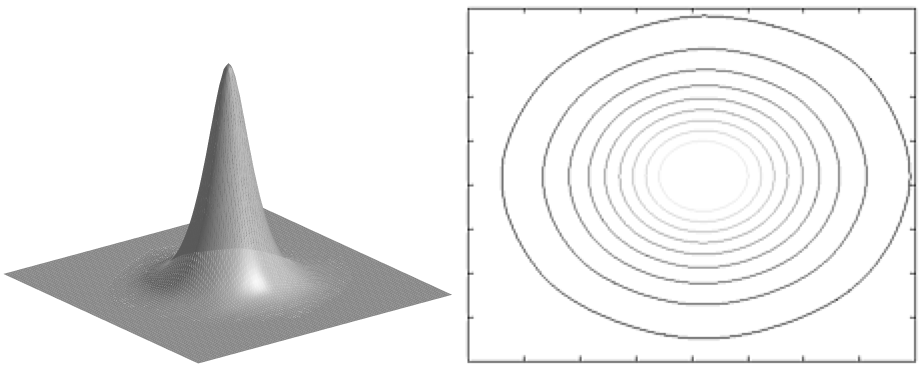

FIG. C.4: *Réponse impulsionnelle du filtre récursif 2D ; courbes de niveaux.*

Cette approximation du noyau gaussien 2D présente un grand intérêt dans le cadre d'une étude multi-échelles. En effet, elle se contente d'un accès aux pixels voisins et évite ainsi le calcul de l'intégrale de (9.1). Elle est également plus rapide car aucune exponentielle n'est calculée mais surtout le passage dans l'espace des fréquences n'est pas obligatoire. Elle s'affranchit également du traditionnel « passage de noyaux matriciels » qui est coûteux en temps de calculs [(45)]

Comme ce type de filtrage approximé est beaucoup plus efficace et précis pour des faibles valeurs de σ, on prendra bien garde d'utiliser au maximum la propriété de linéarité (l'application successive de filtres gaussiens σ_1 et σ_2 est équivalente à la seule application du filtre $\sigma^2 = \sigma_1^2 + \sigma_2^2$) pour construire de façon itérative la pyramide de filtres.

Notons qu'il est aussi possible de transformer le noyau de base du filtre récursif 1D en adaptant son cœfficient α en fonction des variations des niveaux de gris de l'image, pour qu'il préserve ses discontinuités. On parle alors de diffusion anisotrope. Dans ce cas, il devient une fonction de la position, de façon à traiter différemment les zones homogènes et les zones structurées. De tels noyaux proposent un bon compromis entre la suppression des détails indésirables à l'échelle courante et la localisation des bords.

Il est également possible d'utiliser des filtres d'ordre supérieur [van Vliet98], ce qui conduit à une fonction de transfert plus complexe mais permet d'atteindre une bien meilleure isotropicité avec seulement deux directions de balayage.

(45). et ce d'autant plus que la taille du noyau est grande.

Bibliographie

Les nombres en **gras** à la fin de chaque entrée sont les numéros de page de citation.

[Aas99] K. Aas, L. Eikvil, et R. B. Huseby. « Applications of hidden Markov chains in image analysis ». *Pattern Recognition*, 32(4) :703–713, mars 1999. **117**

[Acharya98] R. Acharya, R. Wasserman, J. Stevens, et C. Hinojosa. « Biomedical Imaging Modalities : An Overview ». *In [Singh98]*, pages 20–44, 1998. **84**, **113**

[Amini90] A.A. Amini, T.E. Weymouth, et R.C. Jain. « Using Dynamic Programming for Solving Variational Problems in Vision ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9) :855–867, septembre 1990. **110**

[Andrieu97] C. Andrieu, A. Doucet, et P. Duvaut. « Méthodes de Monte Carlo par chaînes de Markov appliquées au traitement du signal ». Rapport Technique 3, ETIS-ENSEA, 1997. **13**, **14**, **14**

[Arvidsson61] H. Arvidsson. « Angiocardiographic Determination of Left Ventricular Volume ». *Acta Radiologica*, 56(5) :312–39, novembre 1961. **92**, **92**

[Aubert95] F. Aubert et J.-P. Laissy. *Radiologie et Imagerie Médicale.* Que sais-je ?, n° 3004. PUF, 1995. **84**, **113**

[Auger98] F. Auger. *Introduction à la théorie du signal et de l'information.* Collection sciences et technologies. Éditions Technip, 1998. **23**

[Aussem95] A. Aussem. « *Théorie et Applications des Réseaux de Neurones Récurrents et Dynamiques à la Prédiction, à la modélisation et au Contrôle Adaptatif des Processus Dynamiques* ». PhD thesis, Université Paris 5, 1995. **46**, **47**

[Austin75] W. G. Austin. « A reporting system on patients evaluated for coronary artery disease ». *Circulation, American Heart Association*, 51 :7–40, 1975. **118**

[Babaud86] J. Babaud, A. P. Witkin, M. Baudin, et R. O. Duda. « Uniqueness of the Gaussian Kernel ofr Scale-Space Filtering ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1) :26–33, janvier 1986. **126**

[Baldy94] C. Baldy et *et al.*. « Automated Myocardial Edge Detection from Breath-Hold Cine-MR Images : Evaluation of Left Ventricular Volumes and Mass ». *Magnetic Resonance Imaging*, 12(4) :589–598, 1994. **113**, **114**

[Battiti94] R. Battiti. « Using Mutual Information for Selecting Features in Supervised Neural Net Learning ». *IEEE Transactions on Neural Networks*, 5(4) :537–550, 1994. **133**

[Baum70] L. Baum, T. Petrie, N. Soules, et N. Weiss. « A Maximisation technique occuring in the statistical analysis of probabilistic functions of Markov chains ». *Ann. Math. Stat.*, 41 :164–71, 1970. **30**, **32**

[Beier93] J. Beier, T. Joerke, S. Lempert, E. Wellnhofer, H. Oswald, et E. Fleck. « A Comparison of 7 Different Volumetry Methods of Left and Right Ventricle using Postmortem Phantoms ». Dans *IEEE Computers in Cardiology*, pages 33–36, Londres, septembre 1993. IEEE. **92**

[Beier96] J. Beier, E. Wellnhofer, H. Oswald, et E. Fleck. « Accuracy and Precision of Angiographic Volumetry Methods for Left and Right Ventricle ». *International Journal of Cardiology*, 53 :179–88, 1996. **92**, **94**, **172**

[Bellman62] R. E. Bellman. *Applied Dynamic Programming.* Princeton Univ. Press, 1962. **110**

[Bengio94] Y. Bengio et P. Frasconi. « An Input/Ouput HMM Architecture ». Dans G. Tesauro, D. Touretsky, et T. Leen, éditeurs, *Advances in Neural Information Processing Systems 7*, pages 427–434. MIT Press, 1994. **28**

[Benmiloud95] B. Benmiloud et W. Pieczynski. « Estimation des Paramètres dans les chaînes de Markov cachées et segmentation d'images ». *Traitement du Signal*, 12(5) :433–454, 1995. **20**, **30**, **32**, **32**, **183**, **185**

