N° d'ordre : 4183

Université des Sciences et Technologies de Lille
Laboratoire Paul Painlevé

MODÉLISATION ET ESTIMATION STATISTIQUE POUR L'IMAGERIE MÉDICALE : APPLICATION À LA DÉTECTION D'AMERS

Camille IZARD

THÈSE

présentée pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ

Spécialité : **Mathématiques Appliquées**

soutenue publiquement le 21 mai 2008 devant le jury composé de :

Pr. Jean-Louis BON	Université des Sciences et Technologies de Lille Polytech'Lille	Directeur
Pr. Mohamed DAOUDI	Telecom Lille 1	Président
Dr. Bruno JEDYNAK	Université des Sciences et Technologies de Lille Johns Hopkins University, Baltimore, MD	Co-Directeur
Pr. Nikos PARAGIOS	École Centrale de Paris, Chatenay-Malabry	Examinateur
Pr. Anuj SRIVASTAVA	Florida State University, Tallahassee, FL	Rapporteur
Pr. Alain TROUVÉ	École Normale Supérieure, Cachan	Rapporteur

à René,

à mes parents,

"Quand on EM, on ne compte pas."

Après avoir apprivoisé chevaux et vaches au cours de ma vie, je me suis tournée à l'automne 2004 vers la chasse à l'hippocampe qui s'est averée venir de paire avec la chasse aux bugs en tout genre. Je voudrais remercier tout ceux qui m'ont accompagnée au cours de cette aventure.

Il y a eu Bruno Jedynak qui s'est attelé à la tâche ardue de transformer une ingénieur agronome en mathématicienne appliquée. Appliquée j'ai essayé de l'être dès le début mais mathématicienne ą aura mis plus de temps à venir. J'ai d'ailleurs bien cru que la transformation n'aurait pas lieu mais avec une bonne dose de patience et de persévérance, Bruno m'a montré le chemin à emprunter. Je remercie aussi Jean-Louis Bon qui m'a toujours soutenue de faǫn inconditionnelle et notamment à chacune de mes visites à Lille. Sans son aide cette thèse n'aurait pas eu lieu. Grâce à lui je suis arrivée à bon port.

Mes remerciements s'adressent aussi à Craig Starck qui non seulement a fourni les images utilisées pour ce travail mais qui a aussi pris le temps de les marquer précisément et qui m'a appris à mettre un nom sur les différentes régions du cerveau. Je remercie tous les membres du CIS pour leur soutien tout au long de ces années et notamment Mike Miller pour sa confiance, qui, en m'invitant à faire partie du CIS, m'a permis d'être dans un environnement particulièrement porteur et où j'ai beaucoup appris du point de vue scientifique comme du point de vue humain. Enfin je remercie tout spécialement Elliot McVeigh grâce à qui j'ai pu terminer ma thèse en toute sérénité malgré les lois d'immigration américaines qui ont bien failli me jouer des tours. Je remercie également les rapporteurs de ma thèse et les membres du jury d'avoir pris le temps de lire ma thèse et de la commenter, sans prêter gare aux imperfections linguistiques.

Je remercie mes parents de leur soutien tout au long du parcours, ils m'ont aidée à prendre mon envol pour me laisser surprendre par les joies de la vie. Je remercie tous mes amis qui vivent à l'est de l'Atlantique de ne pas m'avoir oubliée malgré la distance et de penser à moi lorsque le parcours devenait plus chaotique, de me réassurer de leur amitié à chacun de mes passages en France. Je pense à tous ceux qui sont là-bas et tout particuliérement à Sophie et à Magali. Je remercie aussi tout particulièrement ceux qui ont partagé ma vie quotidienne au labo au cours de ces 4 années, avec qui j'ai partagé de nombreux repas et avec qui nous avons refait autour d'un café ou au cours d'un vendredi après-midi, l'Europe, les US, 100 fois au moins, à qui j'ai pu étaler la France en long en large et en travers pour pallier à mon mal du pays. Francisco et Stéphanie ont été de réguliers complices.

Enfin je remercie Dr. Vidal de m'avoir éperonnée de ces conseils piquants mais toujours pertinents, de m'avoir poussée à aller plus loin lorsque le courage ou l'envie me manquait. J'admire sa passion et son infatigable curiosité. Je remercie aussi René pour son amour, sa tendresse et sa patience au quotidien. Merci d'avoir été présent à mes cotés tout au long du parcours pour partager mes espoirs, mes peines et mes réussites. Merci d'avoir fait preuve d'une patience infinie au cours de la longue et douloureuse rédaction de ma thèse, de me faire sourire malgré tout dans les moments difficiles et de partager avec moi la joie et la fierté d'avoir menée à bout cette aventure.

Camille

TABLE OF CONTENTS

Table of Contents i

Synopsis 1

1 Introduction 9
- 1.1 Current Trends in Medical Imaging . . . 9
- 1.2 The Landmark Detection Problem . . . 10
  - 1.2.1 Landmarks as Salient Points . . . 10
  - 1.2.2 Geometrical Landmarks . . . 11
  - 1.2.3 Anatomical Landmarks . . . 12
- 1.3 Landmark-based Analysis . . . 13
- 1.4 Thesis Outline and Contribution . . . 13

2 Statistical Modeling for Image Analysis 15
- 2.1 Statistical Learning for Image Analysis . . . 15
  - 2.1.1 Probability and Image Space . . . 15
  - 2.1.2 Statistical Learning and Prediction . . . 17
- 2.2 Statistical Representation of a Set of Images . . . 18
  - 2.2.1 Principal Component Analysis and Active Appearance Models . . . 18
  - 2.2.2 Groupwise Registration and Deformable Templates . . . 19
  - 2.2.3 Registration by Energy Minimization . . . 21
- 2.3 Landmark Detection as a Local Registration Problem . . . 23
  - 2.3.1 Choice of the Deformation Model . . . 24
  - 2.3.2 Different Spline Models . . . 27
- 2.4 Model Estimation . . . 29
  - 2.4.1 Estimation of a Model with Missing Variables . . . 33
  - 2.4.2 Template Estimation in the case of Landmark Detection . . . 34

3 Deformable Intensity Model 37
- 3.1 Previous Work: Image Registration . . . 37
  - 3.1.1 Landmark Detection . . . 38
- 3.2 Deformable Intensity Model . . . 39
- 3.3 Model Selection using a Training Set . . . 40
  - 3.3.1 Direct Estimation of the Deformable Model . . . 40
  - 3.3.2 Learning the Distribution of the Landmark Locations . . . 42
- 3.4 Local Intensity Matching for Landmark Detection . . . 44
- 3.5 Numerical Validity of the Approximations . . . 46
- 3.6 Detection Results . . . 48
  - 3.6.1 Description of the Images . . . 48

6 Deformable Object for Medical Imaging **107**
6.1 Deformable Object Model . . . 107
6.1.1 Model Description . . . 108
6.1.2 Choice of the Deformation . . . 108
6.1.3 A Toy Example . . . 109
6.2 Deformable Intensity Object . . . 112
6.2.1 Image Likelihood . . . 112
6.2.2 Model Estimation . . . 113
6.2.3 Choice of the Partition . . . 115
6.2.4 Landmark Detection . . . 115
6.3 Tissue-based Deformable Intensity Object . . . 117
6.3.1 Image Likelihood . . . 117
6.3.2 Model Estimation . . . 118
6.3.3 Landmark Detection . . . 119
6.4 Experiments . . . 121
6.5 Chapter Conclusion . . . 124

7 Conclusion **127**
7.1 Summary . . . 127
7.2 Assessing the Performance of the Algorithms . . . 128
7.3 Other Applications for Medical Imaging . . . 129

Bibliography **131**

3.6.2 Detection in Brain Magnetic Resonance Images . . . 50
3.6.3 Choice of the Kernel . . . 53
3.7 Chapter Conclusion . . . 54

4 Deformable Edge Model **55**
4.1 The Deformable Edge Model . . . 55
4.1.1 Edge Detection by Directional Intensity Comparison . . . 58
4.2 Model Selection . . . 60
4.2.1 Global Estimation by the EM Algorithm . . . 61
4.2.2 Details of the E-step . . . 62
4.2.3 Details of the Noise Parameter Estimation . . . 63
4.2.4 Details of the Edge Template Estimation . . . 64
4.3 Landmark Detection by Local Edge Matching . . . 65
4.4 Deformable Edge Model with Image-specific Noise Parameters . . . 66
4.4.1 Model Estimation . . . 66
4.4.2 Landmark Detection . . . 68
4.5 Detection Results . . . 70
4.5.1 Synthetic Experiment . . . 70
4.5.2 Detection of a Landmark in Real Images . . . 74
4.6 Chapter Conclusion . . . 75

5 Tissue-based Deformable Intensity Model **77**
5.1 Previous Work: Image Segmentation . . . 77
5.2 A Complete Generative Model . . . 79
5.2.1 The Generative Model . . . 79
5.2.2 Deformable Tissue Model . . . 81
5.2.3 Photometric Model . . . 82
5.2.4 Prior on the Landmark Locations . . . 83
5.2.5 Prior on the Photometry . . . 83
5.3 Model Selection . . . 83
5.3.1 Complete Model Estimation by the EM Algorithm . . . 83
5.4 Prediction of the Landmark Location . . . 87
5.4.1 Combining Segmentation and Registration . . . 88
5.5 Image-specific Photometric Parameters . . . 90
5.5.1 Parameter versus Hidden Variable . . . 90
5.5.2 Model Estimation by the EM Algorithm . . . 91
5.5.3 Landmark Detection . . . 92
5.6 Decoupling Photometry and Geometry . . . 93
5.6.1 Model Description . . . 95
5.6.2 Model Selection . . . 95
5.6.3 Landmark Detection . . . 97
5.7 Experiments . . . 98
5.7.1 Template Estimation . . . 99
5.7.2 Detection Performance . . . 101
5.7.3 Combining Registration and Segmentation . . . 103
5.7.4 Choice of the Parameters . . . 104
5.8 Chapter Conclusion . . . 105

SYNOPSIS

Cette préface résume en français les idées développées en détail dans les différents chapitres de cette thèse écrite en anglais. Nous reprenons ici la structure principale de la thèse et le raisonnement global, mais invitons le lecteur à consulter les chapitres correspondants pour ce qui est de la formulation mathématique du problème et la présentation détaillée des solutions proposées.

Chapitre 1 : Introduction

Les progrès en matière d'acquisition d'images médicales permettent dorénavant d'obtenir des images à fine résolution de l'anatomie humaine. L'interprétation quantitative de ces images passe souvent par de nombreuses étapes d'annotation manuelle, que ce soit pour segmenter les structures d'intérêt ou pour localiser la position de certains amers dans l'image. Non seulement cela représente une charge de travail importante qui limite la taille des études comparatives, mais c'est aussi une source de variation et d'erreur non négligeable. La mise au point de méthodes automatiques est indispensable en vue de la généralisation de l'analyse quantitative des images, d'une part, et en vue de l'obtention de résultats statistiquement comparables, d'autre part. Les domaines d'applications des méthodes d'analyse quantitative automatiques sont multiples en médecine, notamment pour la conception de systèmes d'aide au diagnostique.

L'imagerie médicale est aussi un outil unique pour l'étude de l'anatomie humaine et pour la recherche biomédicale en général. L'anatomie numérique est une nouvelle discipline qui porte principalement sur l'étude les formes anatomiques à partir d'images médicales. L'objectif est de construire un atlas qui représente l'anatomie moyenne et ses variations pour une population donnée. C'est aussi d'identifier et de quantifier les différentes sources de variations de l'anatomie. En effet, des corrélations importantes entre la forme de certains organes et leur fonction ont pu être mises en évidence. L'étude du cœur, par exemple, a permis de corréler l'affinement de la paroi du ventricule gauche et le risque de maladie cardiovasculaire. Pour ce qui est du cerveau, certains changements de la forme de l'hippocampe caractérisent les stades précoces de la maladie d'Alzheimer. Comme bien d'autres sources de variation, l'effet du vieillissement est aussi observable sur les structures du cerveau. Un des objectifs de l'anatomie numérique consiste à distinguer l'évolution normale de l'anatomie au cours du développement et du vieillissement, d'un changement dû à une maladie. De nombreuses applications existent dans le domaine de la médecine, notamment en matière de diagnostique précoce de certaines maladies, avant l'apparition des premiers symptômes cliniques.

L'anatomie numérique est basée sur la comparaison quantitative des structures anatomiques qui passe le plus souvent par la mise en correspondance d'images médicales.

De nombreuses méthodes de comparaison d'images utilisent des amers pour définir les correspondances biologiques entre images. En général ces amers sont placés à la main dans les images. Il serait préférable d'utiliser des méthodes automatiques en vue d'améliorer la rapidité et la reproductibilité de la localisation des amers. Dans cette thèse, nous nous intéresserons plus particulièrement à ce problème de détection d'amers.

Il existe plusieurs définitions pour les amers en imagerie. Dans le domaine de la vision par ordinateur, on définit un amer comme un point particulièrement remarquable dans l'image. L'amer doit être détectable dans des conditions d'imagerie très variables : changements de point de vue, changements des conditions d'illumination. On peut aussi définir un amer comme un point de l'image qui caractérise la géométrie locale comme, par exemple, l'extrémité d'une structure, un maximum de courbure, un point selle. Cependant lorsqu'il s'agit de faire du recalage d'images médicales, il n'est pas clair qu'un maximum de courbure dans une image corresponde biologiquement à un maximum de courbure dans une autre image. On définira donc un amer anatomique comme le point dans l'image qui correspond à un point spécifique de l'anatomie. Contrairement aux deux définitions précédentes, l'amer est défini sur l'anatomie et observé dans l'image. Dans ce cas-là, les correspondances entre amers définissent des correspondances anatomiques.

Quelques méthodes ont été proposées pour la détection d'amers. En général il s'agit de détecter des amers géométriques. Lorsque cet amer est aussi un amer anatomique, ces méthodes peuvent être utilisées pour la détection d'amer anatomique, mais ce n'est pas le cas en général. De plus les méthodes proposées utilisent des connaissances a priori sur le type d'amer recherché. Donc pour chaque amer, il faut connaître la géométrie à laquelle l'amer correspond et, pour chaque type d'amer, il faut construire un modèle différent. Enfin ces méthodes détectent en général les amers un à un.

Nous proposons dans ce travail une méthode générale pour la détection d'amer anatomique dans des images médicales. À partir d'un échantillon d'entraînement constitué d'images dans lesquelles la position des amers a été préalablement marquée par un neurologue, on estime un modèle statistique de l'image autour des amers. Ce modèle peut être ensuite utilisé pour la détection des mêmes amers dans de nouvelles images. Grâce à cette méthode d'apprentissage, il est possible de construire un modèle qui s'adapte automatiquement à tout type d'amers mais aussi à un nombre variable d'amers. La méthode proposée est simple et générique. Nous présentons les performances obtenue pour la détection d'amers sur des images à résonance magnétique du cerveau.

Chapitre 2 : Fondements Mathématiques

Dans le premier chapitre, nous présentons les fondements mathématiques pour l'analyse d'images par patron déformable. Nous montrons comment cette approche peut-être utilisée pour la détection d'amers.

On définit une image sur une grille finie de $\mathbb{R}^d$, qui attribue à chaque nœud de la grille une valeur de $\mathbb{R}$, l'intensité de l'image. D'après cette définition, une image est un vecteur de $\mathbb{R}^S$ si S est le nombre de nœuds de la grille ou le nombre de pixels. Les images vivent dans un espace non-vectoriel de grande dimension. En effet, la moyenne de deux images,

ne peut pas être définie comme la moyenne Euclidienne des vecteurs de $\mathbb{R}^d$ correspondants. Par exemple, si une image diffère de l'image de référence par une translation, la moyenne Euclidienne des deux images est floue. Une bonne notion de moyenne correspondrait dans cet exemple à la translation de l'image de référence dans la direction de l'autre image.

C'est pourtant sur cette hypothèse que se base la plupart des modèles statistiques actuels pour l'analyse d'image. En faisant l'hypothèse que les images vivent dans un espace vectoriel, on représente un ensemble d'images par un modèle linéaire appris par Analyse en Composantes Principales (ACP). Les modes de variations des images sont représentés par les vecteurs propres de la matrice des corrélations des intensités. En supposant que l'espace sous-jacent est vectoriel, le modèle permet localement, autour de la moyenne, une bonne approximation de l'ensemble des images. Cependant la qualité de cette représentation diminue nettement lorsque la distance à la moyenne augmente. Pour autant, ce modèle très simple a permis d'obtenir de bons résultats pratiques pour la segmentation et le recalage d'images médicales.

Les travaux de Grenander sur les prototypes déformables ont ouvert la voie vers un autre type de représentation d'un ensemble d'images. Dans cette approche, chaque image est modélisée comme le résultat d'une déformation aléatoire agissant sur un prototype commun. On définit l'action d'une déformation f sur une image x, $f \cdot x$, par $x \circ f^{-1}$, ce qui signifie que la déformation agit sur le support de l'image. Le problème qui consiste à trouver l'ensemble des déformations d'un prototype vers un ensemble d'images est appelé alignement ou recalage d'images (ou encore "registration" en anglais). Le plus souvent ce problème est écrit sous la forme d'une fonction de coût à minimiser. Cette fonction de coût comporte un terme d'attachement aux données, un terme de régularisation et un coefficient qui équilibre les deux termes précédents. De nombreux choix plus ou moins arbitraires ont été explorés pour chacun de ces termes, ce qui a permis de proposer une multitude d'algorithmes sans pour autant proposer une solution optimale la plupart du temps.