[Bertein98] J.C. Bertein et R. Ceschi. *Processus Stochastique et filtrage de Kalman.* Hermes, 1998. **24**

[Besag74] J. Besag. « Spatial Interaction and the Statistical Analysis of Lattice Systems ». *J. of Royal Statistical Society*, 2 :192–236, 1974. **184**

[Besag86] J. Besag. « On the Statistical Analysis of Dirty Pictures ». *J. of Royal Statistical Society*, 48, 1986. **185**

[Biehl94] M. Biehl et H. Schwarze. « Learning by On-line Gradient Descent ». Rapport Technique LU TP 94-10, Lund University, Sweden, 1994. ftp ://thep.lu.se/pub/Preprints/li_tp_94_10.ps.Z. **61**

[Bishop95] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995. **11**, **11**, **52**, **137**, **142**

[Bossard94] P.-L. Bossard. « *Détections de Contours Réguliers dans des Images Bruitées et Texturéees* ». PhD thesis, INPG, 1994. **109**

[Bourlard94] H. Bourlard et N. Morgan. *Connectionist Speech Recognition – A Hybrid Approach.* Kluwer Academic Publishers, 1994. **150**

[Breiman84] L. Breiman, J.H. Friedman, R.A. Olshen, et C.J. Stone. *Classification and Regression Trees.* Wadsworth Books, 1984. **54**, **54**

[Brody84] W. R. Brody. *Digital Radiography.* Raven Press, 1984. **86**, **91**

[Brugnara92] F. Brugnara, R. De Mori, D. Giuliani, et M. Omologo. « A Family of Parallel Hidden Markov Models ». Dans *Proceedings of Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP'92)*, pages 377–380, San Fransisco, mars 1992. IEEE. **76**

[Buteau98] M. Buteau. « *Analyse d'images radiographiques du thorax pour l'aide au diagnostic des pathologies pulmonaires* ». PhD thesis, Univ. Rennes 1, 1998. **106**, **191**, **192**

[Canu95] S. Canu, X. Ding, et Y. Grandvalet. « *Réseaux de Neurones pour la Prévision à un pas de Temps* », Chapitre 12. Ecole Modulad - ASU, décembre 1995. **44**

[Canu96] S. Canu. « Des Réseaux de Neurones à la Regression Flexible ». Habilitation à diriger des recherches, Université Paris 6, 1996. **11**

[Carpenter87] G. A. Carpenter et S. Grossberg. « ART 2 : Self-Organization of Stable Category Recognition Codes for Analog Input Patterns ». *Applied Optics*, 26(23) :4919–30, décembre 1987. **64**

[Carpenter88] G. A. Carpenter et S. Grossberg. « The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network ». *Computer*, 21(3) :77–88, 1988. **64**

[Celeux95] G. CELEUX, D. CHAUVEAU, et J. DIEBOLT. « On Stochastic Versions of the EM Algorithm ». Rapport Technique RR-2514, INRIA, mars 1995. **16**, **17**, **30**

[Chassery84] J.M. CHASSERY et C. GARBAY. « An iterative segmentation method based on a contextual color and shape criterion ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6) :794–800, 1984. **186**

[Chehikian97] A. CHEHIKIAN, P.-Y. COULON, et F. LUTHON. « Traitement d'images et de séquences d'images... Méthodes, applications et mises en œuvre. ». Ecole des Techniques Avancées, Signal Image Parole, sept. 1997, 1997. **100**, **124**

[Chen92] H. W. CHEN et V. W. SOO. « An Adaptive Back-Propagation Learning Method : a Preliminary Study for Incremental Neural Networks ». *Proc. Intern. Conf. on Neural Networks (IJCNN'92)*, page 173, juin 1992. **62**

[Chen93] H. W. CHEN et V. W. SOO. « Design of Adaptive and Incremental Feedforward Neural Networks ». *Proc. IJCNN*, pages 479–84, 1993. **62**, **62**

[Chen95a] C. H. CHEN, R. G. PAREKH, J. YANG, K. BALAKRISHNAN, et V. HONAVAR. « Analysis of Decision Boundaries Generated by Constructive Neural Network Learning Algorithms ». Rapport Technique TR 95-12, Iowa State University, 1995. **63**

[Chen95b] H. W. CHEN et V. W. SOO. « The Plasticity of Feedforward Neural Networks in Assimilating a Training Instance ». *Neurocomputing*, 7(3) :299–308, avril 1995. **62**

[Clarysse95] P. CLARYSSE, F. POUPON, B. BARBIER, et I. E. MAGNIN. « 3D Boundary Extraction of the Left Ventricle by a Deformable Model with a priori Information ». Dans *2nd IEEE Intern. Conf. on Image Processing (ICIP'95)*, pages 492–95, Crystal City, octobre 1995. **113**, **114**

[Cocquerez95] J.-P. COCQUEREZ et S. PHILIPP, éditeurs. *Analyse d'images : filtrage et segmentation*. Masson, 1995. **101**, **103**, **183**, **191**

[Codrington94] C. W. CODRINGTON et M. F. TENORIO. « Adaptive Gain Networks ». Dans *IEEE International Conference on Neural Networks (ICNN'94)*, pages 339–344, Orlando, juin 1994. **58**

[Cohen91] L. COHEN. « On Active Contour Models and Balloons ». *Computer Vision, Graphics and Image Processing*, 53(4) :211–218, 1991. **109**

[Cohen96] L. D. COHEN et R. KIMMEL. « Global Minimum for Active Contour Models : A minimal Path Approach ». Dans *Proc of* *the Intern. Conf. on Computer Vision and Pattern Recognition*, 1996. **111**

[Collins86] S. M. Collins et D. J. Skorton, éditeurs. *Cardiac Imaging and Image Processing.* McGraw-Hill, 1986. **93**

[Corwin94] E. M. Corwin, A. M. Antonette, et W. C. Oldham. « An Iterative Method for Training Multilayer Networks with Threshold Functions ». *IEEE Transactions on Neural Networks*, 5(3) :507–508, mai 1994. **58**

[Cybenko89] G. Cybenko. « Approximations by superpositions of a sigmoidal function ». *Mathematics of Control, Signals and Systems*, 2(4) :303–314, 1989. **55**

[d'Alché-92] F. d'Alché-Buc, V. andrès, et J. P. Nadal. « Learning Fuzzy Control Rules with a Fuzzy Neural Network ». Dans *International Conf. on Artificial Neural Networks*, pages 715–719, 1992. **174**

[d'Alché-93] F. d'Alché-Buc. « *Modèles neuronaux et algorithmes constructifs pour l'apprentissage de règles de décision* ». PhD thesis, Université Paris XI, 1993. **54**

[d'Alché-94] F. d'Alché-Buc, D. Zwierski, et J. P. Nadal. « Trio-learning : a new strategy for building hybrid neural trees ». *International Journal of Neural Systems*, 5(4) :259–274, 1994. **54**, **174**

[de Angulo95] V. R. de Angulo et C. Torras. « On-Line Learning with Minimal Degradation in Feedforward Networks ». *IEEE Transactions on Neural Networks*, 6(3) :657–68, mai 1995. **62**

[de Figueir92] M. Teles de Figueiredo et J. M.-N. Leitao. « Bayesian Estimation of Ventricular Contours in Angiographic Images ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3) :416–429, septembre 1992. **112**, **114**, **114**, **150**

[Declerck96] J. Declerck, J. Feldmar, et N. Ayache. « Definition of a 4D Continuouse Polar Transformation for the Tracking and the Analysis of LV Motion ». Rapport Technique RR-3039, INRIA - Sophia Antipolis, novembre 1996. **113**

[Delyon93] Bernard Delyon. « Remarks on Filtering of Semi-Markov Data ». Rapport Technique 1986, INRIA, septembre 1993. **21**, **26**, **32**

[Dempster77] M. M. Dempster, N. M. Laird, et D. B. Rubin. « Maximun Likelihood from Incomplete Data via the EM Algorithm ». *Journal of Royal Statistical Society B*, 39 :1–38, 1977. **17**, **30**

[Denolin93] H. DENOLIN, Ph. COUMEL, J. P. BOURDARIAS, et A. LENAERS, éditeurs. *Méthode d'investigation en cardiologie*. Maloine, 1993. **87**, **213**

[Deriche90] R. DERICHE. « Fast Algorithms for Low-Level Vision ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1) :78–87, janvier 1990. **195**

[Deschamps99] T. DESCHAMPS, J.-M. LÉTANG, et L. COHEN. « Automatic Construction of Minimal Paths in 3D Images for Virtual Endoscopy ». Dans *Computer Assisted Radiology and Surgery (CARS'99)*, Paris, juin 1999. **111**

[Digalakis93] V. DIGALAKIS, J. R. ROHLICEK, et M. OSTENDORF. « ML Estimation of a Stochastic Linear System with the EM Algorithm and Its Application to Speech Recognition ». *IEEE Transaction on Speech and Audio Processing*, 1(4) :431–442, octobre 1993. **24**, **26**

[Dodge60] H. T. DODGE, H. SANDLER, D. W. BALLEW, et J. D. LORD. « The Use of Biplane Angiocardiography for the Measurement of Left Ventricular Volume in Man ». *Am. Heart J.*, 60(5) :762–76, novembre 1960. **92**, **92**, **93**

[Donnell98] T. O DONNELL et G. FUNKA-LEA. « 3-D Cardiac Volume Analysis using Magnetic Resonnace Imaging ». Dans *Proc. of 4th IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 240–241, Princeton, octobre 1998. **113**, **114**

[Dorst94] L. DORST et R. Van den BOOMGAARD. « Morphological signal processing and the slope transform ». *Signal Processing*, 38 :79–98, 1994. **192**, **192**

[Dubois16] D. DUBOIS et E. F. DUBOIS. « Estimation of Human Body Surface Area if Height and Weight be known ». *Arch Int Med*, pages 873–971, 1916. **94**

[Duda73] R. O. DUDA et P. E. HART. *Pattern Classification and Scene Analysis*. Wiley, 1973. **8**, **12**

[Dumay92] A.C.M. DUMAY. « *Image Reconstruction from biplane angiographic projections* ». PhD thesis, Technical University Delft, octobre 1992. **111**, **112**

[Fahlman90] S.E. FAHLMAN et C. LEBIERE. « The Cascade-Correlation Learning Architecture. ». Dans D.S. TOURETZKY, éditeur, *Advances in Neural Inforamtion Processing Systems II*, pages 524–532, 1990. **64**

[Fifer86] M. A. FIFER et W. GROSSMAN. « Measurement of Ventricular volumes, Ejection Fraction, Mass and Wall Stress ». Dans W. GROSSMAN, éditeur, *Cardiac Catheterization and Angiography*. Lea and Febiger, 1986. **83**

[Florack92] L. M.J. FLORACK, B. M. ter HARR ROMENY, J. J. KOENDERINK, et M. A. VIERGEVER. « Scale and the differential strcuture of images ». *Image and Vision Computing*, 10(6) :376–388, juillet 1992. **126**, **126**

[Forney73] G. D. FORNEY. « The Viterbi Algorithm ». *Proc. of the IEEE*, 61(3) :268–278, mars 1973. **21**

[Fraser93] A. M. FRASER et A. DIMITRIADIS. « Forecasting Probability Densities by using Hidden Markov Models with mixed States ». Dans A. S. WEIGEND et N. A. GERSHENFELD, éditeurs, *Times Series Prediction : Forecasting the Future and Understanding the Past*, pages 265–282, 1993. voir [Weigend93]. **30**

[Frean90] M. FREAN. « The Upstart Algorithm : A method for Constructing and Training Feedforward Neural Networks. ». *Neural Computation*, 2 :198–209, 1990. **63**

[Funahashi89] K. FUNAHASHI. « On the approximate realization of continuous mappings by Neural Networks ». *Neural Networks*, 2 :183–192, 1989. **55**

[Gallant86] S. I. GALLANT. « Three Constructive Algorithms for Network Learning ». *Proc. Ann. Conf. Sc. Soc.*, pages 652–60, 1986. **63**

[GD95] Garnier DELAMARE. *Dictionnaire des termes de médecine.* Maloine, 1995. 24ème dition. **85**

[Geiger95] D. GEIGER, A. GUPTA, L. A. COSTA, et J. VLONTZOS. « Dynamic Programming for Detecting, Tracking and Matching Deformable Contours ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3) :294–302, mars 1995. reprinted in [Singh98]. **110**, **113**, **114**

[Geman84] S. GEMAN et D. GEMAN. « Stochastic Relaxation, Gibbs Distribution and the Bayesian Restauration of Images ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 :721–41, 1984. **21**, **185**

[Gentric93] P. GENTRIC et H. WITHAGEN. « Constructive method for a new Classifier based on a Radial-Basis-Function Neural Network and accelerated by a Tree ». Dans *IWANN*, 1993. **56**, **64**

[Gérard95] O. Gérard. « Adaptive System for Smart Battery Management based on Neural Networks ». Rapport de DEA, Laforia, septembre 1995. **47**

[Gérard96] O. Gérard et J. N. Patillon. « Une procédure d'optimisation des pentes dans un réseau de neurones ». *VALGO*, 96(1), 1996. **2**, **51**, **56**

[Gérard97] O. Gérard, J.N. Patillon, et F. d'Alché Buc. « Neural Network Adaptive Modeling of Battery Discharge Behavior ». Dans W. Gerstner *et al*, éditeur, *International Conf. on Artificial Neural Networks, Lausanne*, volume 1327 de *Lecture Notes in Computer Science*, pages 1095–1100. Springer-Verlag, 1997. **2**, **38**, **51**, **178**