Il est possible de reformuler le problème de détection des amers anatomiques comme un problème de recalage partiel des images. En effet, si on limite le groupe des déformations de sorte qu'il existe une bijection entre l'ensemble des déformations et l'ensemble des configurations possibles des amers, il est alors équivalent d'optimiser la fonction de coût par rapport à la position des amers ou par rapport aux paramètres de la déformation. Parmi les nombreux modèles de déformation possibles, nous choisissons de travailler sur des déformations paramétrées par le déplacement des amers et interpolées au reste de l'image par des splines. Ce modèle s'adapte facilement à un nombre d'amers variable. Le choix de la fonction d'interpolation permet de varier grandement la nature de la déformation obtenue sans modifier son expression générale, ce qui nous permet de proposer des algorithmes qui ne dépendent pas du choix du type de déformation. En pratique, pour limiter les temps de calcul, on utilisera des modèles de déformation locale telles que les splines Gaussiennes.

Nous proposons une approche statistique pour la modélisation d'images, qui consiste à modéliser la loi jointe des intensités de l'image et des variables cachées, les amers, en utilisant un modèle à prototype déformable. Cela nous permet de dériver des algorithmes

optimaux en utilisant des méthodes de maximisation de vraisemblance. Les hypothèses du modèle sont donc automatiquement prises en compte dans l'algorithme. Dans les chapitres suivants de la thèse, nous présentons une famille de modèles génératifs. La complexité du modèle et donc des algorithmes dérivés augmente au cours des chapitres pour obtenir des modèles qui expliquent mieux les données réelles.

Chapitre 3 : Modèle Déformable pour les Intensités

Dans ce chapitre, on commence par présenter d'un point de vue statistique la méthode habituellement utilisée pour le recalage d'images par comparaison d'intensité. On peut montrer que cette méthode est optimale pour recaler deux images qui diffèrent par un bruit Gaussien de variance fixe. Le prototype, dans ce cas-là, est aussi une image en niveau de gris. Trouver la correspondance entre les deux images revient à trouver la déformation spatiale qui met en correspondance le mieux possible le prototype et l'image, c'est-à-dire qui minimise la somme des carrés des différences d'intensité.

On propose dans le Chapitre 3 de travailler sur un modèle similaire. L'intensité à chaque pixel d'une image est modélisée par une variable aléatoire. Cette variable aléatoire suit une loi Gaussienne dont les paramètres sont donnés par un prototype probabiliste. La position des amers caractérise la déformation spatiale du prototype vers l'image qui permet d'assigner à chaque pixel la loi d'intensité correspondante. En pratique, cela signifie que l'intensité de l'image est comparée à l'intensité du prototype comme dans le modèle précédent, mais cette fois, la variance dépend du pixel. La vraisemblance d'une image dépend donc de la position des amers puisque ceux-ci sont utilisés pour paramétrer la déformation du prototype vers l'image. L'apprentissage de ce modèle est très simple, puisqu'il suffit d'estimer pour chaque pixel du prototype les paramètres d'une distribution Gaussienne à partir des images d'entraînements dans lesquelles la position des amers est donnée. Pour ce qui est du test, c'est-à-dire lorsqu'il s'agit de localiser la position des amers dans une nouvelle image, il suffit d'optimiser la vraisemblance par rapport à la position des amers. Le gradient de la vraisemblance peut être écrit de façon analytique, si bien que la fonction est optimisée par une montée de gradient.

L'algorithme de détection d'amers par comparaison d'intensité (Deformable Intensity Model DIM) est testé sur une série d'images à résonance magnétique du cerveau. Un neurologue a marqué à la main la position des amers. L'algorithme est entraîné sur les 2/3 des images disponibles et testé sur le tiers restant. Les performances sont mesurées par la distance Euclidienne entre la position prédite par l'algorithme automatique et la position identifiée par l'expert. La précision de la prédiction varie entre 1 et 2 mm en fonction de l'amer. Les amers qui se situent dans des régions relativement homogènes en intensité comme la tête de l'hippocampe sont plus difficiles à localiser que les amers situés à la frontière entre deux régions bien distinctes comme autour du corps calleux.

Tout comme le modèle classique par comparaison d'intensité, DIM repose sur la comparaison des niveaux d'intensité entre le prototype et l'image. Si les distributions des niveaux de gris diffèrent, les performances de ces algorithmes sont nettement diminuées. Nous nous intéresserons donc dans ce qui suit à des modèles qui ne sont pas affectés par ces changements d'intensité.

Chapitre 4 : Modèle de Contours Déformables

Une des solutions envisageables pour obtenir des algorithmes dont les performances ne sont pas dégradées par des changements d'intensité est de travailler sur les contours de l'image. En effet, il existe des détecteurs de contours qui s'adaptent aux variations d'intensité. Dans le Chapitre 4, nous présentons un modèle déformable des contours de l'image. Dans ce modèle, une image de contour provient de la déformation d'une image binaire aléatoire obtenue par tirage à partir d'un prototype qui contient à chaque pixel la probabilité d'observer un contour. Le bruit est modélisé par un canal binaire, qui ajoute ou supprime des contours dans l'image finale. Comme toutes les images d'entraînement sont bruitées, la présence ou non d'un contour à un pixel est en fait une variable cachée. Cela signifie qu'en termes d'estimation il faut à la fois estimer la loi des variables cachées et les paramètres du modèle. On utilise donc un algorithme Expectation-Maximization (EM) pour résoudre ce problème d'estimation. L'algorithme s'écrit simplement et il existe une solution analytique pour l'étape de maximisation. En ce qui concerne l'algorithme de test, la vraisemblance d'une nouvelle image dépend uniquement de la position des amers. Le gradient peut s'écrire analytiquement. Il est donc possible de maximiser la vraisemblance par montée de gradient. Toutefois, il arrive que le niveau de bruit varie en fonction de l'image. Dans ce cas-là, les paramètres de bruit deviennent non pas des paramètres du modèle mais des paramètres de nuisance, à estimer pour chaque image. L'algorithme d'estimation est similaire. Par contre pour le test, il faut optimiser la vraisemblance par rapport à la position des amers et aux paramètres de bruit simultanément. Il faut de nouveau utiliser l'algorithme EM pour estimer la position des amers. En pratique, on introduit une petite modification de l'EM qui permet de simplifier l'étape de maximisation. Au lieu de maximiser l'espérance de la log-vraisemblance comme prévu dans l'EM classique, on maximise la vraisemblance directement, puisque sa dérivée est plus facile à calculer.

Le modèle à contours est testé sur des images synthétiques pour commencer. Lorsque le niveau de bruit n'est pas très élevé, l'erreur de prédiction est faible. Mais, lorsque le niveau de bruit augmente, la maximisation par montée de gradient converge vers des minima locaux et réduit les performances globales de l'algorithme. On observe aussi très nettement que les amers sont plus facilement détectés - c'est-à-dire que l'erreur de prédiction est nettement réduite - s'il y a des contours informatifs à proximité. Enfin, l'algorithme est testé sur les IRM de cerveau pour la détection du corps calleux. Bien que les résultats améliorent la localisation des amers, les performances du modèle à intensité ne sont pas égalées.

Chapitre 5 : Modèle de Segmentation Déformable

Dans le Chapitre 5, nous proposons un autre modèle génératif des images en niveau de gris, capable de s'adapter aux changements d'intensité. Le cerveau est composé de trois types de tissus, la matière grise, la matière blanche et le fluide cérébrospinal, qui apparaissent à des niveaux de gris distincts dans les IRM. Cependant l'intensité observée pour un tissu dans une image donnée dépend des paramètres d'acquisition. Un même tissu peut donc dans deux images différentes correspondre à des niveaux de gris différents.

Par contre, la distribution des tissus dans le cerveau, bien que légèrement variable entre individus, ne dépend pas des paramètres d'acquisition des images. Au lieu de construire un modèle déformable sur les intensités, le modèle proposé dans ce chapitre s'appuie sur un modèle déformable des tissus de l'image. Le prototype probabiliste contient à chaque pixel la probabilité d'observer chacun des tissus. Une image provient alors de la déformation d'une image segmentée aléatoire, obtenue à partir du prototype probabiliste. Pour passer de l'image segmentée à une image en niveau de gris, on utilise un mélange de distributions Gaussiennes. Chaque tissu est caractérisé par une distribution Gaussienne dont les paramètres dépendent de la méthode d'acquisition de l'image.

Dans le modèle décrit ci-dessus, la segmentation de l'image, ainsi que les paramètres d'acquisition sont des variables cachées. L'apprentissage se base donc sur un algorithme EM. Il est possible d'écrire analytiquement le gradient de la vraisemblance d'une nouvelle image par rapport à la position des amers. Il suffit donc d'optimiser cette fonction par montée de gradient.

Dans de nombreux cas, il est plus naturel de modéliser les paramètres d'acquisition comme des paramètres de nuisance. Pour notre modèle, cela signifie que les paramètres du mélange de Gaussiennes dépendent de l'image. Bien que l'algorithme d'apprentissage ne soit pas grandement modifié par rapport au cas précédent, la prédiction de la position des amers ne peut plus être obtenue par une simple méthode de gradient puisque la vraisemblance dépend à la fois de la position des amers et des paramètres photométriques. Nous proposons d'optimiser la vraisemblance de nouveau par un EM qui alterne entre l'estimation des paramètres et l'estimation de la position des amers.

Les algorithmes qui résultent de ce modèle permettent non seulement de localiser les amers, mais aussi d'obtenir simultanément la segmentation de l'image, c'est-à-dire d'assigner à chaque pixel le tissu le plus probable. Les algorithmes proposés sont mis en œuvre pour la détection des amers du corps calleux dans des IRM de cerveau en 2D et en 3D. Les performances sont comparables à celles obtenues avec le modèle à intensité proposé dans le Chapitre 3, mais dans le cas du modèle à tissus, les intensités des images n'ont plus besoin de correspondre. La méthode s'applique donc directement à des images provenant de différentes modalités.

Chapitre 6 : Modèle à Objet Déformable

Dans les chapitres précédents, en vue de limiter les temps de calculs pour la prédiction de la position des amers, nous nous sommes restreints à utiliser des modèles de déformation très locale. Ce choix s'est avéré crucial lorsque les images sont de plus grandes dimensions. Cependant il s'agit d'une forte contrainte sur le choix du modèle de déformation qui nous empêche de travailler avec des déformations affines, puisque par définition la composante affine a un support infini. Dans ce dernier chapitre, nous proposons donc un modèle d'image sensiblement différent, qui nous permet d'utiliser tout type de déformations - à support local ou à support infini - tout en limitant l'effort de calcul à un domaine fini de l'image. Nous modélisons une image comme la superposition d'un objet déformable sur une image de fond. Seul l'objet est en pratique soumis à la déformation qui reste paramétrée par le déplacement des amers. Ce modèle d'image est couramment utilisé

en vision par ordinateur mais rarement en imagerie médicale. Le plus souvent le domaine entier de l'image est soumis à la déformation. En pratique, en limitant l'action de la déformation au domaine de l'objet, on limite le calcul de vraisemblance à ce même domaine fini. La vraisemblance de l'image peut en effet être réécrite sous la forme d'un rapport de vraisemblances comparant la probabilité qu'une sous partie de l'image appartienne à l'objet déformable ou à l'image de fond.

Ce modèle de déformation peut être couplé au modèle à intensité et au modèle à tissus. Dans le domaine de l'objet, l'estimation du modèle est inchangée. Intuitivement, l'estimation du modèle de l'objet revient à apprendre une distribution d'intensité ou de tissus à partir des objets recalés. Pour ce qui est de l'image de fond, l'estimation du modèle consiste à "moyenner" les images avant recalage. Cependant, dans chaque image, il convient de ne pas utiliser les intensités des pixels appartenant à l'objet pour l'estimation de l'image de fond.

Quelques expériences ont été menées sur la détection des amers du corps calleux. Les résultats sont comparables aux résultats des modèles précédents. Enfin on présente une variante de ce type de modèles qui s'applique à de nombreux problèmes en imagerie médicale, comme le recalage d'images avec occlusion, la détection de régions anormales.

Conclusion

Les modèles présentés dans cette thèse sont très généraux et peuvent facilement être adaptés à d'autres problèmes en imagerie médicale, mais aussi à d'autres modalités d'images. En augmentant le nombre d'amers, il est possible d'étendre les modèles proposés au recalage d'images et même dans le cas du modèle à tissus, au recalage et à la segmentation simultanés. Enfin, la famille de modèles déformables proposée peut être élargie à d'autres types d'imagerie. S'il ne s'agit plus d'images en niveaux de gris, il faut pouvoir construire une loi de probabilité sur la quantité mesurée à chaque pixel mais aussi comprendre comment la déformation agit sur cette quantité.

CHAPTER 1

INTRODUCTION

In this chapter we will present some of the challenges encountered in medical image analysis. With the progress of the acquisition techniques the number of high resolution images is on the rise. Quantitative analysis is a key element toward a better understanding of the human anatomy and its variations. This is also a promising way to develop new diagnosis tools. Manual measurements which are both time-consuming and error prone limit drastically the range of applications of quantitative analysis.

In practice most of the measurements are based on image and shape comparison. It is therefore crucial to be able to set correspondences between shapes and between images, with minimal user intervention. Anatomical point landmarks are commonly used as control points for image comparison or for shape analysis. Their precise localization in the image is a tedious and time-consuming task that would gain at being automated.

1.1 Current Trends in Medical Imaging

With the advancements in medical image acquisition techniques, it is now possible to acquire high resolution images of the human anatomy. As the resolution and the number of images increase, automated methods are highly needed to analyze and extract quantitative information from the images. Up until now most of the measurements were performed manually, which takes a tremendous amount of time and consequently limits the scope of studies. When working with 3D or 4D images the visualization difficulties add up to the complexity of the anatomy to make manual quantification even more challenging. This is why automated methods are needed to extract large amount of reliable information from the images.

Up until recently medical images were mostly used as a visualization tool. With the development of quantitative analysis methods, they start playing a different role in medical practices and biomedical research. Examples of quantitative analysis are the delineation of a region of interest, the segmentation of a tumor and the analysis of its evolution in relation with the usage of a drug, the detection of lesion for automatic image screening. Machine learning and computer vision techniques are combined to propose new image analysis tools to physicians to perform what is called Computer-Assisted Diagnosis.

However, the purpose of medical imaging goes even beyond building this type of tools, it is also to build models of the human anatomy, towards better understanding of the human body organization and functions. This discipline, called Computational Anatomy [39] focuses on building atlases representing the average anatomy and its different modes of variations. Although the anatomy is globally similar across individuals, at a finer scale there exist many differences in shape, volume or orientation for example. Several sources of variation can be identified such as simple individual variations, aging, effects

of pathologies. It is a great challenge to understand the different sources of variations and quantify them. By building statistical models of the anatomy it is possible to capture the observed variations in the population and try to distinguish normal from abnormal variability.

The fundamental assumption in Computational Anatomy is that each individual anatomical shape or image is equivalent to a prototype anatomy with respect to some deformations. The mathematical formulation follows Grenander's line of work on pattern theory [38], the object variations are captured by a deformation group acting on the prototype. This approach is used to build a statistical model of a set of images or anatomical shapes in which the template represents the average anatomy while the deformation set is used to characterize the modes of variation. Two organs have received a lot of attention so far: the brain and the heart. The study of the brain by Computational Anatomy shows the evolution of the brain structures as age proceeds, but also the effects of neurological diseases such as the Alzheimer's disease, correlated for example to a loss of volume of the hippocampus and the Huntington's disease which is related to the shrinkage of the caudate [56]. As for the heart, studies using Computational Anatomy have shown that there exists a correlation between the thickness of the ventricle wall and the risk of cardiac disease, [7, 56].

Computational Anatomy faces three major challenges which consist in finding the appropriate deformation from the template to each individual image, estimating the template, building a metric on the deformation space. To facilitate the estimation of the appropriate deformation between two images (or between a template and an image), anatomical landmarks are often used as control points. However, most of the time, the landmarks are still manually located which is time-consuming and error prone. Automated landmarking methods are needed in order to apply Computational Anatomy to larger datasets.

We propose a generic automatic method for landmark detection. It not only needs to be accurate but also simple and fast. It should not need to be given more information about the landmarks than a limited training set of images in which the landmarks have been located. Finally the algorithm should generalize to a variable numbers and types of landmarks, and to different images modalities and dimension.

1.2 The Landmark Detection Problem

There exist several definitions of landmarks in the literature. It is important to differ them properly as the detection method for one type of landmarks does not necessarily adapt to other types of landmarks. We classify the landmarks in three categories: the salient points, the geometrical landmarks and the anatomical landmarks.

1.2.1 Landmarks as Salient Points

In computer vision as it is in the common language, a landmark is a noticeable object in the field of view or in a set of elements. Numerous methods for image mosaics, image registration, video tracking are based on the detection of landmarks also called salient points. Most of these methods are composed of three steps: 1) extracting a set of points of interest in each image or frame, and build a descriptor of each extracted location, 2)

finding the correspondences between extracted points by comparing their descriptors, 3) recovering the displacement or the transformation, eliminating incoherent information when needed. Various methods have been proposed in order to extract the points of interest or salient points from each image. The salient points are located at the center of an image patch with characteristic intensity distribution.

Corners have received a great deal of attention as they are numerous in images containing manly made objects and provide precise 2D information, while edges contain only 1D information. Harris [41] proposed a corner detector based on the eigenvalues of the 2D Hessian matrix of the image intensity, computed on a moving window. This efficient corner detector has been extended to be invariant to scale, affine transformation and change of illumination. Filters have been proposed to detect noticeable parts of the image such as edges or simply patches of the image that contains discriminative information. For instance the SIFT descriptor (Scale Invariant Feature Transform) [52] detect interesting points by examining the local intensity gradient distribution. That distribution is also used to describe the points of interest.

These methods have been demonstrated on image registration problems in computer vision, but detecting corners in medical imaging is slightly more tedious, since the structure usually have smooth boundaries. Robustness of the matching algorithms comes from the number of salient points available images used for computer vision. In medical imaging though, the cues are less numerous and less strong, and the resulting measurements are sensitive to the drift of the point correspondences. The position of some landmarks in the image is not as strongly correlated to other points in the image as it is in computer vision where the 3D space can be modeled by projective geometry. The salient point matching techniques can certainly be used for registering globally two images but it seems less reliable to study the subtle changes of the anatomy in a subregion of a complex organ.