[Gérard98] O. Gérard et S. Markam-Ebeid. « Automatic Contour Detection by Encoding Knowledge into Active Contour Models ». *4th IEEE Workshop on Applications of Computer Vision, (WACV'98) Princeton, NJ*, pages 115–120, octobre 1998. **3**, **115**, **117**, **158**, **178**

[Gérard99a] O. Gérard, F. d'Alché Buc, S. Makram-Ebeid, P. Gallinari, et T. Artières. « Automatic Contour Extraction in Images using a 2-D Hidden Markov Model ». Dans *Intern. Conf. on Artificial Neural Networks (ICANN'99)*, IEE Conference Publication numero 470, pages 455–460, Edimburgh, septembre 1999. **117**, **150**, **178**

[Gérard99b] O. Gérard, F. d'Alché Buc, S. Makram-Ebeid, P. Gallinari, et T. Artières. « Modèle Markovien et programmation dynamique pour l'extraction de contours : application à des images médicales ». Dans *17ème Colloque sur le Traitement du Signal et des Images (GRETSI'99)*, pages 1205–1208, 1999. **116**

[Gérard99c] O. Gérard, J.N. Patillon, et F. d'Alché Buc. « Discharge Prediction of Rechargeable Batteries with Neural Network ». *Journal of Integrated Computer-Aided Engineering*, 6(1) :41–53, 1999. **2**, **3**, **51**, **115**, **178**

[Ghahramani96a] Z. Ghahramani et G. E. Hinton. « Parameter Estimation for Linear Dynamical Systems ». Rapport Technique CRG-TR-96-2, University of Toronto, février 1996. **27**, **32**

[Ghahramani96b] Z. Ghahramani et M. I. Jordan. « Factorial Hidden Markov Models ». A. I. Memo 1561, AI Lab, MIT, janvier 1996. ftp ://publications.ai.mit.edu. **30**

[Giordana97] N. Giordana et W. Pieczynski. « Estimation of Generalized Multisensor Hidden Markov Chains and Unsupervised

Image Segmentation ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5) :465–75, mai 1997. **183**

[Goutte96] C. Goutte et C. Ledoux. « Overview of Connectionist Control Using MLP ». Rapport Technique 96/15, LAFORIA, mai 1996. **45**, **62**

[Goutte97] C. Goutte. « *Statistical Learning and Regularisation in Regression : Application to system identification and time series modelling* ». PhD thesis, Université Paris 6, 1997. **9**

[Graffigne93] C. Graffigne et C. Labourdette. « Some Segmentation Algorithms and their parallel implementation ». Dans *49th session Bull. Int. Stat. Inst.*, pages 199–215, 1993. **183**

[Griffith74] R. L. Griffith, C. Grant, et H. Kaufman. « An Algorithm for Locating the Aortic Valve and the Apex in Left-Ventricular Angiocardiograms ». *IEEE Trans. on Biomed. Eng.*, 21 :345–349, 1974. **130**

[Groshong92] B. R. Groshong, L. A. Spero, et J. T. Cusma. « Maximum A Posteriori Optimization - A Method for Calculation of Dynamic Changes in Ventricular Contours from Angiographic Image Sequence ». Dans *IEEE Computers in Cardiology*, pages 355–358, Durham, octobre 1992. **112**, **114**

[Grossberg76] S. Grossberg. « Adaptive Pattern and Universal Recording II : Feedback Developpement and Coding of Neural Feature Detectors ». *Biological Cybernetics*, 23 :121–134, 1976. **61**, **64**

[Hall79] E. L. Hall. *Computer Image Processing and Recognition.* Computer Science and Applied Mathematics. Academic Press, 1979. **100**, **124**

[Hao94] J. Hao, J. Vandewalle, et S. Tan. « Predictive Control of Non-linear Systems based on Identification by Back-Propagation Networks ». *International Journal of Neural Systems*, 5(4) :335–44, décembre 1994. **62**

[He91] Y. He et A. Kundu. « 2-D Shape Classification using Hidden Markov Model ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11) :1172–1184, novembre 1991. **110**

[Hébert00] J.L. Hébert, D. Chemla, O. Gérard, K. Zamani, R. Frank, Y. Lecarpentier, et G. Fontaine. « Evaluation nouvelle de la dysplasie ventriculaire droite arythmogène (DVDA) par ventriculographie droite et gauche ». *Accepté pour les Xèmes journées européennes de la société Française de Cardiologie*, 2000. **96**

[Hertz91] J. Hertz, A. Krogh, et R. G. Palmer. *Introduction to the theory of Neural Computation.* Addison-Wesley, 1991. **55**

[Hornik89] K. Hornik, M. Stinchcombe, et H. White. « Multilayer Feedforward Networks are Universal Estimators ». *Neural Networks*, 2 :359–366, 1989. **52**, **55**

[Hough62] P. V. C. Hough. « A method and means for recognizing complex patterns ». U. S. Patent 3,069,654, 1962. **108**

[Hubel88] D. H. Hubel. *Eye, Brain, and Vision.* Scientific American Library, 1988. **124**, **124**

[Hunter93a] I. A. Hunter et H. Maître. « Use of Snakes to link Edge Points to Create Left Ventricular Boundaries in Echocardiographic Images ». Rapport Technique 7.345 Hunt, ENST, 1993. **113**, **114**

[Hunter93b] I. A. Hunter, J. J. Soraghan, et T. S. Durrani. « Detection of Echocardiographic Left Ventricle Boundaries Using Neural Networks ». Dans *IEEE Computers in Cardiology*, pages 201–4, Londres, septembre 1993. **113**, **114**, **114**

[Illingwort88] J. Illingworth et J. Kittler. « A survey of the Hough transform ». *Computer Vision, Graphics and Image Processing*, (44) :87–116, 1988. **108**

[Indejehago81] J.-P. Indejehagopian. « Estimation Récursive des paramètres d'un modèle de régression variant au cours du temps ». *R.A.I.R.O Recherche Opérationnelle/ Operations Research*, 15(1) :39–49, 1981. **75**

[Ingles75] N. B. Ingles, G. T. Daughers, E. B. Stinson, et E. L. Alderman. « Measurement of midwall myocardial dynamics in intact man by radiography of surgically implanted markers ». *Circulation*, 61 :859–867, 1975. **95**

[INSERM95] INSERM, éditeur. *Le cœur, fonctionnement dynamique et traitement.* Nathan, 1995. **84**

[Isard96] M. Isard et A. Blake. « Contour tracking by stochastic propagation of conditional density ». Dans *Proc. European Conf. Computer Vision*, pages 343–356, 1996. **175**

[Jordan86] M. Jordan. « Attractor Dynamics and Parallelism in a Connectionist Sequential Machine ». Dans *Proceedings of the Eigth Annual Conference of the Cognitive Science Society*, pages 531–546. Erlbaum, 1986. **46**

[Jordan94] M. I. Jordan et R. A. Jacobs. « Hierarchical Mixtures of Experts and the EM Algorithm ». *Neural Computation*, 6 :181–214, 1994. **49**

[Jordan95] M. I. JORDAN et R. A. JACOBS. « Modular and Hierarchical Learning Systems ». Dans M. ARBIB, éditeur, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995. **48**

[Jordan96] M. I. JORDAN, Z. GHAHRAMANI, et L. K. SAUL. « Hidden Markov Decision Trees ». Rapport Technique 9606, MIT Computational Cognitive Science, 1996. ftp ://psyche.mit.edu/pub/jordan. **76**

[Jordan99] M. I. JORDAN, éditeur. *Learning in Graphical Models*. MIT Press, 1999. **30**, **31**

[Kalman60] R. E. KALMAN. « A New Approach to Linear Filtering and Prediction Problems ». *Transaction of the ASME, Series D, Journal of Basic Engineering*, 82 :35–45, mars 1960. In [Sorenson85] pp. 16-26. **24**, **24**

[Karsch80] K. R. KARSCH, U. LAMM, H. BLANKE, et K. P. RENTROP. « Comparison of nineteen quantitative models for assessment of localized left ventricular wall motion abnormalities ». *Clinical Cardiology*, 3 :123–, 1980. **95**

[Kass87] M. KASS, A. WITKIN, et D. TERZOPOULOS. « Snakes : Active Contour Models ». Dans *Proc. Conf. on Computer Vision*, pages 259–268, 1987. **109**

[Kass88] M. KASS, A. WITKIN, et D. TERZOPOULOS. « Snakes : Active Contour Models ». *International Journal of Computer Vision*, pages 312–331, 1988. reprinted in [Singh98]. **109**

[Kègl98] B. KÈGL, A. KRZYZAK, T. LINDER, et K. ZEGER. « A Polygonal Line Algorithm for Constructing Principal Curves ». Dans *Neural Information Processing*, 1998. to be published. **174**, **174**

[Kennedy70] J. W. KENNEDY, S. E. TRENHOLME, et I. S. KASSER. « Left Ventricular Volume and Mass from single-plane Cineangiocardiogram. A Comparison of anteroposterior and Right anterior oblique Methods ». *American Heart Journal*, 80(3) :343–352, septembre 1970. **83**, **95**

[Koenderink84] J. KOENDERINK. « The structure of Images ». *Biological Cybernetics*, 50 :363–70, 1984. **125**

[Kohonen82] T. KOHONEN. « Self-organized formation of topologically correct feature maps ». *Biological Cybernetics*, 43 :59–69, 1982. **47**

[Kohonen89] T. KOHONEN. *Self-Organization and Associative Memory (3rd ed.)*. Springer-Verlag, 1989. **47**

[Kruschke91] J. K. Kruschke et J. R. Movellan. « Benefits of Gain : Speeded Learning and Minimal Hidden Layers in Back-Propagation Networks ». *IEEE Transactions on Systems, Man, and Cybernetics*, 21(1) :273–280, janvier 1991. **57**, **58**, **60**

[Laurent72] P.-J. Laurent. *Approximation et Optimisation*, volume 13 de *Enseignement des science*. Hermann, 1972. **53**

[LeCun87] Y. LeCun. « *Modèles Connexionnistes de l'Apprentissage* ». PhD thesis, Université Paris 6, 1987. **11**

[Legrand96] L. Legrand, F. Marzani, et L. Dusserre. « A Method for Automatically Detecting the Systole and Diastole Phases in Sequences of Angiographic Images ». *IEEE Symposium on Computer-Based Medical Systems*, pages 30–35, 1996. **84**

[Leray98] Ph. Leray. « *Apprentissage et Diagnostic de systèmes complexes : Réseaux de Neurones et Réseaux Bayésiens. Application à la gestion en temps réel du trafic téléphonique français* ». PhD thesis, Université Paris 6, septembre 1998. **63**, **133**, **133**, **147**

[Létang93] J.-M. Létang. « *Intégration temporelle et régularisation statistique appliquées la détection d'objets mobiles dans une séquence d'images* ». PhD thesis, INPG, octobre 1993. **26**

[Létang98] J.-M. Létang, M. Philip, et S. Makram-Ebeid. « Fast Post Processing Techniques for 3-D Angiography Volumes ». Dans *Computer Assisted Radiology and Surgery, 12th International Symposium (CARS'98)*, page 894, Tokyo, juin 1998. **186**, **195**

[Levin93] A. Levin et K. Narendra. « Control of Nonlinear Dynamical Systems Using Neural Networks–Controllability and Stabilization ». *IEEE Transactions on Neural Networks*, 4(2) :192–206, mars 1993. **45**

[Levin95] A. Levin et K. Narendra. « Identification Using Neural Networks ». *Neural Computation*, 7(2) :349–357, 1995. **45**, **46**

[Levin96] A. Levin et K. Narendra. « Control of Nonlinear Dynamical Systems Using Neural Networks-Part II : Observability, Identification, and Control ». *IEEE Transactions on Neural Networks*, 7(1) :30–42, janvier 1996. **46**, **62**

[Lindeberg90] T. Lindeberg. « Scale-Space for Discrete Signals ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3) :234–254, mars 1990. **125**

[Linden95] D. Linden, éditeur. *Handbook of batteries.* McGraw-Hill, NY, 1995. **39**

[Llacer93] J. LLACER, B. M. ter HAAR ROMENY, L. M.J. FLORACK, et M. A. VIERGEVER. « Representation of Medical Images by Visual Response Functions ». *IEEE Engineering in Medicine and Biology*, pages 40–47, mars 1993. **126**, **127**

[Looney96] C. G. LOONEY. « Stabilization and speedup of convergence in training feedforward neural networks ». *Neurocomputing*, 10 :7–31, 1996. **57**

[Maintz96] J.B. A. MAINTZ, P. A. van den ELSEN, et M. A. VIERGEVER. « Evaluation of Ridge Seeking Operators for Multimodality Medical Image Matching ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4) :353–365, avril 1996. **130**

[Maître85] H. MAÎTRE. « Un panorama de la transformation de Hough ». *Traitement du Signal*, 2(4), 1985. **108**

[Maître87a] H. MAÎTRE et Y. WU. « Combination of Dynamic Programming and Autoregressive Model for Elastic Matching of Distorted Images ». Dans *5^{th} Scandinavian Conf. on Image Analysis*, 1987. **110**, **160**

[Maître87b] H. MAÎTRE et Y. WU. « Improving Dynamic Programming to solve image registration ». *Pattern Recognition*, 20(4) :443–462, 1987. **109**, **110**, **150**, **160**

[Maître89] H. MAÎTRE et Y. WU. « A Dynamic Programming Algorithm for Elastic Registration of Distorted Pictures based on Autoregressive Model ». *IEEE Acoustics, Speech, and Signal Processing*, 37(2) :288–297, février 1989. **109**

[Makram-Ebe89] S. MAKRAM-EBEID, J.-A. SIRAT, et J.-R. VIALA. « A Rationalized Error Back-Propagation Learning Algorithm ». Dans *International Joint Conference on Neural Networks (IJCNN'89)*, pages 373–380, Washington, juin 1989. **57**, **58**