1.2.2 Geometrical Landmarks

Geometrical Landmarks are defined as points in the image characterized by some mathematical and/or geometrical properties. Examples are corners, maximum curvature points. Note that this type of landmarks is defined in the image directly and not on the object.

Two common methods are used to detect such landmarks, with quite good performance rate. The first method is based on filters of the image, such as differential filters [73], dedicated to finding a specific geometric pattern in the image. The tentative locations are those with high response to a particular filter. This is a fast and simple method for detection of geometrical landmarks. The main drawbacks comes from the false positive detections which need to be handle by the matching algorithm. It also requires to design a large set of filters to detect all types of landmarks.

The main competing approach for landmark detection in medical images takes advantage from the fact that structures keep the same topology and global geometry even across individuals. The method consists of matching a geometrical shape model to the data [29, 82]. For example, in order to detect the tip of a structure, one can use a semi-ellipsoid or a paraboloid whose parameters are optimized such that the geometrical model matches the local intensity discontinuities. If the ellipsoid models well the structure and is correctly

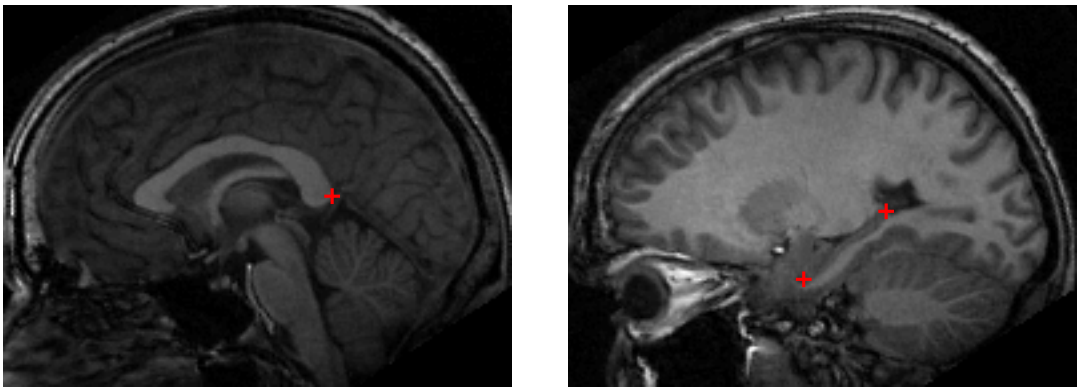


Figure 1.1: **Left**: Sagittal slice of a brain MR image containing the Splenium of the Corpus Callosum (SCC). The red cross represents the location of SCC1, the tip of SCC. **Right**: Sagittal slice of a brain MR image containing the Head and the Tail of the Hippocampus, respectively represented by the leftmost and the rightmost cross.

aligned with its boundaries, the tip of the structure should coincide with the extremity of the ellipsoid [29]. While this method performs well for the detection of landmarks in regions with visible and regular contours, it presents two main drawbacks. First the method requires a geometrical description of the structures surrounding the landmark. It assumes that the local geometry can be described with a simple geometric object. The second drawback comes from the necessity of a visible and reliable contour next to the landmarks to be detected. The amount of a priori knowledge necessary for the choice of the geometrical model of each landmarks is the essential drawback of the method. Providing a mathematical characterization of the geometry around the landmark is often a challenging task. Even though this method has been used with success for the detection of the extremity of the ventricles [29], it would not perform well on the detection of many other landmarks, if the contrast around the landmarks is not as large as it is around the ventricles.

1.2.3 Anatomical Landmarks

Anatomical landmarks are pixels or voxels of an image that correspond to specific locations in the anatomy and therefore are used to set biologically meaningful correspondences between images. Examples are the corner of the eye, the tip of the nose for a face; the head and the tail of the hippocampus in the brain. Contrarily to the salient points and the geometrical landmarks, these points are defined on the object, here the anatomy, and located in the image. Figure 1.1 depicted two sagittal slices of the brain in which few examples of landmarks have been located by an expert.

Although they sometimes correspond to mathematical landmarks or salient point, anatomical landmarks do not need to be located at a position that can be described by a simple geometric model. This is the case of the head of the hippocampus: even though it is located at the tip of an ellipsoidal structure, the methods for mathematical landmarks and salient points do not work. Indeed the absence of clear contour around the landmark makes its detection particularly challenging. The region is in fact rather homogeneous in terms of

intensity and the salient point detectors usually do not detect any interesting points in this region.

We are interested in anatomical landmarks because they carry biologically meaningful information, that other types of landmarks do not necessarily carry. Therefore we know that matching them is meaningful, while matching two maxima of curvature does not necessarily mean that the underlying structure is well aligned. Little has been proposed to detect this type of landmarks except some ad-hoc solutions to detect some specific landmarks such as the Anterior Commissure (AC) and the posterior Commissure (PC) that are commonly used for global registration of brain images. No generic methods have been proposed to learn from examples the appropriate discriminative model that can be used to detect anatomical landmarks.

1.3 Landmark-based Analysis

In many applications it is possible to take advantage of the information carried out by the landmarks. Anatomical landmarks are commonly used to define correspondences between images. Many methods for image registration rely on correspondences between landmarks. Depending on the specific application the landmark correspondences are used to provide an initialization, or as control points. When used as control points, they can either guide the whole registration or be combined to an intensity cost function.

Because they set biologically meaningful correspondences between images and shapes, the configuration of the landmark positions has been used as a representation of shape to perform statistical analysis. The main application is in morphometrics, described in details in [9]. Metrics have been proposed on the space of shapes parametrized by their landmark configurations so that classical statistical method can be applied to perform clustering and more generally shape modeling. [9, 20] present in details several applications to the domain of shape analysis.

If the number of landmark correspondences between images is large enough to encode precisely the deformation that links the two images, Cootes et al.[16] have proposed to learn a statistical model on the deformation encoded by the landmark positions. The resulting model, called Active Shape Model (ASM), has been successfully applied to diverse problem in medical imaging, such as image registration and segmentation.

1.4 Thesis Outline and Contribution

Thesis Outline In this thesis we propose a family of template-based statistical models for medical image analysis. We choose to work with statistical models because they are able to capture the variability of training images in a compact representation. In addition, if the model is generative, it is possible to sample new images from this model and last but not least statistical models can be used to derived optimal algorithms based on maximum likelihood principles. In the statistical framework there exists a clear connection between the modeling assumptions and the optimal cost function. We combine the statistical approach with the methods used in Computational Anatomy, i.e. the images are modeled as the results of the action of random deformation on a deformable template. All the

models presented in this work rely on this modeling principles. Throughout the chapters the statistical model complexity increases to try to obtain more realistic models. We will derive from each of them an automatic landmark detection algorithm to illustrate their properties and specificities. We assess the performance of the resulting algorithms on the detection of landmarks in brain magnetic resonance (MR) imaging.

In the first chapter we present the mathematical framework for template-based image modeling and review some of the statistical models previously proposed to encode a set of images. We present the principles of image registration and show that by choosing an appropriate set of deformations, it is possible to formulate the landmark detection problem as a local registration problem.

In Chapter 3, we present the Deformable Intensity Model (DIM), which models the intensity distribution at each pixel by a Gaussian distribution, whose parameters are given by a deformable template. It is a generic version of the classical method for intensity-based template matching, which computes the square differences between the image and the template. Like every methods relying on the intensity values, the DIM is sensitive to changes of intensity, which can significantly alter the detection performance.

Therefore, in Chapter 4 we propose the Deformable Edge Model (DEM), which models the edge distribution in the image rather than the intensity distribution. In Chapter 5, we present the Tissue-based Deformable Intensity Model (T-DIM). In that model, we assume that even if images have very different intensity distribution, they share the same segmentation template.

Finally, in Chapter 6, we propose to model an image as the superimposition of a still background image and of a deformable object. In this formulation, the likelihood of an image can be rewritten as a likelihood ratio between the object model and the background model. The resulting algorithm has interesting properties, such as reducing the computational load from the complete image support to the object support even if the deformation model has infinite support

Main Contribution The major contribution of this thesis is to propose a generic modeling method for medical imaging. We present a family of generic (or at least explicative) models based on these modeling principles. The resulting models are suitable for a range of applications and image modalities. We present in this thesis the specific case of detecting landmarks in T1 weighted MR images, but this family of models is actually applicable to image segmentation, (multi-modality) registration and object detection.

As for the landmark detection problem, we propose a novel formulation in terms of local registration, which allows us to derive generic algorithms that adapt automatically to any anatomical landmark or set of landmarks. On the contrary of most of the competing approaches for landmark detection, we do not need any prior knowledge about the landmark, since the discriminative pattern is learnt automatically from the training set. The possibility to detect landmarks automatically is a key step towards the generalization of automatic landmark-based registration techniques.

CHAPTER 2

STATISTICAL MODELING FOR IMAGE ANALYSIS

Because images live in a high-dimensional non-Euclidean space, classical statistical methods do not provide a proper approach for image modeling. Deformable models are commonly used to deal with images, modeling an image as the deformation of a template image. Such a representation allows one to compare images by finding the deformation that makes them alike and building metrics on the deformation space. It also allows one to build a generative statistical model of an image set, where a template image represents the main tendency of the population and the deformation distribution encodes the modes of variations. While the generative deformable template is a powerful representation, it requires to deal with the estimation of model parameters: the template and the deformation distribution.

In this chapter we discuss how statistical learning methods can be adapted to work on image and emphasize on the deformable template approach. We then show how these models can be used for landmark detection and discuss the choice of the deformation model for that specific application.

2.1 Statistical Learning for Image Analysis

2.1.1 Probability and Image Space

Definition of an image Let us consider a bounded domain $\Omega \in \mathbb{R}^d$ and a finite regular grid Λ enclosed in Ω. An image is a function which assigns at each location in Ω a weight for each node s of the grid Λ. We denote $x : \Lambda \rightarrow \mathbb{R}$ the intensity function or image. Figure 2.1 represents an image. Assuming the grid contains S nodes, an image is defined has a vector of $\mathbb{R}^S$. However, every vector of $\mathbb{R}^S$ does not represent an image (i.e. a scene or an object) and the set of images, denoted $\mathcal{X}$, is a subspace of $\mathbb{R}^S$. $\mathcal{X}$ is a non-Euclidean space and the addition of two images cannot be defined as the sum of their two characteristic vectors. Figure 2.2 illustrates the problem of defining the addition of two images. The rightmost and the leftmost images represent two sagittal slice of globally registered images. The image at the center is the Euclidean average of those two images. The white matter of the corpus callosum is distorted by this operation and does not present the usual characteristics of that structure.

Definition of Landmarks A landmark is a specific location in the bounded image domain: $\forall k \in \{1, \cdots, K\}$, $y_k \in \Omega$. It is assumed that the landmarks are identifiable in each image and that they set exact correspondence. We denote by $y = (y_1 \cdots y_K)^\top$ the random

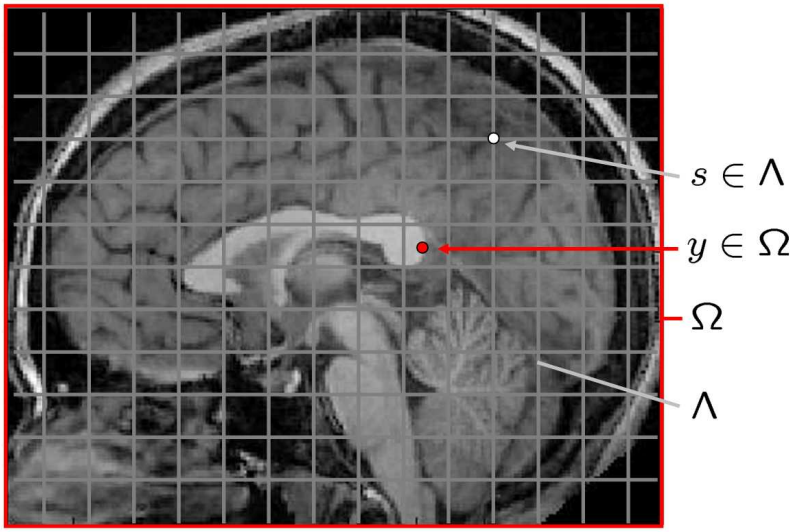


Figure 2.1: Example of a sagittal slice of brain MR image. The domain Ω is the whole image support, while Λ is a finite grid that covers the image. In practice we will work on a finer grid, at the pixel level. Notice that the landmark y, does not need to be on the grid.

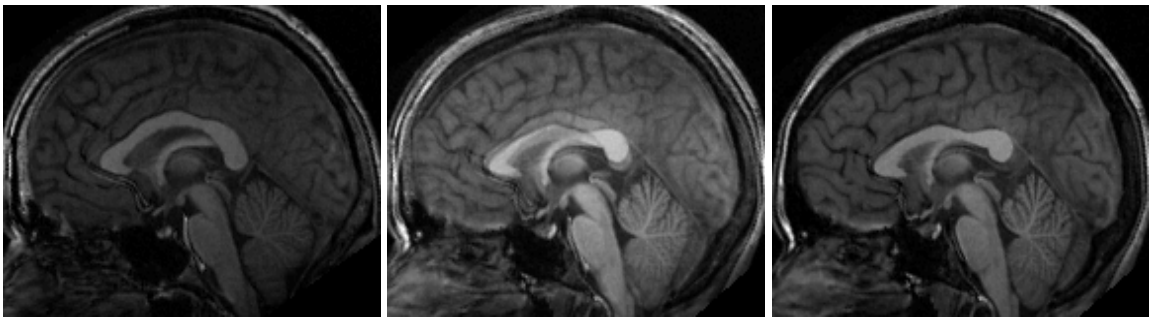


Figure 2.2: The rightmost and leftmost images represent corresponding sagittal slices of brain MR images. The images were manually aligned to the Talairach reference frame. The center image is the Euclidean average of the image intensity vector. Notice how the central white structure, the corpus callosum, is distorted by the averaging in the center image.

vector taking values in $\mathbb{R}^{dK}$ representing the position of a set of landmarks in an image. y is observed in the training set but is unknown in the testing set.

Probability of images as vectors of $\mathbb{R}^S$ Let $x(s)$ be the random variable representing the intensity at pixel s. It takes value in $\mathbb{R}$, therefore the probability space associated to this random variable is $(\mathcal{X}_s, \mathcal{B}, \boldsymbol{\mu})$, where $\mathcal{X}_s$ is the sample space of the image intensity at voxel s, $\mathcal{B}$ the Borel σ-algebra and $\boldsymbol{\mu}$ the associated measure. The real random variable $x(s)$, maps the sample space $\mathcal{X}_s$ to $\mathbb{R}$ and the probability of an event is defined with respect to the measure $\boldsymbol{\mu}$.

Since the image is a finite array of intensity, we model it as a vector of S real random variables and the appropriate probability space is the product probability space $(\mathcal{X}, \mathcal{B}^S, \boldsymbol{\mu}^S)$

with

$$\mathcal{X} = \otimes_{s=1}^{S} \mathcal{X}_s, \quad \mathcal{B}^S = \otimes_{s=1}^{S} \mathcal{B}, \quad \boldsymbol{\mu}^S = \otimes_{s=1}^{S} \boldsymbol{\mu}.$$

$\mathcal{X} \subset \mathbb{R}^S$ is the sample set of images defined on the lattice Λ. We define the real random vector $x = \{x(s), \forall s \in \Lambda\}$, mapping $\mathcal{X}$ onto $\mathbb{R}^S$.

Joint Probability of Images and Landmarks Since the kth landmark is defined as a specific location in the image domain $\Omega \subset \mathbb{R}^d$, the corresponding probability space is simply $(\Omega, \mathcal{B}^d, \boldsymbol{\nu})$, $\mathcal{B}^d$ the d product of the Borel σ-algebra and $\boldsymbol{\nu}$ the appropriate measure. The probability space for the set of K landmarks is the product space $(\Omega^K, \mathcal{B}^{dK}, \boldsymbol{\nu}^K)$ with

$$\Omega^K \subset \mathbb{R}^{dK}, \quad \mathcal{B}^{dK} = \otimes_1^K \mathcal{B}^d, \quad \boldsymbol{\nu}^K = \otimes_1^K \boldsymbol{\nu}.$$

In consequence the probability space corresponding to the joint probability of the image and the landmark is the product space of the image probability space and the landmark probability space. The resulting space is $(\Omega^K \otimes \mathcal{X}, \mathcal{B}^{dK} \otimes \mathcal{B}^S, \boldsymbol{\nu}^K \otimes \boldsymbol{\mu}^S)$.

2.1.2 Statistical Learning and Prediction

We denote by $x^{(i)} \in \mathbb{R}^S$ the vector of intensities of the ith training image and by $y^{(i)} \in \mathbb{R}^{dK}$ the location of K landmarks in that image. Given a set of N gray-scale images on which the landmarks have been located manually $(x^{(1)}, y^{(1)})$, ..., $(x^{(N)}, y^{(N)})$, the problem consists of detecting the location of the landmarks y in a new image x.

Considering the image as a random vector of $\mathbb{R}^S$ and the landmarks as a random vector of $\mathbb{R}^{dK}$, the landmark detection problem can be seen as a classical prediction problem. That is, given a training set of N independent observations, estimate a predictor of the landmark location based on the image intensities: $h : \mathcal{X} \to \mathbb{R}^{dK}$. The predictor is learnt from the training set and used to locate the landmarks in a new image. Each pixel intensity is a random variable, which means that there exists a much larger number of variables than samples, which makes linear model impossible to use directly on the row data. Feature selection methods may be used to reduce the number of variables.A good review of the different proposed methods for feature selection is given in [40]. It is not straightforward though to propose a selection method, adapted to images, which are arrays of highly correlated and redundant variables. A common approach for feature selection consists of ranking the variables, using for example the correlation with the predicted variable y and selecting the top variables to construct a predictor.