[Makram-Ebe93] S. MAKRAM-EBEID et J. BREITENSTEIN. « Computer Vision Tools Based on Neural Nets for Automatic Evaluation of the Left Ventricle Ejection Fraction ». *LEP Annual Review*, pages 48–50, 1993. **138**

[Makram-Ebe99] S. MAKRAM-EBEID et O. GÉRARD. « Image Processing method and System Involving Contour Detection Steps ». US Patent, 1999. **150**, **178**

[Mallat89] S. G. MALLAT. « A Theory for Multiresolution Signal Decomposition : the Wavelet Representation ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 :674–693, 1989. **105**

[Marchand90] M. Marchand. « A Growth Algorithm for Neural Network Decision Trees ». *Europhysics Letters*, 12 :105–109, 1990. **54**

[Marr82] D. Marr. *Vision : A Computational Investigation into the Human Representation and Processing of Visual Information.* W. H. Freeman, 1982. **124**

[Marroquin87] J. Marroquin, S. Mitter, et T. Poggio. « Probabilistic Solution of ill-posed Problems in Computational Vision ». *J. of the American Statistical Association*, 82 :76–89, 1987. **20**, **185**

[McInerney96] T. McInerney et D. Terzopoulos. « Deformable Models in Medical Image Analysis : a Survey ». *Medical Image Analysis*, 1(2) :91–108, 1996. **109**

[Mehra70] R. K. Mehra. « On the Identification of Variances and Adaptive Kalman Filtering ». *IEEE Trans. on Automatic Control*, AC(15) :175–184, avril 1970. In [Sorenson85]. **27**

[Meila95] M. Meila et M. I. Jordan. « Learning Fine Motion by Markov Mixtures of Experts ». A.I. Memo 1567, AI Lab, MIT, novembre 1995. ftp ://publications.ai.mit.edu. **48**

[Mézard89] M. Mézard et J.-P. Nadal. « Learning in Feedforward Layered Networks : The Tiling Algorithm ». *Journal of Physics*, A(22) :2191–2204, 1989. **63**

[Moore98] B. Moore. « ART 1 and Pattern Clustering ». *Proc. of the 1988 Connectionist Models Summer School*, pages 174–85, 1998. **64**

[Movellan87] J. R. Movellan. « Self-regulated Temperature in Back Propagation Networks ». Dans *Ninth Annual Berkeley-Stanford Conference*, 1987. **57**

[Narendra90] K. Narendra et K. Parthasarathy. « Identification and Control of Dynamical Systems using Neural Networks ». *IEEE Transactions on Neural Networks*, 1 :4–27, mars 1990. **45**, **62**

[Narendra95] K. Narendra, J. Balakrishnan, et M. Ciliz. « Adaptation and Learning using Multiple Models, Switching and Tuning ». *IEEE Control Systems*, pages 37–51, juin 1995. **62**

[Nerrand93] O. Nerrand, P. Roussel-Ragot, L. Personnaz, G. Dreyfus, et S. Marcos. « Neural Networks and Non-Linear Adaptive Filtering : Unifying Concepts and New Algorithms ». *Neural Computation*, 5(2) :165–197, 1993. **46**

[Notten84] P.H.L. NOTTEN. « *Rechargeable Nickel-Metalhydrid Batteries : a Successful New Concept* », volume 281, Chapitre 7. NATO ASI Series E, 1984. **39**

[Ollivier93] J. P. OLLIVIER, J. MONSEGU, et J. de BOURAYNE. « *Angiocardiographie du ventricule gauche* ». Maloine, 1993. In [Denolin93], chapitre 29. **84**

[Olstad96] B. OLSTAD et A. H. TORP. « Encoding of a priori Information in Active Contour Models ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9) :863–872, septembre 1996. **110**, **116**, **150**

[Park91a] D. C. PARK, M. A. EL-SHARKAWI, et R. J. MARKS. « An Adaptively Trained Neural Networks ». *IEEE Transactions on Neural Networks*, 2(3) :334–45, mai 1991. **61**

[Park91b] J. PARK et I. W. SANDBERG. « Universal Approximations using Radial-Basis-Function Networks ». *Neural Computation*, 3 :246–57, 1991. **65**

[Park96] J. PARK. « *Model-Based Shape and Motion Analysis : Left Ventricle of a Heart* ». PhD thesis, Dept. of Computer and Information Science, University of Pennsylvania, 1996. **113**, **114**

[Patillon96a] J. N. PATILLON, S. GOURRIER, O. GÉRARD, et R. EINERHAND. « Final Report on Smart Battery Management using ANN ». Rapport Technique C 96 - 632, LEP, 1996. (non disponible). **71**

[Patillon96b] J. N. PATILLON, S. GOURRIER, O. GÉRARD, et R. EINERHAND. « Report on Definition of a Database for Smart Battery Management ». Rapport Technique C 96 - 628, LEP, 1996. (non disponible). **74**

[Patillon96c] J.N. PATILLON, O. GÉRARD, F. d'Alché BUC, S. GOURRIER, et J.N. NADAL. « Smart Battery Management ». *1995 LEP Annual Review*, pages 52–54, 1996. **40**

[Patillon98] J.-N. PATILLON et O. GÉRARD. « System for monitoring charging/discharging cycles of a rechargeable battery, and host device including a smart battery ». United States Patent, 1998. Patent number : 5,825,156, date : Oct. 20, 1998. **178**

[Peng95] A. PENG et W. PIECZYNSKI. « Adaptive Mixture Estimation and Unsupervised Local Bayesian Image Segmentation ». *Graphical Models and Image Processing*, 57(5) :389–99, 1995. **183**, **185**

[Philip97] M. PHILIP. « Post processing for 3D Rotational Angiography Volumes ». Rapport Technique S 97-962, LEP, décembre 1997. **186**, **187**, **195**

[Pican94a] N. PICAN et F. ALEXANDRE. « Highly Adaptive Neural Networks for adaptive Neuro-Control : the OWE Architecture ». Dans *IEEE Systems, Man and Cybernetics, San Antonio*, 1994. **77**

[Pican94b] N. PICAN, J.C. FORT, et F. ALEXANDRE. « An On-line Learning Algorithm for the Orthogonal Weight Estimation of MLP ». *Neural Processing Letters*, 1(1) :21–24, 1994. **76**

[Pican95] N. PICAN. « *Approches Statique et Dynamique de la Modulation des efficacités Synaptiques dans les Réseaux de Neurones* ». PhD thesis, Univ. Nancy 1, 1995. **77**

[Pican96a] N. PICAN. « Intrinsic and Parallel Performances of the OWE Neural Network Architecture ». Dans *International Conference on Artificial Neural Networks*, pages 755–760, 1996. **76**

[Pican96b] N. PICAN, D. FOHR, et J.-F. MARI. « HMMs and OWE Neural Network for continuous Speech Recognition ». Dans *ICSLP Proceedings*, 1996. **77**

[Pieczynski94] W. PIECZYNSKI. « Champs de Markov cachés et estimation conditionnelle itérative ». *Traitement du Signal*, 11(2) :141–53, 1994. **16**, **30**, **183**, **185**

[Pope85] D. L. POPE, D. L. PARKER, P. D. CLAYTON, et D. E. GUSTAFSON. « Left Ventricular Border Recognition Using a Dynamic Search Algorithm ». *Radiology*, 155(2) :513–17, 1985. **112**, **114**, **114**, **172**

[Press92] W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, et B. P. FLANNERY. *Numerical Recipes in C*. Cambridge University Press, 2 édition, 1992. **53**

[Qian98] J.-Z. QIAN. « Can a Computer "see" the Beating Heart from "snow-storm" Images ». Dans *Proc. of 4th IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 280–281, Princeton, octobre 1998. **113**

[Quilan86] J.R. QUILAN. « Induction of decision trees ». *Machine Learning*, 1 :81–106, 1986. **54**

[Rabiner86] L.R. RABINER et B.H. JUANG. « An Introduction to Hidden Markov Models ». *IEEE Acoustics, Speech and Signal Processing Magazine*, 3 :4–16, janvier 1986. **28**, **30**

[Radix70] Jean-Claude RADIX. *Introduction au Filtrage Numérique*. Eyrolles, 1970. **23**, **24**

[Rauch65] H. E. Rauch, F. Tung, et C. T. Striebel. « Maximum Likelihood Estimates of Linear Dynamic Systems ». *Journal of the American Institute of Aeronautics and Astronautics*, 3 :1445–1450, août 1965. In [Sorenson85] pp. 243-248. **23**, **24**, **26**

[Reiber96] J. H.C. Reiber, A. R. Viddeleer, G. Koning, M. J. Schalij, et P. E. Lange. « Left Ventricular Regression Equations from Single Plane Cine and Digital X-ray Ventriculograms Revisited ». *International Journal of Cardiac Imaging*, 12 :68–78, 1996. **92**, **93**

[Richard91] M. D. Richard et R. P. Lippman. « Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities ». *Neural Computation*, 3(4) :464–483, 1991. **11**

[Rivals95] I. Rivals. « *Modélisation et commande de processus par réseaux de neurones ; application au pilotage d'un véhicule autonome* ». PhD thesis, Université Paris 6, 1995. **46**

[Robert92] C. Robert. *L'analyse statistique Bayésienne.* Economica, 1992. **12**

[Robert95] L. Robert, S. Canu, J.F. Lerallut, et Y. Maingourd. « Detection and Reconstruction of Cavity Contours in single frame 2D Echocardiography using periodic splines Approximation on a Roughly Pre-processed Image, and Automatic Tuning of the Parameters of the Model ». Dans *Proc. IEEE Engineering in Medicine and Biology*, 1995. **112**, **113**

[Rousselet83] M. Rousselet. *Mathématiques sur ZX 81 et Spectrum.* Poche Informatique n° 5. Editions Techniques et Scientifiques Françaises, 1983. **93**

[Rumelhert86] D. E. Rumelhert, G. E. Hinton, et R. J. Williams. « Learning Internal Representations by Error Propagation ». *Parallel Distributed Processing*, 1 :318–362, 1986. **45**

[Saad95a] D. Saad et S. A. Solla. « Dynamics of On-Line Gradient Descent Learning for Multilayer Neural Networks ». *Proc. of Neural Information Processing Systems (NIPS-8'95)*, 1995. **61**

[Saad95b] D. Saad et S. A. Solla. « Exact Solution for On-Line Learning in Multilayer Neural Networks ». *Physical Review Letter*, 74 :4337–4340, 1995. **61**

[Salzenstei98] F. Salzenstein et W. Pieczynski. « Sur le choix de Méthode de segmentation statistique d'images ». *Traitement du Signal*, 15(2) :119–27, 1998. **183**

[Saporta90] G. SAPORTA. *Probabilités Analyse des données et statistique.* Technip, 1990. **9**, **148**

[Schmitt97] F. SCHMITT. « Psychophysique de la vision ». Support de cours. Télécom Paris, Dpt Images., 1997. **124**

[Sethi90] I. K. SETHI. « Entropy Nets : From Decision Trees to Neural Networks ». *Proc. of the IEEE*, 78(10) :1605–1613, octobre 1990. **54**

[Sethi95] I. K. SETHI. « Neural Implementation of Tree Classifiers ». *IEEE Trans. on Systems, Man, and Cybernetics*, 25(8) :1243–1249, août 1995. **54**

[Sethian96] J. A. SETHIAN. *Level Set Methods.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1996. **111**

[Sheehan86] F. SHEEHAN et *al.*. « Advantages and Applications of the Centerline Method for Characterizing Regional Ventricular Functions ». *Circulation*, 74(2) :293–305, août 1986. **95**, **172**

[Sheehan98] F. SHEEHAN et *al.*. « Quantitative Three dimensional Echocardiography : Methodology, Validation, and Clinical Applications ». Dans *Medical Image Computing and Computer-Assisted Intervertion*, volume 1496 de *Lecture Notes in Computer Science*, pages 102–109. Springer, 1998. **113**

[Shumway82] R. H. SHUMWAY et D. S. STOFFER. « An Approach to Time Series Smoothing and Forecasting using the EM Algorithm ». *J. Time Series Analysis*, 3(4) :253–264, 1982. **26**, **27**

[Silverman85] B. W. SILVERMAN. *Density Estimation for Statistics and Data Analysis.* Monographs on Statistics and Applied Probability 26. Chapman and Hall, 1985. **9**

[Singh98] A. SINGH, D. GOLDGOF, et D. TERZOPOULOS, éditeurs. *Deformable Models in Medical Image Analysis.* IEEE, 1998. **109**, **199**, **205**, **209**

[Sirat90] J. SIRAT et J.-P. NADAL. « Neural trees : a new tool for classification ». *Network*, 1 :198–209, 1990. Snowbird 1990. **54**

[Solla95] S. A. SOLLA et D. SAAD. « On-Line Learning of Realizable Tasks ». Dans *International Conference on Neural Networks*, pages 383–388, 1995. **61**

[Solla96] S. A. SOLLA. « The Dynamics of Learning from Examples ». Inter. Conf. on Artificial Neural Networks (ICANN'96) Tutorial, Bochum, juillet 1996. **61**

[Sorenson85] H. W. Sorenson, éditeur. *Kalman Filtering : Theory and Application.* IEEE Press, 1985. **24**, **209**, **212**, **215**

[Stearns88] S. D. Stearns et R. A. David. *Signal Processing Algorithms.* Prentice-Hall, 1988. **45**

[Stévenin96] A. Stévenin-Barbier. « *Réseaux Connexionnistes et traitement symbolique-numérique : une application en compréhension du langage* ». PhD thesis, Université Paris 6, 1996. **29**, **46**

[Tehrani91] S. Tehrani, T. E. Weymouth, et G. B.J. Mancini. « Model Generation and Partial Matching of Left Ventricular Boundaries ». Dans *Medical Imaging V*, volume 1445, pages 434–45. SPIE, 1991. **112**, **114**