We propose to address the landmark detection problem by building a generative model which takes into account the peculiar structure of the images. As before, the model is learnt from the training set of labeled images and used to locate the landmarks in new images. The choice of the model makes the difference from classical statistical learning methods. Statistical models for image analysis have only started to received attention. The slow emergence of statistical methods in image analysis and understanding comes from the peculiarity of the image space, which is a non-Euclidean space.

The statistical learning approach can generally be summarized in a three step procedure. First the relations between the observed variables and the variables of interest

need to be modeled. The modeling assumptions lead to a specific log-likelihood function, denoted $\ell(x, y; \theta)$, with x the data, i.e. the image, y the landmark location, and θ the model parameters. Often the likelihood function is based on a generative (or at least explicative) model of the data, i.e. a joint probability distribution of the observations x and the landmark y (or a conditional distribution: x given y).

Once the modeling choices have been made, the selection of the model, i.e. the estimation of the model parameters θ, is performed by log-likelihood maximization using the training set:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \ell(x^{(i)}, y^{(i)}; \theta).$$

Finally the estimation of the landmark location is given by the Maximum Likelihood Estimator (MLE) using the selected model:

$$\hat{y} = \arg\max_{y} \ell(x, y; \hat{\theta}).$$

Because the model is automatically learnt from the training set, statistical learning allows us to consider different types of landmarks without having to manually tailor a geometric model of the local shape, as it is the case in [66, 29]. Therefore it is essentially possible to work with any landmark that can be labeled by a specialist. Naturally the result of the detection method relies on the availability of a database of training images in which the landmarks have been consistently labeled. The error in the training set will affect the prediction performance of the system.

2.2 Statistical Representation of a Set of Images

2.2.1 Principal Component Analysis and Active Appearance Models

Linear models have been explored to represent a set of images, assuming that the curved image space can be locally modeled as a Euclidean linear space. In [74] such model was used for digitalized human face modeling with same scale and illumination. An image is modeled as a linear combination of some so-called eigenfaces $\boldsymbol{P_p}$:

$$x = \bar{x} + \boldsymbol{P}_p \boldsymbol{b}_p,$$

where $\bar{x}$ is the mean intensity of the training images, $\boldsymbol{P}_p$ a set of orthogonal modes of intensity variation and $\boldsymbol{b}_p$ the vector of gray-level parameters. The model is learnt by Principal Component Analysis (PCA) on the intensity covariance. The eigenfaces correspond to the eigenvectors exhibited by PCA. An image is therefore encoded by the vector $\boldsymbol{b}_p$. Sampling the image space based on this model consists in producing a set of coefficients vectors, which combined to the eigenfaces is supposed to reproduce a random image similar to the ones of the training set. Such a procedure will in practice not perform well at sampling the image space, specially for larger variations from the average. This comes from the assumption of linearity of the image space while it is in reality a curved space. Locally it performs well but discrepancies appear at further distance from the average.

This representation models badly the scale and pose variations. A brute force solution consists in enlarging the training set by adding some instances of faces with different orientations and scales. The modes of variations will therefore incorporate these transformations. It is more natural though to distinguish the variations due to the pose and the scale from the face variations. This is what was proposed in [16, 14] where the shape variations and intensity variations are modeled separately. It is still based on the fundamental idea of approximating locally the curved image space by a linear space, except that the set of images is manually annotated with points correspondences. Shape variations are modeled using a linear model of the position of the corresponding landmarks, learned by PCA:

$$y = \bar{y} + \boldsymbol{P}_g \boldsymbol{b}_g,$$

where $\bar{y}$ is the mean landmark vector, $\boldsymbol{P}_g$ a set of orthogonal modes of shape variation and $\boldsymbol{b}_g$ a vector of shape parameters. In order to estimate the photometric model, the training images are firstly registered (or warped) onto the mean shape using the points correspondences. Then the photometric model is estimated by PCA on the intensity covariance of the warped images $\tilde{x}$:

$$\tilde{x} = \bar{x} + \boldsymbol{P}_p \boldsymbol{b}_p,$$

where $\bar{x}$ is the mean intensity of the warped images, $\boldsymbol{P}_p$ a set of orthogonal modes of intensity variation and $\boldsymbol{b}_p$ the set of gray-level parameters. In this model each image is encoded by two vectors $\boldsymbol{b}_p$ and $\boldsymbol{b}_g$, representing respectively the intensity and shape variation. Sampling from this model consists in choosing randomly $\boldsymbol{b}_p$ to create a gray level image which is then deformed based on the shape variation resulting from the choice of a random shape vector $\boldsymbol{b}_g$.

Linear models have been broadly used to analyze anatomical structures in medical imaging. They can provide shape constraints for image segmentation, or for setting image correspondences [15]. They are also used to analyze shape variations induced by development processes and diseases [9]. However the shape model estimation requires to provide dense correspondences between images, and therefore requires a training set with many manually set correspondences to capture the shape variation through PCA. In addition, in order to avoid overfitting the number of variation modes needs to be limited.This simple model has been used with success in many practical problems.

More recently the Principal Geodesic Analysis [28] has been proposed, generalizing the PCA to a curved space. Models of complex structure such as images can be built, using this approach.

2.2.2 Groupwise Registration and Deformable Templates

Comparing a Pair of Images

Recall from (2.1.1) that an image is defined as real-valued function x on a finite grid Λ containing S nodes, called pixels in 2D and voxels in 3D. We have already mentioned that the sum of two images as vectors of $\mathbb{R}^S$ is not stable on the set of images. Let us consider two images, one of them being the translation of the other. Intuitively we would like the distance between this two images to be small since they represent the same scene

or object. Computing the Euclidean distance between their vectors in $\mathbb{R}^S$ clearly is not efficient at comparing images as the distance between two translated images would be large, the intuition suggests it should be small.

Images and Deformable Templates

In this section, we consider an image as a function from $\mathbb{R}^d$ to $\mathbb{R}$. The main differences from the definition proposed before is that the image is define on $\mathbb{R}^d$ and not a bounded domain, which avoid boundaries issues. Furthermore the intensity is defined at every location of the image support and not only at the nodes of the finite grid Λ introduced in the preceding definition. This new model for images lets us define first the action of a deformation on an image and then an equivalence relation between images.

In Grenander's work on pattern theory [38], an equivalence relation is defined on the image set, so that the images are compared on the quotient space. Two images are said to be equivalent if there exists a transformation that maps one image onto the other. To provide a rigorous definition of the equivalence relation, we present first to discuss how deformations act on images.

Let $\mathcal{F} = \{f : \mathbb{R}^d \to \mathbb{R}^d \text{ such that } f^{-1} \text{ exists}\}$ be a set of smooth deformations of $\mathbb{R}^d$, and $\circ$ the composition law. $\mathcal{X}$ is the set of images defined on $\mathbb{R}^d$. Deforming an image consists in deforming its support and assigning the intensity value of the image to the corresponding location after deformation. Let $x^{(1)}$ be an image defined on $\mathbb{R}^d$, and $f \in \mathcal{F}$ a smooth transformation from $\mathbb{R}^d$ to $\mathbb{R}^d$. The image $x^{(2)}$ is also defined on $\mathbb{R}^d$ is the result of the deformation of $x^{(1)}$ by f:

$$x^{(2)} = f \cdot x^{(1)} \quad \Leftrightarrow \quad \forall t \in \mathbb{R}^d, \quad x^{(2)}(t) = x^{(1)}(f^{-1}(t)). \tag{2.1}$$

We now refer to the action of the deformation f onto x by $f \cdot x$. The composition law is associative: Given three images $x^{(1)}, x^{(2)}, x^{(3)} \in \mathcal{X}$, and the deformations $f_1, f_2 \in \mathcal{F}$, if $x^{(2)} = f_1 \cdot x^{(1)}$ and $x^{(3)} = f_2 \cdot x^{(2)}$, then

$$\forall s \in \mathbb{R}^d, \quad x^{(3)}(s) = x^{(2)}(f_2^{-1}(s)) = x^{(1)}\left(\left(f_1^{-1} \circ f_2^{-1}\right)(s)\right).$$

We define the following equivalence relation between two images:

Definition 2.1. *Let $\mathcal{X}$ be a set of images and $(\mathcal{F}, \circ)$ a group of deformations of $\mathbb{R}^d$. We define the equivalence relation:*

$$\forall x^{(1)}, x^{(2)} \in \mathcal{X}, \quad x^{(1)} \sim x^{(2)} \Leftrightarrow \exists f \in \mathcal{F} : x^{(2)} = f \cdot x^{(1)}.$$

By definition this relation is reflexive, symmetric and transitive, and generates a partition on the space of images. The set of images equivalent to x_0 given a set of deformations $\mathcal{F}$ is called the orbit of x_0 and is denoted by $\mathcal{O}(x_0)$.

The partition in orbits of the space of images provides a natural way to model images from the same orbit as a deformation of a common template, using the transitivity of the equivalence relation. Given $x_0 \in \mathcal{X}$, and $x^{(1)}, x^{(2)} \in \mathcal{O}(x_0)$, there exist $f_1, f_2 \in \mathcal{F}$ such that $x^{(1)} = f_1 \cdot x_0$, and $x^{(2)} = f_2 \cdot x_0$. Then $x^{(1)} \sim x^{(2)}$ because

$$x^{(1)} = f_1 \cdot (f_2^{-1} \cdot x^{(2)}) = (f_1 \circ f_2^{-1}) \cdot x^{(2)}.$$

Representing the image space as a quotient space $\mathcal{X}/\mathcal{F}$ is the underlying idea of deformable templates for image analysis. The choice of a template x_0 and of a deformation set $\mathcal{F}$ provides a partition of the image space into the images that belong to the orbit of the template and those who do not. The problem of finding the transformation or deformation that warps the template onto an image is called registration or image warping.

The quotient space representation is very practical for image modeling and provides a rigorous way to compare images, by building metrics on the quotient space. The challenge of this representation comes not only from the estimation of the transformations from the template to the image instances but also from the estimation of the template itself, since typically only image instances are observed. We will discuss this issue in the following sections.

The equivalence relation is easy to define for images modeled as functions from $\mathbb{R}^2$ to $\mathbb{R}$. Unfortunately in practice, images are finite arrays of intensity values. Therefore defining the action of a deformation on an image is not easy. We will therefore work with a deformable template rather than deforming the images onto the template support. In order to perform real computation though, we will need some times to use an interpolation function. We introduce a generic interpolation function $h : \mathbb{R}^d \times \Lambda \to \mathbb{R}$, which assigns at each location in $\mathbb{R}^d$ a set of weights corresponding to each node of the image grid. At a fixed location in $\mathbb{R}^d$, the weights of the different nodes sum up to 1: $\sum_{s\in\Lambda} h(t,s) = 1$. Their exist different ways to define the weights, depending on the specific technique used for interpolation.

2.2.3 Registration by Energy Minimization

General Formulation

Most of the time, the registration of two or more images is formulated as an energy minimization problem. Registering two images consists in finding the transformation $f \in \mathcal{F}$ which sets the correspondences between the two images. Since the registration algorithms are often not symmetrical, the image which is deformed is called the template or source image x_0 and the other image is called the target image x. The result depends intrinsically on the choice of the energy and of the deformation model. Usually the energy function $\mathcal{J}$ is composed of two weighted terms: the data term $\mathcal{A}$ which measures the similarity between the deformed template $f \cdot x_0$ and the target image x, and the regularization term $\mathcal{R}$ which ensures some smoothness properties to the deformation. Without the regularization term, the registration problem is ill-posed and would have many (potentially improper) solutions. Denoting by γ a weighting term, the energy function associated to the warping of a source image x_0 onto a target image x is of the form:

$$\mathcal{J}(x, x_0, f, \gamma) = \mathcal{A}(x, x_0, f) + \gamma\mathcal{R}(f). \tag{2.2}$$

In the case of a groupwise registration, i.e. when the problem consists in finding the set of transformations $(f_1, \cdots, f_N)$ which aligns a set of images $(x^{(1)}, \cdots, x^{(N)})$ with a common unobserved template image x_0, the energy function becomes:

$$\mathcal{J}(x_1^N, x_0, f_1^N, \gamma) = \sum_{i=1}^{N} \mathcal{A}(x^{(i)}, x_0, f_i) + \gamma\mathcal{R}(f_i), \tag{2.3}$$

denoting by x_1^N, f_1^N respectively the set of images and the set of transformations.

Therefore in order to design a registration algorithm, one typically chooses a deformation model, a data attachment term and a regularization term, which define an energy function to be minimized. Most of the time the choices are made arbitrarily, therefore numerous possibilities have been explored, surveys on registration techniques for medical images [87, 35] present some of the possible combinations of choices of data and regularization terms.

The Data Attachment Term

Similarity measures are typically classified in two categories. The first group is based on sparse feature correspondences. The problem is composed of three steps: first extracting the features, then establishing their correspondences between images and finally interpolating the deformation to provide dense correspondences between the images. This is the type of registration that one typically uses when landmarks are given. The main advantage of this type of similarity measures is its sparseness which makes it particularly suitable for large images and/or fast computation algorithms. Their main drawback is that since they rely on the correspondences of features only, they tend to be less accurate in regions with few features. Our problem of landmark detection corresponds precisely to the two first steps of the feature-based registration techniques, i.e. detecting interesting features and matching them across images.

The second category of similarity functions, so-called intensity-based measures, compares the intensity of the deformed template $f \cdot x_0$ and of the target image x. As opposed to the feature-based measures, this type of cost function relies on a dense comparison between the deformed template and the image. Classical similarity functions are the absolute intensity difference [6], the sum of squared intensity difference (SSD) (e.g., [77, 31, 3]) or the correlation coefficient [63]. Some other cost functions are based on other functions of the image such as local Fourier coefficients [33], edge repartition [51] to cite few of them. Finally another category of functions is based on information theory criterion, comparing the intensity distribution of the source and the target, using joint entropy [71, 13] or mutual information [13, 77, 80, 54]. .

More recently, efforts have been made to join the two mainstream registration similarity functions using both feature correspondences and dense intensity matching, [43, 25, 42].

The Deformation Model

The choice of the deformation model is often driven by the specific problem and application at hand. If a global alignment is sufficient, rigid or affine transformations can be used to perform registration. One can take advantage of their low dimension to use robust estimation methods. On the other hand though, these transformations are generally not enough to model the deformation of structures in the brain.

Therefore non-rigid deformation models are often preferred to model subtle changes in the anatomy. There exists numerous representations for non-rigid (and non-affine) deformations. Low dimensional representation such as free-form deformations, or more generally spline-based deformations, are parametrized by the displacement of control points.

The deformation is obtained by interpolating the control points displacements to the rest of the image using smooth basis function. The choice of the basis function influences significantly on the properties of the resulting deformation.

Other approaches consists in modeling the image as a physical continuum whose deformation follows some mechanic models such as elastic or fluid deformation. In this case the deformation field (or the velocity field) is the solution of a Partial Differential Equation (PDE). Examples of image registration using these models can be found in e.g.[5, 18, 10, 50].

The weight parameter of (2.3) is most of the time manually tuned, and sometimes varies as the optimization is carried out to favor first rigid deformations and afterwards allow non-rigid deformation to improve the matching result. It is thought that such techniques prevents the optimization algorithm from getting trapped in local minima.

Thesis Contribution to the Image Matching Problem

It is sometimes difficult to understand all the underlying assumptions made by choosing an energy function rather than another. It would be interesting to know for certain types of images what energy function is optimal. In [77, 65, 33] it was shown that if two images differ by a Gaussian noise, the optimal matching function is the sum of squared differences (SSD). Unfortunately it is common that this model is not enough to model image differences and in these cases, it is not known what energy function would be optimal.

In this thesis, we propose to build a family of deformable statistical models for medical images. We present different examples of model for gray-level images, edge images and multi-modal gray-level images (i.e. coming from different acquisition protocols). Using maximum likelihood principles, we derive simple algorithms for image matching based on the modeling assumption and provide the corresponding optimal energy function. It is possible to understand the connection between the assumptions made and the resulting energy function. In all cases the derived cost functions are very intuitive and in simple cases correspond to some of the well-known energy functions such as SSD. The framework we present in this thesis is very generic and can be used for several problems in medical imaging such as image registration, image segmentation (partitioning the image pixels in different groups based on intensity for instance), and image tracking (following an object in a video sequence). It can also be used for joint problems such as simultaneous registration and segmentation of images. In this thesis, we illustrate the different models on the specific problem of landmark detection in brain MRI, defined in detail in Chapter 1. In fact, the proposed approach to landmark detection can be seen as a local registration problem.

2.3 Landmark Detection as a Local Registration Problem

We model a landmarked image as the result of a bijective deformation acting on a template x_0, such that the landmarks location in the template $\bar{y}$ are mapped to their location in the image y. The training set of N landmarked images $\{(x^{(1)}, y^{(1)}), \cdots, (x^{(N)}, y^{(N)})\}$ is modeled as the result of a set of random deformations acting on a common template x_0. For each image i, the landmark matching constraint is fulfilled: $f_i(\bar{y}) = y^{(i)}$. When $\bar{y}$ is fixed, it is equivalent to estimate the location $y^{(i)}$ or to estimate the deformation

f_i, which is the deformation that sets correspondences between the landmarks. Since landmarks characterize the local geometry, it makes sense to look for a similar pattern in the target image by image matching. Therefore the landmark detection can be seen as a local registration since it consists in finding the deformation that matches the best the template and the image at or around the landmarks. In terms of energy function, the landmark detection problem is similar to a registration problem:

$$\hat{f} = \arg\max_{f \in \mathcal{F}} \mathcal{A}(x, x_0, f) + \gamma \mathcal{R}(f) \quad \text{and} \quad \hat{y} = \hat{f}(\bar{y}). \tag{2.4}$$

As we argued in the preceding section, there exist numerous ways to build a similarity function. We choose to derive it from a statistical model of the image. Similarly, many choices of regularization term have been proposed, generally designed to minimize the amount of deformation. In our case the regularization term is part of the choice of the deformation model.