[ter Haar R96] B. M. ter Haar Romeny. « Introduction to Scale-Space Theory : Multiscale Geometric Image Analysis ». Rapport Technique ICU-96-21, Utrecht University, septembre 1996. **126**, **126**

[Thimm96] G. Thimm, P. Moerland, et E. Fiesler. « The Interchangeability of Learning Rate and Gain in Backpropagation Neural Networks ». *Neural Computation*, 8 :451–460, 1996. **60**

[v. Leeuwen86] P.J. v. Leeuwen et J.H.C. Reiber. « Automated detection of left ventricular boundaries from 35mm contrast cineangiograms ». *Signal Processing III*, pages 1409–1412, 1986. **111**, **111**, **114**

[Vaerman97a] V. Vaerman. « Unsupervised Segmentation of the Human Heart using Deformable Contours ». Dans *Proc. of World Congress on Medical Physics and Biomedical Engineering*, volume 35, page 634, 1997. **113**, **114**

[Vaerman97b] V. Vaerman, C. de Solà Fàbregas, et G. Menegaz. « The Hyperquadrics : an efficient Parametric Surface Representation ». Rapport Technique LTS 97.10, Signal Processing Lab, Swiss Federal Institute of Technology, Lausanne, novembre 1997. **108**

[van der Zw92] PNJ van der Zwet, G. Koning, et J.H.C. Reiber. « Left Ventricular Contour Detection : A Fully Automated Approach ». Dans *IEEE Computers in Cardiology*, pages 359–362, Durham, octobre 1992. **111**, **114**

[van Vliet98] LJ. van Vliet, I.T. Young, et P.W. Verbeek. « Recursive Gaussian Derivative Filters ». Dans A.K. Jain, S. Venkatesh, et B.C. Lovell, éditeurs, *Proc. 14th Int. Conf. on Pattern Recognition (ICPR)*, pages 509–514. IEEE Computer Society Press, 1998. **198**

[Vaucher96] G. VAUCHER. « Neuro-biological bases for spatio-temporal data coding in artificial neural networks ». Dans *ICANN'96*, volume 1112 de *Lectures Notes in Computer Science*, pages 703–708. Springer, 1996. **46**

[vdB94] R. van den BOOMGAARD et A. SMEULDERS. « The Morphological Structure of Images : The Differential Equations of Morphological Scale-Space ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11) :1101–13, novembre 1994. **106**, **191**

[vdB96] R. van den BOOMGAARD, L. DORST, S. MARKAM-EBEID, et J. SCHAVEMAKER. « Quadratic Structuring Functions in Mathematical Morphology ». Dans P. MARAGOS, R.W. SCHAFER, et M.A. BUTT, éditeurs, *International Symposium on Mathematical Morphology*, pages 147–154. Kluwer Academic Publisher, 1996. **147**, **193**

[Vincent91] L. VINCENT et P. SOILLE. « Watersheds in Digital Spaces : An Efficient Algorithm Based on Immersion Simulations ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6) :583–598, juillet 1991. **186**

[Viterbi67] A. J. VITERBI. « Error Bounds for convolutional codes and an asymptotically optimum decoding algorithm ». *IEEE Trans. Information Theory*, IT-13 :260–269, 1967. **21**

[Wan93] E. WAN. « *Times Series Prediction by using a Connectionist Network with delay lines* », pages 195–207. Addison-Wesley, 1993. in [Weigend93]. **46**

[Weigend93] A. S. WEIGEND et N. A. GERSHENFELD, éditeurs. *Time Series Prediction : Forecasting the Future and Understanding the Past*, volume XV de *SFI Studies in the Science of Complexity*. Addison-Wesley, 1993. **44**, **205**, **218**

[Weigend95] A. S. WEIGEND et M. MANGEAS. « Nonlinear Gated Experts for Time Series : Discovering Regimes and Avoiding Overfitting ». *International Journal of Neural Systems*, 6 :373–399, 1995. ftp ://ftp.cs.colorado.edu/pub/Time-Series/MyPapers/experts.ps.Z. **48**

[Welch95] G. WELCH et G. BISHOP. « An Introduction to the Kalman Filter ». Rapport Technique TR 95-041, University of North Carolina at Chapel Hill, 1995. ftp ://ftp.cs.unc.edu/pub/publications/techreports/95-041.ps.Z. **27**

[West85] J. B. WEST, éditeur. *Best and Taylor's Physiological Basis of Medical Practice*. Williams & Wilkins, 11 édition, 1985. **82**, **83**

[Widmer97] G. WIDMER. « Tracking Context Changes through Meta-Learning ». *Machine Learning*, 27 :259–286, 1997. **78**

[Williams89] R. WILLIAMS et D. ZIPSER. « Experimental Analysis of the real-time recurrent learning algorithm ». *Connection Science*, pages 87–111, 1989. **46**

[Witkin83] A. P. WITKIN. « Scale-space filtering ». *Proc. International Joint Conference of Artificial Intelligence*, pages 1019–22, 1983. **125**

[Wong86] W.-H. WONG, R. L. KIRKEEIDE, et K. L. GOULD. « Computer Applications in Angiography ». Dans S. M COLLINS et D. J. SKORTON, éditeurs, *Cardiac Imaging and Image Processing*, pages 206–39. McGraw-Hill, 1986. **93**, **95**

[Yamada88] H. YAMADA, C. MERRITT, et T. KASVAND. « Recognition of Kidney Glomerulus by Dynamic Programming Matching Method ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5) :731–737, septembre 1988. **110**

[Young98] A. A. YOUNG. « Model Tags : Direct 3D Tracking of Heart Wall Motion from Tagged MR Images ». Dans *Medical Image Computing and Computer-Assisted Intervertion*, volume 1496 de *Lecture Notes in Computer Science*, pages 92–101. Springer, 1998. **113**

[Yu94] X. YU, N. K. LOH, et W. C. MILLER. « Training Hard-Limiting Neurons Using Back-Propagation Algorithm By Updating Steepness Factors ». Dans *International Conference on Neural Networks (ICNN'94)*, pages 526–530, Orlando, juin 1994. **58**

[Yuille92] A. L. YUILLE, P. W. HALLIAN, et D. S. COHEN. « Feature Extraction from Faces using Deformable Templates ». *International Journal of Computer Vision*, 8(2) :99–111, 1992. **108**

[Ziegler92] U. M. ZIEGLER, L. W. HAWKES, et R. C. LACHER. « Rapid Learning with Large Weight Changes and Plasticity ». *Proc. of Inter. Joint Conf. on Neural Networks, Baltimore (IJCNN'92)*, 2 :146–151, juin 1992. **61**, **62**

[Zurada93] J. M. ZURADA. « Lambda Learning Rule for Feedforward Neural Networks ». Dans *International Conference on Neural Networks (ICNN'93)*, pages 1808–1811, San Fransisco, mars 1993. **57**

Index

Cet index ne contient que les références aux pages les plus importantes. Celles notées en gras renvoient à la définition du terme. Cet index peut être également utilisé comme liste de sigles et comme index des personnes citées.