2.3.1 Choice of the Deformation Model

In order to formulate the landmark detection as a registration problem, i.e. as the estimation of a deformation, fixing the reference location $\bar{y}$ is not enough, as there exists an infinity of deformations that map the reference location to the appropriate landmark location in the image. Therefore to build a bijective map from the set of landmark configurations to the set of deformations, it is necessary to reduce the set of deformations such that there exists a unique deformation that corresponds to a configuration y.

We will first discuss how to build the appropriate deformation set using rigid or affine transformations. Two issues arise from the use of this type of deformation. First, they are restricted to a limited amount of landmarks. Second they may require intensive computation due to their infinite support. Therefore we present in greater details the spline-based deformation models. Indeed this model has several important advantages over rigid and affine transformations. First they are directly parameterized by the displacement of the landmarks and second the model adapts esily to a variable number of landmarks. We will also discuss the problem of the choice of the kernel and the consequence in terms of deformation support and computation.

Rigid Transformation

We denote $\boldsymbol{GL}(d)$ the linear group of all real valued $d \times d$ matrices with non-zero determinant (invertible) and with matrix multiplication as composition law. The affine group is the semi-direct product of groups $\boldsymbol{GL}(d) \otimes \mathbb{R}^d$ with elements $\{(A, a) : A \in \boldsymbol{GL}(d), a \in \mathbb{R}^d\}$, such that $(A, a) \circ (B, b) = (AB, Ab + a)$. In homogeneous coordinates the affine group is a matrix group with elements:

$$\begin{pmatrix} A & a \\ 0 & 1 \end{pmatrix}, \quad \begin{matrix} A \in \boldsymbol{GL}(d) \\ a \in \mathbb{R}^d \end{matrix}, \quad \text{such that } (A, a) \cdot t = \begin{pmatrix} A & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}.$$

Estimating the transformation from the landmark configuration requires to solve in (A, a) the linear system of dK equations such that the landmark matching constraint is satisfied.

We denote by $\bar{y} \in \mathcal{M}(d,K)$ the reference location in the template and $y \in \mathcal{M}(d,K)$ the landmark location in an image, the linear system to be solved is:

$$\begin{pmatrix} A & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \bar{y} \\ \mathbb{1}_K^\top \end{pmatrix} = \begin{pmatrix} y \\ \mathbb{1}_K^\top \end{pmatrix}.$$

Three situations may occur:

- there exists an *infinity of solutions* if the linear system is under-constrained, i.e. if the number of displacement constraints is inferior to the number of parameters.
- there are *no solutions* to the linear system, if the number of constraints is larger than the number of parameters. A least square based estimation can be used but the bijection from the space of landmark configurations to the space of deformations is not guaranteed anymore.
- the linear system has a *unique solution* which occurs when the number of equations and the number of constraints matches.

Therefore in order to build a bijection from the configuration set $\mathcal{M}(d,K)$ to $\mathcal{F}$ the set of deformations, it is needed that the number of landmarks coordinates dK and the number of free parameters be equal. Since an affine transformation of $\mathbb{R}^d$ contains at most 6 independent parameters if $d = 2$ and 12 if $d = 3$, it can be used to represent the displacement of at most 3 landmarks in 2D or 4 landmarks in 3D. If the number of landmarks K is inferior to 3 (if $d = 2$) or 4 (if $d = 3$), the set of transformations $\mathcal{F}$ needs to be a subset of the affine transformation: $\mathcal{F} \subset \boldsymbol{GL}(d) \otimes \mathbb{R}^d$ such that $\dim(\mathcal{F}) = dK$ and such that there exists a unique f that maps $\bar{y}$ onto y. There exist several subsets of $\boldsymbol{GL}(d) \otimes \mathbb{R}^d$ that fulfill the landmark matching condition, the choice of one subset rather than another is arbitrary. For instance, if one needs to model the deformation of 2 landmarks in 2D, it is possible to use a composition of a translation (2 parameters), a rotation in the plane (1 parameter) and a scaling (1 parameter) or to shuffle the order of these transformations, leading to a different set of deformation. Not only that, the set is generally not a group.

Three other difficulties arise from the usage of affine transformations: one is limited to 3 or 4 landmarks, the deformation set needs to be redefined from scratch each time a landmark is added and finally the rigid and affine transformations have by definition an infinite support which can lead to involved computations when comparing large images.

Non-rigid Deformation

The main advantage of non-rigid deformations over affine transformations comes from the possibility to deal automatically with as many landmarks as desired without changing the deformation model.

In the landmark detection problem, the main goal is to locate K landmarks. We naturally parameterize the deformation $f : \mathbb{R}^d \to \mathbb{R}^d$ with the landmarks displacements from the reference location $\bar{y}$ to the image location y, using spline interpolation to define the deformation on the rest of the domain. Again,there exists an infinity of deformations fulfilling the landmark matching constraint: $f(\bar{y}) = y$, therefore we need to reduce the set

of possible deformations. The spline interpolation setting is particularly suitable to build such a deformation set. It can indeed be proven that it is enough to choose the interpolating basis function to ensure that given the location of the landmarks in the template there exists a unique deformation that maps the landmarks to their location in the image.

To be more precise, we introduce the spline interpolation framework following the formulation in [21, 78, 83]. Spline interpolation is used to interpolate the landmark displacements to the rest of the domain, defining a dense vector field. The interpolation problem can be seen as finding the deformation of minimal norm that fulfills the landmark matching constraints:

$$\text{Find } f \in \mathcal{F}, \text{ that minimizes } \|f - Id\|^2 \text{ under the constraint that } f(\bar{y}) = y. \tag{2.5}$$

The amount of deformation is measured by the norm of the deformation $\|f - Id\|_{\mathcal{F}}$ assuming that $\mathcal{F}$ is a Hilbert space of continuous real-valued functions on Ω_T and that $\kappa : \Omega_T \times \Omega_T \to \mathbb{R}^d$ is its associated self-reproducing kernel. It can be proven that if $\mathcal{F}$ is a Hilbert space with self-reproducing kernel κ, and $(\bar{y}_1, \cdots, \bar{y}_K)$ is a set of distinct points of Ω_T, all the functions $f \in \mathcal{F}$ that minimize the norm $\|f - Id\|_{\mathcal{F}}$ can be written as a linear combination of the kernel function:

$$\forall f \in \mathcal{F}, \forall t \in \Lambda_T, \quad f(t) = t + \sum_{k=1}^{K} \kappa(t, \bar{y}_k)\beta_k, \quad \text{with } \beta_k \in \mathbb{R}^d. \tag{2.6}$$

Therefore if the solution of the interpolation problem (2.5) is to be found in $\mathcal{F}$, it is of the form of (2.6). We denote S_{dK} the $dK \times dK$ matrix containing the kernel value for every pair of landmarks $\forall k, k'$, $S_{k,k'} = \kappa(\bar{y}_k, \bar{y}_{k'})$. $\beta \in \mathbb{R}^{dK}$ is the stacked vector of coefficients β_k. It follows that:

$$\|f - Id\|^2_{\mathcal{F}} = \beta^\top S_{dK}\beta \quad \text{and} \quad f(\bar{y}) = y \Leftrightarrow S_{dK}\beta = y \tag{2.7}$$

There exists a unique solution to the interpolation problem determined by the linear coefficients β_k, obtained by solving the linear system arising from the landmark matching constraints: $\beta = S_{dK}^{-1} y$.

In practice, the kernel associated to a specific Hilbert space is not always trivial to identify, but thanks to the Mercer's theorem it is possible to work backwards. Instead of looking for the reproducing kernel of a given Hilbert space, it is possible to directly choose the kernel function. Indeed the theorem states that for all symmetric definite positive kernel there exists a Hilbert space in which κ is the reproducing kernel. Therefore, it is enough to choose such a kernel to ensure that there exists a Hilbert space in which the interpolation method holds, without specifying its norm. Radial basis functions (RBF) are often used in imaging to interpolate the motion of control points to the rest of the image domain. Classical examples of interpolating functions are the B-splines, Gaussian splines and the Thin-Plate Splines (TPS). Because the TPS incorporates an affine transformation in the deformation model, it has infinite support, while B-splines and Gaussian splines have local supports or can be easily approximated on a local support. The deformable models we propose in the following chapters hold whichever the chosen kernel.

In what follows we will present some kernels in relation with our problem. Ideally we would like to use a kernel which has local support but also model the effect of affine

transformation. This is why we present in what follows the Gaussian kernel, which can be considered as a local kernel, the Thin Plate Spline which incorporates affine transformation in the deformation model but also the Clamped Plate Spline model that restrict the domain to a subregion of the image.

2.3.2 Different Spline Models

Gaussian Spline A simple example is the Gaussian kernel:

$$\kappa(t, \bar{y}_k) = \exp\left(-\frac{\|t - \bar{y}_k\|^2}{2\sigma^2}\right). \tag{2.8}$$

The spread of the displacement is controlled by the variance of the kernel. When the variance increases the displacement is interpolated on a wider window around the landmarks, resulting in a smoother deformation. If one tries to model small translation of a region of the image via the displacement of sparse landmarks, the kernel variance σ^2 needs to be large to avoid local distortions of the image region. Therefore the Gaussian spline model is good at localizing the computational effort but not good at capturing global transformations unless the number of landmarks is large enough which is not the case generally in the landmark detection problem.

Affine Spline at Infinity In order to overcome the distortion effect created by the Gaussian spline, one can choose to modify the kernel and incorporate affine transformations. The affine kernel as defined in [85] is:

$$\kappa_{aff}(t, \bar{y}_k) = \frac{t^\top \bar{y}_k}{\lambda} + \frac{1}{\omega}, \tag{2.9}$$

with λ and ω parameters of the kernel. The nature of the resulting deformation depends on the choice of the parameters. Λ controls the weight of the linear transformation while ω controls the translation. If added to a Gaussian kernel, the affine kernel allows one to add an affine component to the generated deformation. The amount of affine deformation and local non-rigid deformation is controls by the kernel parameters. The support of the affine part is infinite, therefore the resulting deformation acts on the whole domain of the image.

Thin-Plate Spline The usual model for non-rigid deformation with an affine component is the Thin-Plate Spline model (see e.g.[67]). This interpolation method corresponds to the physical problem of fitting an infinite surface to the data while minimizing the bending of the surface [21, 78]. While used for decades in mechanics and physics, it was first introduced for the registration of images [8]. This approach has been then broadly used for the estimation of deformation from control point correspondences. The bending energy of a surface is defined as the norm of the deformation:

$$\forall f \in \mathcal{F}, \quad \|f\| = \iint_{\mathbb{R}^2} (\Delta f)^2 \, dt, \tag{2.10}$$

with Δ the Laplacian operator. In this definition, affine transformations have zero norms, therefore the unique solution of the interpolation problem is the sum of an affine term and a non-rigid term. Given a set of landmarks $(\bar{y}_1, \cdots, \bar{y}_K)$, the interpolation problem admits a unique solution of the form:

$$f(t) = At + b + \sum_{k=1}^{K} \beta_k \kappa(t, \bar{y}_k), \text{ with } \beta_k, b \in \mathbb{R}^d \text{ and } A \in \boldsymbol{GL}(d).$$

It can be shown that the optimal kernel function is, depending upon the dimension of the ambient space $\mathbb{R}^d$:

$$\text{if } d = 2, \quad \kappa(t, \bar{y}_k) = \frac{\|t - \bar{y}_k\|^2}{16\pi} \log(\|t - \bar{y}_k\|^2),$$
$$\text{if } d = 3, \quad \kappa(t, \bar{y}_k) - \frac{1}{8\pi}\|t - \bar{y}_k\|.$$

In [36, 69], a generalization of the bending energy is proposed to constraint the interpolated deformation to be smoother or to deal with data of higher dimension.

Since affine transformations have zero norm, the optimization of the bending energy arises to an optimal combination of affine transformation and non-rigid deformation. By choosing the affine transformation that explains as much as possible the data, the bending energy is reduced and corresponds only to the residual non-rigid deformation. This deformation model is particularly appropriate to take into account affine transformation and non-rigid deformation simultaneously. The interpolated deformation is therefore acting on the whole domain of the image.

Because they add an affine component, both the deformations generated by interpolation using the TPS of the affine kernel have infinite support. It means for image comparison that unless one truncates arbitrarily the cost function to a bounded region of the image, it is not possible to reduce the computational cost to a specific region of the image.

Clamped Plate Spline In order to limit the computational cost, it is useful to build deformation models in which the displacement of the landmarks have a local effect only or at least do not affect the border of the image. However we would also like the deformation model to deal with affine transformations. As we have seen above, it is not straightforward to build such a deformation model. Some attempts have been made to use TPS on a bounded support by defining the bending energy on the bounded domain [36, 70], and thus approximating the TPS solution. Far from the boundaries of the domain, the resulting deformation is unchanged compared to the TPS, but at the boundaries it differs significantly. Another approach proposed in [75] consists in looking for a function which on one hand minimizes the same bending energy as the TPS (2.10), satisfying the landmark matching constraints and vanishing smoothly at the limit of a specified domain (a unit disk) around the object of interest in the image. The kernel is the Green function of the biharmonic equation with the appropriate limit conditions. In the plane, the kernel function is:

$$\forall t \in \Lambda_T \setminus \{\bar{y}_k\}, \quad \kappa(t, \bar{y}_k) = \|t - \bar{y}_k\|^2 (A^2(t, \bar{y}_k) - 1) - \log A(t, \bar{y}_k)), \tag{2.11}$$
$$\text{with} \quad A(t, \bar{y}_k) = \frac{\sqrt{\|t\|^2\|\bar{y}_k\|^2 - 2\|t\|\|\bar{y}_k\| + 1}}{\|t - \bar{y}_k\|}.$$

The value of the kernel at each landmarks is defined by continuity.

To illustrate the different kernels, we present in Figure 2.3 the result of the interpolation of the displacement of 5 landmarks for different choice of kernel. Subfigure 2.3(a) illustrates the displacement of the 5 landmarks in the plane. Subfigure 2.3(b) presents the resulting deformation when the kernel is a Gaussian with $\sigma = 0.25$. The deformation acts on a limited area around the landmarks. In Subfigure 2.3(c) an affine kernel was added to the Gaussian kernel, notice how the grid is globally displaced but locally less bended. Subfigure 2.3(d) corresponds to the same kernel but in which only the linear component is penalized. Notice how the grid is translated and locally deformed to fulfill the landmark matching constraints. Subfigure 2.3(e) represents the Thin Plate Spline results. Notice it differs from Subfigure 2.3(c), the affine transformation is larger but the local deformation of the grid is less important. Finally Subfigure 2.3(f) represents the interpolated deformation when using the Clamped Plate Spline model. Inside of the deformation domain (the unit disk) the deformed grid is similar to the one of TPS. The deformation vanishes on the unit circle, therefore next to the boundary the deformation is very different.

The possibility to work with a local support function is a great advantage for landmark detection because it reduces the computational load to small regions around the landmarks, where most of the information is contained. In the following chapters, to avoid boundary issues, we will assume that the deformation does not affect the boundary of the image. In the experiments we present we use the Gaussian kernel. In Chapter 6 though we propose a model in which it is possible to deal with infinite support deformation while keeping the matching to a finite domain of the image, which allows us to reduce the computational cost.

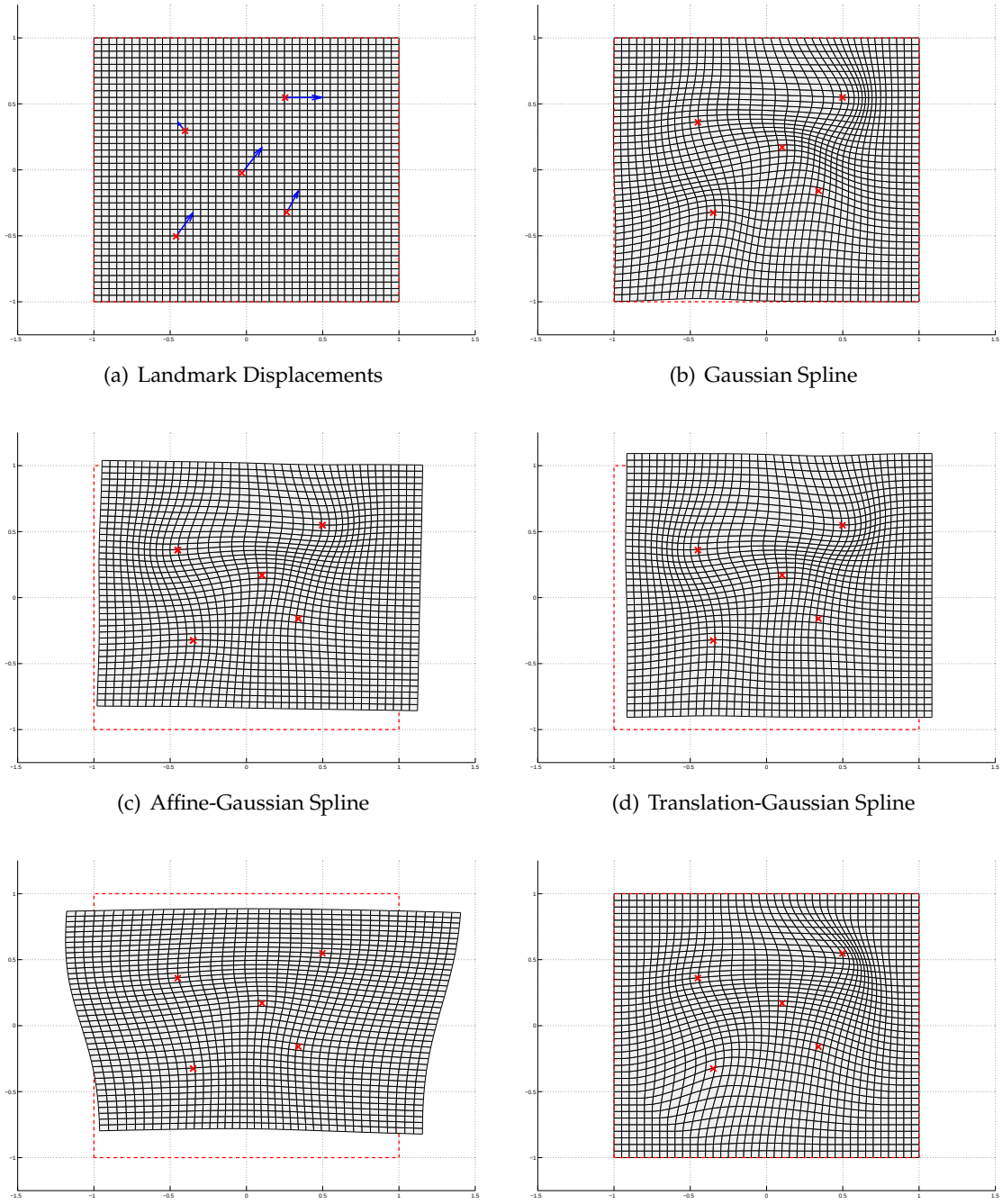
To summarize, choosing the appropriate deformation set $\mathcal{F}$ is equivalent to choosing the norm of the deformation set. This norm can be used as a regularization term in the energy function (2.4), in which case it is enough to look for the best deformation in $\mathcal{F}$, using its generic form (2.6). Therefore the landmark detection problem becomes:

$$\hat{f} = \arg\max_{f \in \mathcal{F}} \mathcal{A}(x, x_0, f), \text{ with } f(t) = t + \sum_{k=1}^{K} \kappa(t, \bar{y}_k)\beta_k. \tag{2.12}$$

Thus it is equivalent to optimize the energy function with respect to the landmark locations y or with respect to the deformation parameters β. In the following chapter we write the optimization problem as a function of the landmark position but in practice for implementation, it is sometimes easier to work on the deformation parameters directly.

2.4 Model Estimation

Using the quotient space representation, it is possible to build a deformable model based on a template x_0 and a probability distribution on the set of chosen deformations $\mathcal{F}$. Both template and probability distribution are the model parameters. In this section we will discuss some of the learning strategies encountered in the literature. Indeed the choice of the template and the estimation of the transformation distribution is a classical



(a) Landmark Displacements

(b) Gaussian Spline

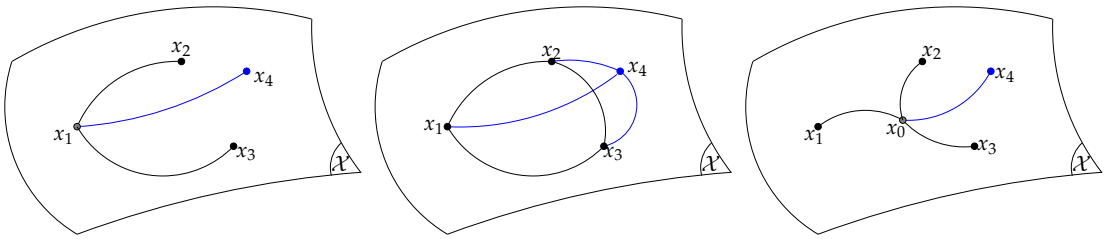
(c) Affine-Gaussian Spline

(d) Translation-Gaussian Spline

(e) Thin Plate Spline

(f) Clamped Plate Spline

Figure 2.3: Different spline models. Subfigure (a) represents a finite grid with 5 landmarks (red crosses) displaced along the blue arrows. Each subfigure from (b) to (f) represents the interpolated deformation when using the corresponding spline kernel. The dashed square represents the image domain boundaries before deformation. (b) Gaussian spline with $\sigma = 0.25$. (c) Affine kernel added to the Gaussian kernel ($\lambda = 1$,$\omega = 1$,$\sigma = 0.25$), (d) Translation kernel added to the Gaussian kernel ($\lambda = 10^3$,$\omega = 1$,$\sigma = 0.25$), (e) Thin-Plate Spline, (f) Clamped Plate Spline vanishing outside of the disc of radius 1.



(a) Arbitrary Choice (b) Discriminative Model (c) Generative Model

Figure 2.4: Different strategies to represent a set of images. Given a set of three images x_1, x_2, x_3, each subfigure represents a different strategy of representation. In (a) one of the images of the training set is chosen as reference, and therefore a new image x_4 is compared to this reference image. In (b) no template images have been chosen, and a new image is compared to each of the training samples. Finally in (c), a template is found and therefore the new image is compared to that template only.

problem. The particularity in the problem we are interested in is that we have a training set of labeled images, i.e. that local correspondences between images are given. The main strategies for dealing with the template issue are illustrated in Figure 2.4.

Choosing the Template

A simple and broadly used way to overcome the template estimation problem, consists in choosing one of the training images as a reference and register all the other images onto that template, as illustrated in Figure 2.4(a). By choosing one of the training images, the estimated template is a noisy and deformed estimate of the true unobserved template x_0. Hence some images are badly represented by this estimate. Ad hoc solutions have been proposed to reduce the noise in the template such as filtering the chosen image in order to remove the noise, but the resulting template remains a deformed image of the true template x_0.

Some template-free approaches have been proposed (see e.g.[57]) to overcome the issue of estimating a template. By performing pairwise comparisons between images, a distance matrix is built. Figure 2.4(b) illustrates this approach. While it is a useful representation of the set of images to perform classification, it is not possible to generate new images from this model. In addition, given a new image, one needs to register it to each of the training images independently. It is possible to use this approach to locate landmarks in a new image: find the set of deformation that register each of the training images with the new image. Each time the image is registered to one of the training image, it produces a tentative location for the landmarks. Therefore one can use the average location as a predictor of the landmark position in the new image. The average predicted location would probably be better than the result of the registration of a single image, but the lower generalization error comes with a very high computational cost. Furthermore it is neither possible to sample new images nor to compress the training data.

The best way to represent a set of images is to model each of the training images as

a random element of the orbit of a common template x_0, under the action of a group of deformations $\mathcal{F}$. This model is efficient, because each image is defined by a single deformation, as illustrated in 2.4(c). It is also possible to build a statistical model, for which the template represents the main tendency while the deformation distribution represents the variations of the data set. Images can be sampled from this model. The downside though is that the template needs to be estimated.

Online Template Estimation

Because the set of images is not a vector space it is not possible to simply average the images pixel by pixel to produce a template image. The result would not belong to the image space. Since an image is modeled as the result of a random deformation of the template, it has been proposed in [34] to average the deformations rather than the images to estimate a template. The estimation algorithm consists in starting from an initial guess (one of the images for example), estimating the set of deformations $(f_1, \cdots, f_N)$ that register the current template estimate onto the training images and applying the average deformation to the current template estimate. The procedure is repeated until convergence. There is no guarantee though that it will converge. If the correspondences between images are known, it is equivalent to finding the barycenter of the deformation set that maps the training images to each other and applying it to some initial guess.

Consider the problem of learning a template from a set of grayscale images. Such an algorithm depends on the initial guess and would not necessarily represent the image set in terms of grayscale images. Because there exists some intensity variation in addition of the geometric variation, one really seeks the image

$$x_0 = \arg\min_x \sum_{i=1}^{N} \|x - x^{(i)} \circ f_i\|^2. \tag{2.13}$$

When the image correspondences are given, the common method for template estimation (see e.g.[16]), consists first in registering the images onto the template support and then averaging at each pixel the resulting vector of intensities. In general interpolation is needed since the images are defined on a discrete grid only. If the correspondence are not given, an iterative algorithm is used to alternate the estimation of the template and the registration of the training images to the template [33].

In all what precedes we have assumed that an image is the result of the action of a deformation on a template. Therefore we define the template as the image:

$$x_0 = \arg\min_x \sum_{i=1}^{N} \|x \circ f_i^{-1} - x^{(i)}\|^2. \tag{2.14}$$

This is the definition that was also proposed in [2, 53]. In our case the deformations are given by the landmark correspondences in the training set. However in general in template estimation problems they are unknown. The deformations can either be modeled as nuisance parameters [33] or as hidden variables as argued in [2]. In both cases the optimization is carried out by the Expectation-Maximization (EM) algorithm (or its mode approximation). Modeling the deformations as nuisance parameters makes the

computation easier but prevents from estimating the distribution of the deformations and thus to estimate a complete generative model. If the deformations are modeled as hidden variables, the E-step of the EM algorithm consists in computing the expectation of the posterior distribution, i.e. to integrate over all possible deformations. In [1] a solution using a Markov Chain Monte Carlo approximation coupled to the Stochastic Approximation of the EM algorithm is used to overcome the computation difficulties. The method is computationally challenging but has proven to deal particularly well with noisy data.

We will face in the following chapters the same dilemma: should we model the unobserved quantities as hidden variables or nuisance parameters?

2.4.1 Estimation of a Model with Missing Variables

Let us review briefly the EM algorithm. The Expectation-Maximization algorithm is an iterative scheme, used to maximize a likelihood function with respect to the model parameters when some variables are unobserved or missing. We will use some generic notations for this subsection. Let x and y be continuous random vectors. x is observed but y is unobserved or hidden. Let θ be a vector of parameters to be estimated. Given θ, the estimation of the distribution of the missing variable vector would be straightforward and similarly, given the hidden variables, the model parameters could be estimated using a least-square method for example. Since θ is unknown and y is unobserved, it is necessary to perform a joint estimation. The EM algorithm is one of the methods that can be used for this joint estimation. Introduced in [19], it is now well understood and largely used to solve estimation problems with missing data. The objective consists of finding the vector of parameters θ^* which maximizes the distribution of the data given the parameters.

$$\theta^* = \arg\max_{\theta} \ln p_\theta(x) = \arg\max_{\theta} \ln \int_y p_\theta(x,y)dy. \tag{2.15}$$

We assume that y has a density denoted $p_{\theta'}(y)$. Using the Bayes' formula:

$$\ln p_\theta(x) = \int_y p_{\theta'}(y|x) \ln p_\theta(x) dy \tag{2.16}$$

$$= \underbrace{\int_y p_{\theta'}(y|x) \ln p_\theta(x,y) dy}_{Q(\theta,\theta')} - \int_y p_{\theta'}(y|x) \ln p_\theta(y|x) dy, \tag{2.17}$$

such that:

$$\ln p_\theta(x) - \ln p_{\theta'}(x) = Q(\theta,\theta') - Q(\theta',\theta') + DL(p_{\theta'}(y|x), p_\theta(y|x)). \tag{2.18}$$

$DL(\cdot,\cdot)$ is the Kullback-Leibler distance. Since it is always non-negative, $Q(\theta,\theta') - Q(\theta',\theta')$ is a lower bound of the density $p_\theta(x)$. In practice it is enough to find θ such that $Q(\theta,\theta') - Q(\theta',\theta') \geq 0$ in order to get an increase of the likelihood function. Intuitively, the EM algorithm iterates between finding an easy-to-compute lower bound of the likelihood function and maximizing it with respect to the model parameters. The algorithm can be seen as a double maximization: in the E-step the lower bound is maximized with respect to the distribution of the hidden variable $p_{\theta'}(y)$ using the model parameters learnt at the

Algorithm 2.1 The EM algorithm

Initialize with θ' the model parameters,

Iterate until convergence:

- **E-step**: $$p_{\theta'}(y) \leftarrow \arg\max_{p_\theta(y)} \int_y [\ln p_\theta(x,y)]\, p_{\theta'}(y|x) dy,$$
- **M-step**: $$\theta' \leftarrow \arg\max_{\theta} \int_y [\ln p_\theta(x,y)]\, p_{\theta'}(y|x) dy$$

previous iteration θ' and in the M-step the lower bound is maximized with respect to the model parameters θ. The Em algorithm is summarized in Algorithm 5.5.

The E-step consists in maximizing $Q(\theta, \theta')$ with respect to the hidden variable distribution. There exists a closed form solution to this maximization problem: $p_{\theta'}(y) \leftarrow p_{\theta'}(y|x)$. As for the M-step, in simple cases, the maximization is written in closed form, but often this is not the case. In practice it is enough to augment the lower bound $Q(\theta, \theta')$ to ensure that the likelihood increases at each iteration, therefore when necessary the M-step can be written as a maximization following the gradient direction. Under some regularity conditions of the likelihood function, the estimated model parameters correspond to a local maximum of the likelihood. Details and proves can be found in e.g.[81, 55].

Often the likelihood is multi-modal, the EM converges to a local maximum which depends on the initialization conditions. Therefore to obtain a reliable estimate of the parameters it is often necessary to run the algorithm with several initializations, or to provide a good initialization.

2.4.2 Template Estimation in the case of Landmark Detection

Often the template is a deterministic function from Λ_T to $\mathbb{R}$ (or from Ω_T to $\mathbb{R}$), i.e. an image. It is most of the time considered as a model parameter, therefore the training of the model consists in optimizing some cost function (e.g.the likelihood of the training data) with respect to the template. It is a crude assumption though to assume that all the images of the training set come from a single deterministic template. Therefore we model the template as a hidden variable and learn its distribution from the training data. In an abuse of language, we will call probabilistic template the distribution of the template. The probabilistic template is a function that assigns at each node t in the finite grid Λ_T a probability distribution $\pi(t)$, interpolated to the rest of the domain Ω_T.

In the case of landmarks detection, the training set contains some landmark correspondences, which are very few compared to the size of the image and very sparsely distributed compared to the complexity of the deformation that perfectly map the template to the images. By working with a probabilistic template rather than with a deterministic template, we capture both the variability coming from the different geometries which is not captured by the landmark correspondences but also the intensity variations and the localization error of the landmarks.

Probabilistic templates are often called probabilistic atlases. They are commonly encountered in brain image segmentation to provide some prior information such as the

distribution of the tissue types [49, 61, 86, 27, 4, 62]. This type of atlases is most of the time obtained by computing the proportions of each tissue type or the intensity distribution at each pixel across registered (segmented) images. The registration method varies but is often affine, creating a blurred atlas of the tissue types, see e.g.[23]. In [47] the author focuses on defining an atlas as a set of labeled nodes on a triangle mesh. The label carries the information of the tissue type distribution at that location and the rest of the template domain is defined by linear interpolation on the triangulation. The tissue distribution and the location of each node are obtained by the Expectation Maximization algorithm. In this case the hidden variables are the segmentation in the template at each node as well as the deformation of the mesh from the template domain to each of the training image.

CHAPTER 3

DEFORMABLE INTENSITY MODEL

Deformable models have been proposed to solve numerous problems in computer vision and medical imaging. Examples of successful applications are in image segmentation, image registration and object detection. Many combinations of energy functions, deformation models and regularity constraints have been explored. While some of these combinations have achieved good performance, the need of tailoring the solution to each problem prevents from developing generic deformable models.

The usage of statistical deformable models though allows one to derive from the modeling assumptions the appropriate cost function and algorithm for training and testing on a variety of image related applications such as segmentation and/or registration.

In the following three chapters, we will propose a family of statistical deformable models. We illustrate how these statistical models can be used to derive intuitive yet mathematically sound algorithms to solve the automatic landmark detection problem.

In this chapter we focus on an intensity-based model comparable to the classical matching techniques using sum of square intensity differences as a cost. The model selection as well as the landmark detection is performed by likelihood maximization using the proposed intensity model, leading to an intensity matching algorithm for landmark detection. Performance are assessed on the detection of 4 landmarks on two sets of 2D brain MRI.

3.1 Previous Work: Image Registration

While the problem of image registration has received a lot of attention, statistical models for registration and image warping have only rarely been explored. Some attempts of developing a statistical framework for image warping have been made, [65, 33]. Most of the time an image is modeled as the deformed image x of a template x_0 by some random bijective deformation f, assuming that the target image intensity and the template differs by a Gaussian noise:

$$\forall s \in \Lambda, \quad x(s) = x_0(f^{-1}(s)) + \epsilon(s), \quad \text{with } \epsilon(s) \sim \mathcal{N}(0, \tau^2), \tag{3.1}$$

with $x(s)$ the random variable of the image intensity at pixel s, $x_0(t)$ the intensity in the template at location t, and f the random registering deformation, i.e. such that x similar to $f \cdot x_0$. In other terms the conditional probability of the random variable representing the intensity at pixel s, $x(s)$, given the random transformation f is:

$$\forall s, \quad \Big(x(s)|x_0(f^{-1}(s))\Big) \sim \mathcal{N}(x_0(f^{-1}(s)), \tau^2). \tag{3.2}$$

Since the righthand side of (3.1) follows a Gaussian distribution, the conditional probability becomes at each pixel:

$$\forall s, \quad p(x(s)|f) \propto \exp\left(-\frac{|x(s) - x_0(f^{-1}(s))|^2}{2\tau^2}\right). \tag{3.3}$$

For the whole image x, assuming conditional independence of the pixel,

$$p(x|f) \propto \exp\left(-\frac{\sum_{s\in\Lambda} |x(s) - x_0(f^{-1}(s))|^2}{2\tau^2}\right). \tag{3.4}$$

Given two images, a source image x_0 and a target image x, this model provides a way to estimate using a likelihood maximization scheme the "best" deformation from the source image to the target.

$$\hat{f} = \arg\max_{f\in\mathcal{F}} \ln p(x, f), \tag{3.5}$$

$$= \arg\min_{f\in\mathcal{F}} \sum_{s\in\Lambda} |x(s) - x_0(f^{-1}(s))|^2. \tag{3.6}$$

The deformation that minimizes the sum of squared intensity difference (SSD) of the two images corresponds to the maximum likelihood estimate. The cost function derived from the model corresponds exactly to the SSD cost function proposed first in [6]. SSD has since then been broadly used for image matching and tracking in video sequences, and is considered as a benchmark for other techniques.

3.1.1 Landmark Detection

As explained in Chapter 2, the landmark detection problem can be seen as a local registration problem. However, the particularity of our problem comes from the existence of a training set of labeled images. Therefore the whole landmark detection is composed of two subproblems: learning the model parameters based on the training images and then estimating the location of the landmarks in a new image using the previously learnt model. We denote $\theta \in \Theta$ the model parameters, $x_1^N \in \mathbb{R}^{SN}$ the training set of N images, $y_1^N \in \mathcal{Y} \subset \mathbb{R}^{dKN}$ the location of the landmarks in the training images and $x \in \mathbb{R}^S$ a new image. In term of likelihood, the problem consists first in selecting the model parameters:

$$\hat{\theta} = \arg\max_{\theta\in\Theta} \ell(x_1^N, y_1^N; \theta). \tag{3.7}$$

Finding the landmark location for a new image consists then in solving:

$$\hat{y} = \arg\max_{y\in\mathcal{Y}} \ell(x, y; \hat{\theta}). \tag{3.8}$$

Because the landmark locations are given in the training set, the registering deformation are known. Thus the model estimation is straightforward.

3.2 Deformable Intensity Model

In our case we are interested in building a model of the joint probability $p(x, y)$ of the image x and the location of the landmarks y using a probabilistic deformable template, learnt from training data. Depending on the problem and the type of images, one may consider to model the image intensities directly or else use some other image descriptors. In this chapter we work exclusively with the intensity, but numerous types of features could be used instead of the image intensity. The analysis remains essentially the same regardless of the choice of the image feature used, as long as the feature is directly observable in the images. It might require though to be able to build statistics on non-Euclidean spaces. For example if one considers oriented edges, the orientation distribution lies on a sphere. Similarly if one chooses the local structure tensor, it is necessary to build statistics on the corresponding manifold. The ability to deal with features other than scalar makes it possible to extend the method to other types of data such as Diffusion Tensor Images.

Using Bayes' formula, the joint probability of the image x and the location of the landmarks y is

$$p(x, y) = p(x|y)p(y). \tag{3.9}$$

As it is often the case in generative models of images, we will assume statistical independence of the image intensities given the location of the landmarks such that the conditional probability can be written as a product over the image support. Assuming that the image is defined on a finite grid Λ of $\mathbb{R}^d$, we have

$$p(x, y) = \prod_{s \in \Lambda} p(x(s)|y)p(y). \tag{3.10}$$

This assumption ignores the spatial correlation of the noise. Hence, if one generates an image using this model, it is not possible to reconstruct smooth images like the training instances due to the independence of the noise at each pixel.

In the previously described Gaussian model, the noise τ is a global parameter of the model and is independent from the location in the image. In addition, the template or source image is a deterministic function from Λ_T to $\mathbb{R}$. In our approach we choose to work with probabilistic templates, because we believe that the deformations defined by few landmarks are not enough to model the geometric variability of the images. Using a probabilistic template allows us to capture both the photometric and the geometric variations at each pixel. Recall that in the Gaussian model the intensity value at a pixel s is modeled with:

$$\forall s, \quad x(s) = x_0(f_y^{-1}(s)) + \epsilon(s), \quad \text{with } \epsilon(s) \sim \mathcal{N}(0, \tau^2), \tag{3.11}$$

with f_y the deformation that maps the landmark location of the template $\bar{y}$ to y in the image. In the probabilistic deformable intensity model we propose to model the intensity value with:

$$\forall s, \quad x(s) = x_0(f_y^{-1}(s)) + \epsilon(s), \quad \text{with } \epsilon(s) \sim \mathcal{N}(0, \tau^2(f_y^{-1}(s))). \tag{3.12}$$

It means that the noise model is defined on the template grid Λ_T and is assumed to have different standard deviation at each pixel. As a consequence, the corresponding likelihood function is similar to the likelihood function obtained with the Gaussian model, except that the intensity difference is normalized by a pixel-specific variance and the normalization constant now depends on the location:

$$\ell(x,y) = -\sum_{s\in\Lambda} \frac{(x(s) - x_0(f_y^{-1}(s)))^2}{2\tau^2(f_y^{-1}(s))} - \sum_{s\in\Lambda} \log \tau^2(f_y^{-1}(s)) - \sum_{s\in\Lambda} \frac{1}{2}\log 2\pi + \log p(y). \tag{3.13}$$

The log-likelihood is maximal if the deformation f_y^{-1} maps the template (or source image) to a region of similar intensity in the image (or target image). This method relies on the intensity similarity, therefore if the target image intensity distribution differs significantly from the one of the template, the matching result could be incorrect. That is why before using this model, the image intensity is normalized in a way that source and target images have similar intensity ranges. This preprocessing can be an important limitation to automatic analysis of medical images, because there are often outliers in these images which prevent full automation of the intensity normalization.

3.3 Model Selection using a Training Set

Model selection consists in learning the parameters θ of the deformable model from the training set of images $(x^{(i)}, y^{(i)})_1^N$. The model is composed of two sets of parameters: the template distribution parameters $(\forall t \in \Lambda_T,\ x_0(t),\ \tau(t))$ and the landmark prior distribution $p(y)$. Considering the landmarked images as independent samples of $p(x,y)$, the likelihood of the training set is:

$$\begin{aligned}\ell_{tot}(x_1^N, y_1^N; \theta) &= \ell(x_1^N | y_1^N; x_0, \tau^2) + \ell(y_1^N; p(y)) \\ &= \sum_{i=1}^{N} \left[-\sum_{s\in\Lambda} \frac{(x^{(i)}(s) - x_0(f_{y^{(i)}}^{-1}(s)))^2}{2\tau^2(f_{y^{(i)}}^{-1}(s))} - \sum_{s\in\Lambda} \log \tau^2(f_{y^{(i)}}^{-1}(s)) - \sum_{s\in\Lambda} \frac{1}{2}\log 2\pi + \log p(y^{(i)}) \right].\end{aligned} \tag{3.14}$$

The likelihood function is a sum of two independent terms, therefore the optimization with respect to the template and with respect to the prior distribution of the landmarks can be performed independently from each other.

3.3.1 Direct Estimation of the Deformable Model

We recall the method to generate images from the probabilistic deformable intensity model. A random image is sampled from the template, which contains the conditional probabilities $p(x|y)$ and a landmark location is drawn from $p(y)$. The transformation set $\mathcal{F}$ is chosen such that each landmark configuration corresponds to a unique deformation of $\mathbb{R}^d$, as detailed in Chapter 2. The random image is deformed using the transformation f_y to produce the final image in which the landmarks lie in y.

The template is learned by likelihood maximization with respect to (x_0, τ):

$$\ell(x_1^N|y_1^N; x_0, \tau^2) = \sum_{i=1}^{N} \sum_{s\in\Lambda} \ln p(x^{(i)}(s)|y^{(i)}). \tag{3.15}$$

Using the deformable model assumption,

$$\forall s, \quad x^{(i)}(s)|y^{(i)} \sim \mathcal{N}(x_o(t), \tau^2(t)) \quad \text{with } t = f_{y^{(i)}}^{-1}(s). \tag{3.16}$$

We denote $\pi(u,t)$ the probability to observe at $t \in \Lambda_T$ the intensity value u. The conditional likelihood function becomes,

$$\ell(x_1^N|y_1^N; x_0, \tau^2) = \sum_{i=1}^{N} \sum_{s\in\Lambda} \ln \pi(x^{(i)}(s), f_{y^{(i)}}^{-1}(s)). \tag{3.17}$$

Two issues arise from this expression. Firstly, for a fixed $s \in \Lambda$, $f_{y^{(i)}}^{-1}(s)$, the pre-image of pixel s, depends on the image since the deformation $f_{y^{(i)}}$ is image specific. Secondly, there are cases in which even though the deformation is invertible, f_y^{-1} does not have a simple analytic form. This is the case with the spline based deformations such as the one introduced in 2.3.2.

Remark 3.1. *In the case of rigid transformations, since we assume that the number of landmarks and the number of degrees of freedom coincide, there exists a unique correspondence between a set of landmarks and a transformation. Often the inverse transformation is explicit. Yet the location $f_{y^{(i)}}^{-1}(s)$ depends on the image. Therefore, in order to estimate the template, one needs to perform some approximations even for rigid transformation.*

We propose to approximate the likelihood function by performing a change of variable, such that instead of working at a fixed location $s \in \Lambda$ in the target, we will be referring to a fix location $t \in \Lambda_T$ in the template. For the sake of simplicity we assume that the size of a pixel (or voxel) is $1mm^2$ (or $1mm^3$). The sum over the pixel of the image is approximated with the integral over the support of the image with respect to ds the area (or volume) of integration,

$$\ell(x_1^N|y_1^N; x_0, \tau^2) \approx \sum_{i=1}^{N} \int_{\Lambda_i} \ln \pi(x^{(i)}(s), f_{y^{(i)}}^{-1}(s))ds. \tag{3.18}$$

For each image i, we perform the change of variable $s = f_{y^{(i)}}(t)$, and denote by $|J_{f_{y^{(i)}}}(t)|$ the absolute value of the deformation Jacobian at t.

$$\ell(x_1^N|y_1^N, x_0, \tau^2) = \sum_{i=1}^{N} \int_{f_{y^{(i)}}^{-1}(\Lambda_i)} \ln \pi \left(x^{(i)}(f_{y^{(i)}}(t)), t\right) |J_{f_{y^{(i)}}}(t)|dt. \tag{3.19}$$

We assume that for all image i, the finite grid $f_{y^{(i)}}^{-1}(\Lambda_i)$ covers the same domain Ω_T as the regular grid of the template Λ_T. Finally we approximate the function by resampling it along the regular grid of the template Λ_T and discretize it:

$$\ell(x_1^N|y_1^N; x_0, \tau^2) = \sum_{i=1}^{N} \sum_{t\in\Lambda_T} \ln \pi \left(x^{(i)}(f_{y^{(i)}}(t)), t\right) |J_{f_{y^{(i)}}}(t)|. \tag{3.20}$$

The above approximation of the likelihood function will appear regularly in the estimation of the model, from now on we will refer to it as the "approximated integral change of variable". Not only this approximation allows us to avoid the inverse of the deformation but overall it transforms the joint optimization with respect to all the pixel parameters in as many independent problems as pixels in the finite lattice Λ_T. The likelihood optimization becomes separable in independent maximization with respect to each of the Gaussian distribution, parameterized by $(x_0(t), \tau^2(t))$.

By fixing the position t, the observations need to be taken at $f_{y^{(i)}}(t)$ which does not necessarily lie on the initial discrete grid of the image. Linear interpolation on the original image is used to recover the missing intensity values. The interpolation technique must be adapted to the type of image feature.

After the change of variable, the log-likelihood function becomes

$$\begin{aligned}&\sum_{i=1}^{N} \ln \pi \left(x^{(i)}(f_{y^{(i)}}(t)), t \right) |J_{f_{y^{(i)}}}(t)| \\ &= \sum_{i=1}^{N} \left[-\frac{1}{2} \ln 2\pi - \frac{1}{2} \ln \tau^2(t) - \frac{|x(f_{y^{(i)}}(t)) - x_0(t)|^2}{2\tau^2(t)} \right] |J_{f_{y^{(i)}}}(t)|, \end{aligned} \tag{3.21}$$

and its maximization in each pixel t with respect to $x_0(t)$ and $\tau(t)$ admits closed form solutions:

$$\forall t \in \Lambda_T, \quad \hat{x}_0(t) = \frac{\sum_{i=1}^{N} x(f_{y^{(i)}}(t)) |J_{f_{y^{(i)}}}(t)|}{\sum_{i=1}^{N} |J_{f_{y^{(i)}}}(t)|}, \tag{3.22}$$

$$\forall t \in \Lambda_T, \quad \hat{\tau}^2(t) = \frac{\sum_{i=1}^{N} \left[x(f_{y^{(i)}}(t)) - x_0(t) \right]^2 |J_{f_{y^{(i)}}}(t)|}{\sum_{i=1}^{N} |J_{f_{y^{(i)}}}(t)|}. \tag{3.23}$$

The MLE expression is similar to the classical MLE of a Gaussian sample, except that each sample is weighted depending on whether the region shrinks or expands during the registration of the training images onto the template. If the Jacobian is locally equal to 1, it is simply averaging the intensity values, after registration.

Remark 3.2. *Most of the time in the literature, the deformation is defined from the image to the template. The estimation of the template is simply seen as a two step procedure. First, the images are registered and then the registered images are average pixel by pixel. There are then no weight coefficients in the template estimation.*

3.3.2 Learning the Distribution of the Landmark Locations

Since the location of the landmarks is observed in the training set, one can use the training sample locations to learn the marginal distribution of the landmark locations. However, if the number of landmarks increases, but the number of samples remains small, it may become necessary to introduce a prior on the landmark covariances in order to avoid overfitting the training samples.

Uniform Prior Landmark Location Distribution In anatomical landmark detection, it is often the case that the images are first globally aligned. In the case of brain analysis, often the image are aligned to the Talairach space [45], which brings brains to same orientation and scale. As a consequence, using the same global orientation, it is possible to build a prior distribution on the location of the landmarks in the image, because the anatomy is globally the same even for different individuals. Thus it is possible to build a simple prior on the landmark location, using a set of subsets $(\Lambda(1), \ldots, \Lambda(K))$ of the image domain Λ, such that each landmarks follows a Uniform distribution on the corresponding subset of image domain. More specifically, if we consider K statistically independent landmarks, we obtain

$$p(y) = \prod_{k=1}^{K} p(y_k), \quad y_k \sim \mathcal{U}(\Lambda(k)), \quad \Lambda(k) \subset \Lambda. \tag{3.24}$$

Covariance Estimation Beyond the prior on the location, it would be informative to build a prior on the covariance structure of the landmark locations. However, the number of training landmark configurations N is usually small in comparison with the dimensionality of the space in which the landmark configurations live in $\mathbb{R}^{dK}$, where d is the dimension of the image and K is the number of landmarks. Assuming the landmark location follows a Gaussian distribution $\mathcal{N}(\bar{y}, \Sigma_y)$, with $\bar{y} \in \mathbb{R}^{dK}$ and Σ_y a $dK \times dK$ positive definite matrix, it is tedious in the small sample setting to obtain an accurate estimate of the covariance matrix Σ_y. The empirical estimates of the mean and variance of the distribution are given by

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}, \quad \text{and} \quad \hat{\Sigma}_{\text{emp}} = \frac{1}{N-1}\sum_{i=1}^{N}(y^{(i)} - \bar{y})(y^{(i)} - \bar{y})^{\top}. \tag{3.25}$$

If $N \geq dK + 1$, but N small, the sample variance S is non-singular but provides a poor estimate of the true covariance matrix. If $N \leq dK$ the sample estimate $\hat{\Sigma}_{\text{emp}}$ is singular and therefore unusable.

Estimation by Shrinkage of the Eigenvalues The objective of this technique is to ensure positiveness and definiteness of the covariance matrix estimate. Because only few samples are available for the empiric estimation, the largest eigenvalues tend to be overestimated while the smallest eigenvalues are underestimated. Therefore by regularizing using a diagonal matrix containing the average eigenvalue on the diagonal, the largest eigenvalues are reduced and the smallest increased. The regularized estimate is:

$$\hat{\Sigma}_y(\lambda) = (1-\lambda)\hat{\Sigma}_{\text{emp}} + \lambda \left(\frac{tr(\hat{\Sigma}_{\text{emp}})}{dK} \right) I_{dK}, \tag{3.26}$$

with λ the weight parameter that controls the trade-off of the regularization term and the data term. This regularization method was also used for example in [30] in order to estimate covariance matrices for Robust Linear Discriminant Analysis.

Bayes Estimator using an Inverted Wishart Prior Alternatively, Greene et al.[37] and Tadjudin et al.[72] use a Bayesian formulation to estimate the covariance matrix. According to the multidimensional Fisher theorem the covariance estimate of a Normal distribution $\hat{\Sigma}_{\text{emp}}$, follows a Wishart distribution:

$$\hat{\Sigma}_{\text{emp}} \sim \mathcal{W}\left(\frac{1}{N-1}\Sigma_y, N-1\right) \tag{3.27}$$

where $\mathcal{W}$ denotes the central Wishart distribution with $N-1$ degrees of freedom and parameter matrix $\frac{1}{N-1}\Sigma_y$. The family of inverted Wishart distribution is a convenient conjugate prior for the true covariance Σ_y. Assuming Σ_y follows an inverted Wishart distribution:

$$\Sigma_y \sim \mathcal{W}^{-1}\left(a\Sigma_0, a+dK+1\right), \quad a > 0,$$

where $\mathcal{W}^{-1}$ is the inverted Wishart distribution, the prior mean is Σ_0 and its concentration around the mean is controlled by $a+dK+1$.

The Bayes estimator of Σ_y is given by

$$\hat{\Sigma}_y = \frac{(N-1)\hat{\Sigma}_{\text{emp}} + a\Sigma_0}{N-1+a}$$

Notice that the Bayes estimator consists of biasing the sample covariance towards the mean of the inverted Wishart distribution. a controls the trade-off between the prior information and the data.

In the case of landmarks location, the matrix parameter Σ_0 encodes the landmarks correlation, that we can choose to be decreasing at larger distance, such that close landmarks have a priori larger covariance.

3.4 Local Intensity Matching for Landmark Detection

The objective of landmark detection is to predict the value of y in a new image x using the model learnt on the training set $(x_0, \tau^2, p(y))$.

In the case of a generative model, it is possible to estimate y using the Bayes' estimate, which consists in maximizing the posterior probability of the landmarks,

$$\hat{y} = \arg\max_y p(x|y)p(y). \tag{3.28}$$

If we use an improper flat prior for y, i.e. a prior that does not contain any information on the location of the landmarks, the Bayes' estimator and the maximum likelihood estimator coincide.

Remark 3.3. *The derivation of the estimation algorithm is obtained for an improper prior, but notice that the prior on y would simply add a term to the matching cost function, acting as a penalization on the likelihood. The resulting estimator varies depending on the choice of the prior but the data term contribution remains unchanged.*

The log-likelihood of the new gray scale image is

$$\ell(x|y;\hat{x}_0,\hat{\tau}^2) = -\frac{1}{2}\sum_{s\in\Lambda}\left[\ln 2\pi + \ln\hat{\tau}^2(f_y^{-1}(s)) + \frac{|x(s)-\hat{x}_0(f_y^{-1}(s))|^2}{\hat{\tau}^2(f_y^{-1}(s))}\right]. \tag{3.29}$$

The landmark location is chosen to maximize the likelihood function:

$$\hat{y} = \arg\max_{y\in\mathcal{Y}} \ell(x|y;\hat{x}_0,\hat{\tau}^2), \tag{3.30}$$

Local Intensity Matching Notice that the likelihood increases if the intensities of the deformed template matches the intensities of the image. In the case of matching using SSD, the noise parameter τ is constant throughout the template, assigning the same weight to each pixel of the image. Because the variance in the Deformable Intensity Model (DIM) varies depending on the location in the template, the pixels with lower variance have greater weight in the cost function than the pixels for which the intensity variance is large. Lower variance appears in the homogeneous parts of the image since even slightly misaligned the intensity across images would match in the center of large homogeneous regions. On contour though, the intensity will vary significantly more as we will observe a mixture of intensities. Around the landmarks though, the contour are well aligned since the template is learnt from the locally aligned images based on the landmark correspondences. As a consequence the pixels surrounding the landmarks have the most important contribution to the cost function, since locally the intensity distribution has lower variance. The homogeneous parts, even though they correspond to low variance intensity distribution do not contribute significantly to the variation of the likelihood. In consequence the cost function specializes in matching the intensity around the landmarks. That is why we call it a "local intensity matching" method.

It is important to notice that similarly to the SSD matching method, if the intensity distribution in the image differs significantly from the intensity distribution of the template, the best match in terms of likelihood may not be the geometric match.

Optimization Method The optimization is performed by a steepest gradient ascent combined with a line search method to determine the size of the step at each iteration. We initialize the gradient ascent with the identity deformation, which brings the reference landmarks at the same location in the image:

1. Initialize the gradient ascent with $y \leftarrow \bar{y}$,
2. Iterate until convergence:
   (a) Compute $\nabla_y \ell(x,y;\hat{x}_0,\hat{\tau}^2,\hat{p}(y))$,
   (b) Find $a \geq 0$ such that $a \leftarrow \arg\max_{a\geq 0} \nabla_y \ell(x,y;\hat{x}_0,\hat{\tau}^2,\hat{p}(y))$,
   (c) $y \leftarrow y + a\nabla_y \ell(x,y;\hat{x}_0,\hat{\tau}^2,\hat{p}(y))$.

Computation of the Likelihood Gradient We have seen in Chapter 2.3.2 that spline-based deformations have numerous advantages for landmark detection. Recall that the expression of these deformations is of the form:

$$\forall t, \quad f_y(t) = \sum_{k=1}^{K} \kappa(t, \bar{y}_k)\beta_k, \tag{3.31}$$

with $\beta_k \in \mathbb{R}^d$ determined by the landmark matching constraints $f_y(\bar{y}) = y$. Even though we choose the kernel κ such that f_y is invertible, for $K > 1$, f_y^{-1} does not have a simple analytical form preventing the exact computation of the likelihood gradient. Therefore the optimization of the likelihood, as aforementioned, is not tractable with this set of deformations. A solution consists of using the "approximated integral change of variable" $t = f_y^{-1}(s)$ in the likelihood expression, which gives:

$$\ell(x|y; \hat{x}_0, \hat{\tau}) \approx -\frac{1}{2} S \ln 2\pi - \frac{1}{2} \sum_{t \in \Lambda_T} \left[\ln \tau^2(t) + \frac{|x(f_y(t)) - x_0(t)|^2}{\tau^2(t)} \right] |J_{f_y}(t)|. \tag{3.32}$$

Hence, the terms to be derived are now on one hand the intensity value in the image and on the other hand the Jacobian of the deformation. Without entering in the details of the computation, it is possible to obtain an analytical expression of the Jacobian gradient with respect to y. As for the intensity function, we consider it as a continuous function $x : \mathbb{R}^d \to \mathbb{R}$ such that the derivative of the composition is:

$$\frac{\partial x(f_y(t))}{\partial y_{kl}} = \langle \frac{\partial x}{\partial c_l}(f_y(t)), \frac{\partial f_y^{(l)}}{\partial y_{kl}}(t) \rangle, \tag{3.33}$$

with $\frac{\partial x}{\partial c_l}(f_y(t))$ the derivative of x with respect to the lth cartesian coordinate and $\frac{\partial f_y^{(l)}}{\partial y_{kl}}(t)$ the partial derivative of the lth coordinate of the deformation with respect to the lth coordinate of the kth landmark.

The complete gradient expression is:

$$\begin{aligned} \frac{\partial \ell(x|y; \hat{x}_0, \hat{\tau})}{\partial y_{kl}} = & -\frac{1}{2} \sum_{t \in \Lambda_T} \left[\ln \tau^2(t) + \frac{\left(x(f_y(t)) - x_0(t)\right)^2}{\tau^2(t)} \right] \frac{\partial |J_{f_y}(t)|}{\partial y_{kl}} \\ & - \sum_{t \in \Lambda_T} \frac{x(f_y(t)) - x_0(t)}{\tau^2(t)} \, |J_{f_y}(t)| \, \langle \frac{\partial x}{\partial c_l}(f_y(t)), \frac{\partial f_y^{(l)}}{\partial y_{kl}}(t) \rangle. \end{aligned} \tag{3.34}$$

This approximation of the gradient requires to use linear interpolation on the image to estimate $x(f_y(t))$ for all possible values of y and t.

3.5 Numerical Validity of the Approximations

In order to verify the numerical validity of the approximated integral change of variable (cf 3.3.1), we perform a simple experiment, in which we compare the numerical value

$$A = \sum_{s \in \Lambda} x(s) \tag{3.35}$$

$y - \bar{y}$	A	B	C	D
[0.5 0.5]	390.63	388.96	388.53	**390.75**
[2.0 2.0]	380.63	377.20	**381.54**	390.62
[4.0 4.0]	370.50	367.61	**369.75**	390.62

Table 3.1: Numerical approximations of the image integral

with the approximated functions:

$$B = \sum_{t \in f_y^{-1}(\Lambda)} x(f_y(t))|J_{f_y}(t)|, \quad \text{and} \quad C = \sum_{t \in \Lambda_T} x(f_y(t))|J_{f_y}(t)|. \tag{3.36}$$

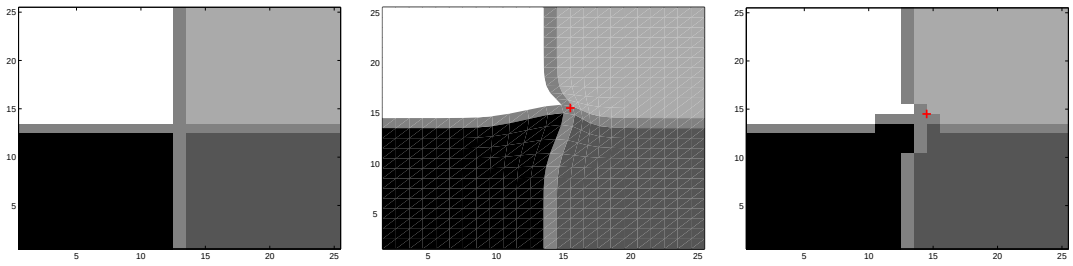
Both B and C are obtained by approximating the discrete function A as an integral in which a change of variable is performed. In both cases the integral is discretized again for the final computation. While B corresponds to the sum at each $t \in f_y^{-1}(\Lambda)$, C is computed after resampling on the regular grid Λ_T.

To be complete and to assess whether the computation of the Jacobian is useful, we also compare the function C with $D = \sum_{t \in \Lambda_T} x(f_y(t))$. We expect that as the deformation becomes larger and "less rigid", the difference between D and A will increase.

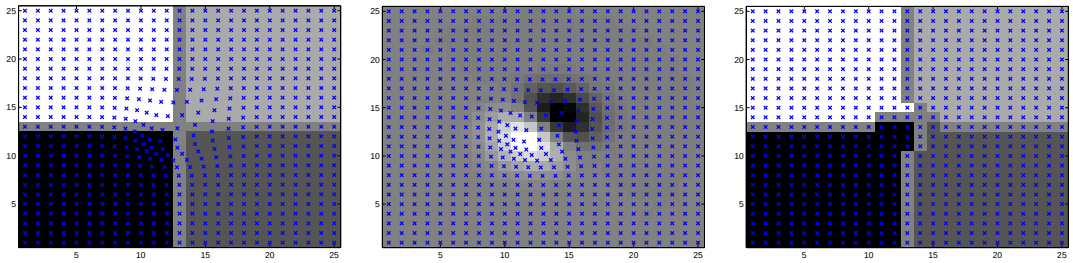
We perform an experiment on a 25×25 pixels image which is deformed by a Gaussian spline of variance 4 driven by the displacement of one landmark. The displacement varies between 0.5 and 4 pixels along each axis. The deformation does not affect the image boundary. Figure 3.1 represents the different computations performed on a deformed image of the template. The blue crosses in 3.1(b) and 3.1(c) represents the location of the center of each pixel used to approximate the integral function, either on the template or on the image support.

Table 3.1 presents the numerical values of A or of its approximations B, C, and D. First notice that A the image integral decreases in our experiment when the deformation becomes larger. This is because the regions with high intensity shrink under the action of the deformations we used while the region with lower intensities on the contrary are expending. By comparing A to its approximations B, C we measure the effect of the change of variable and resampling on a regular grid. It turns out that the numeric error is lower than 1 percent of the likelihood value. This is negligible compared to the variations of the likelihood function when the deformation varies in our experiments. It is also visible in Table 3.1 up to some extent only because in this numerical validation, we used quite homogeneous images, which limits the variations of the image integral. As expected though, it looks like the Jacobin has an important role when the deformation becomes larger. Indeed the difference between A and D increases when the deformation becomes larger.

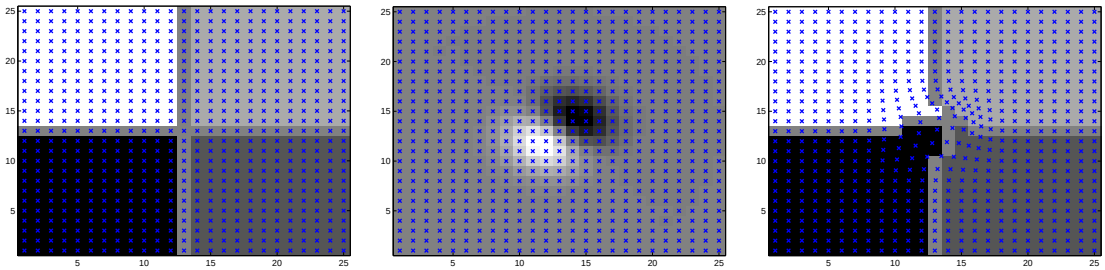
While this experiment does not allow us to conclude in every situation it is an indication that the proposed approximation which consists in approximating the image function as an integral over the pixels, change the variable and resample on a regular grid, is a reasonable compromise between computational practicality and precision.



(a) Deformable Template and Discretization. Left: the intensity template, Center: the deformed template, Right: the resulting discretized image, used to compute (A).



(b) Discrete approximation of the integral (B). Left: Irregular sampling in the template, Center: Irregular sampling of the Jacobian function, Right: Regular sampling of the image.



(c) Approximated function: after change of variable and resampling (C). Left: Regular sampling of the template, Center: Regular sampling of the Jacobian function, Right: Irregular sampling of the image.

Figure 3.1: Numerical approximation of an image function using a discretized integral with or without resampling. Top line: represents a non-rigid deformation of the template and the corresponding discretized image. Middle line: represents the approximation of the discrete sum. Bottom Line: approximation of the discrete sum with resampling.

3.6 Detection Results

In this section we present some experiments performed on medical images using the Deformable Intensity Model.

3.6.1 Description of the Images

We use 47 T1-weighted Magnetic Resonance (MR) brain images acquired on a Philips-Intera 3-Tesla scanner (MPRAGE), with resolution 1mm^3, encoded in gray-level intensity from 0 to 1462. Brains were first manually transformed into standardized Talairach space [45] using Analysis of Functional Neuroimages (AFNI) [17] to provide a canonical orientation (anterior and posterior commissures (AC and PC) made co-linear) and approximate

alignment. The purpose of this step is to surround the entire brain within a grid system, so that all the scans can be aligned in the same position. The registering piecewise affine transformation relies on the manual detection of the anterior and posterior commissures as well as landmarks on the cortical periphery. The resulting image has a fixed volume of $161 \times 191 \times 151$ voxels in our case.

Brains were viewed in continuously synchronized sagittal, axial, and coronal planes. Dr. Craig Stark from the Department of Psychological and Brain Sciences at the Johns Hopkins University, provided the images and located the landmarks in the database of images, obtained on healthy individuals. Two sets of landmarks have been located in the image of each patient individually: the extremity of the splenium of the corpus callosum, which lies in the center of the brain and the left hippocampus which is one of the structure of the temporal lobe.

The manual landmarking procedure for locating the Splenium of the Corpus Callosum starts by identifying the mid-sagittal plane because the best visualization is given by the sagittal view. The extremity of the splenium (SCC1) is defined as the most posterior extent of the corpus callosum. While the most posterior extent of the corpus callosum often lie long the mid-sagittal slice, in several instances, it lies at few millimeters from that slice. In these cases, the splenium was identified as the most posterior extent in this slice.

The localization of the landmarks along the hippocampus is slightly more tedious. The set of landmarks is composed of 3 main landmarks (Head and Tail of the Hippocampus and the Uncus Apex) and 12 lateral landmarks. The landmarking procedure starts with identifying the head of the hippocampus (HoH). Coordinated coronal and sagittal views were used to identify the quasi-invisible white matter line separating the hippocampus from the amygdala, the surrounding structure. HoH is defined as the furthest extent of the hippocampus in the anterior and inferior directions. It is equally difficult to define and locate consistently the extremity of the hippocampus tail, as the structure slowly fades away from the image along the sagittal axis. To ensure its consistent landmarking we define the tail of the hippocampus (HT) as the furthest extent of the hippocampus (in posterior and superior directions) on the sagittal slice that contains the head of the hippocampus. The Uncal Apex (UA) marks the separation between the head and the body of the hippocampus, [22]. It is identified by spanning the image along the coronal axis and marking the point where the uncus first appears. Once these three landmarks (HoH, HT and UA) have been located, the main hippocampal axis is defined by the segment HoH-HT. Three orthogonal planes are defined along this axis, on which the lateral landmarks are identified at the boundary of the hippocampus and other structures. O1R, O1L, O1S and O1I respectively refer to the right, left, superior and inferior extremity of the hippocampus in the first transversal slice.

In order to reduce the computational load, we extract subvolumes from the whole brain images around the regions of interest. A first set of images of size $45 \times 50 \times 50$, denoted 3D-SCC, is extracted from the region around the splenium of the corpus callosum. A second set of images of size $50 \times 66 \times 50$, denoted 3D-Hippo is extracted around the left hippocampus.

In order to better assess the algorithm performance, we simplify further the problem by extracting 2D sagittal images containing either SCC1 (denoted then 2D-SCC) or the head

and tail of the hippocampus (denoted 2D-Hippo). To be able to test easily the algorithm on 2 surrounding landmarks we identify SCC2 which is defined as the most inferior point of the Splenium of the Corpus Callosum on the 2D slice containing SCC1.

Because the images were acquired with different contrast settings, they have very variable intensity ranges. Because the intensity model is sensitive to the intensity variation, the images are normalized such that their intensity are between 0 and 255, with a median of 125. A set of 17 images, sampled randomly from the data set, is kept on the side of the training phase and used for independent testing of the learnt model. The testing set is the same for all the experiments in the current and following chapter.

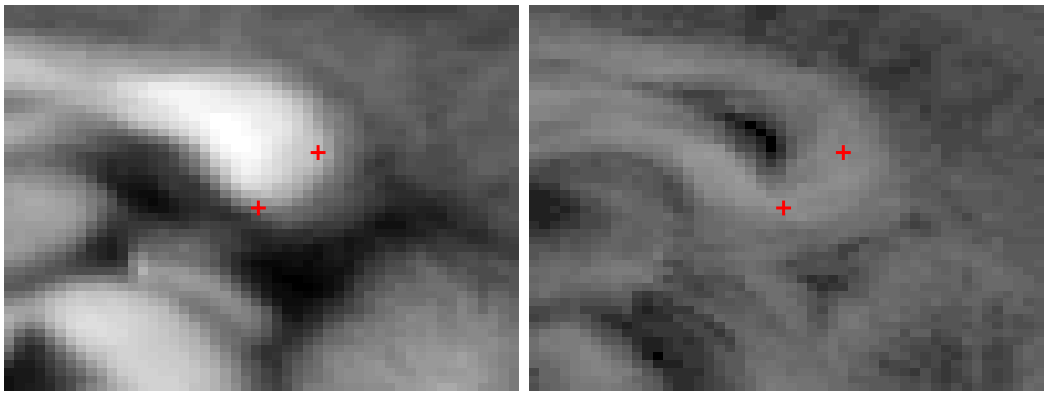
3.6.2 Detection in Brain Magnetic Resonance Images

Model Estimation

The model requires that one chooses manually a deformation model, its parameters and a prior on the landmark locations. In our experiments we use a constant prior and choose a Gaussian kernel with $\sigma = 7$. We work simultaneously on the detection of the two landmarks SCC1 and SCC2 in the data set 2D-SCC. We compare the estimated template depicted in Subfigures 3.3(a) and 3.3(b), to the average and standard deviation of the stack of training images before registration, shown in Subfigures 3.2(a) and 3.2(b). The edges around the landmarks are sharper in the estimated template than in the average. This is because during learning the images are registered based on the landmarks correspondences. Since the Gaussian spline model has local support only, it is only around the landmarks that the registration is visible. The variance of the intensity distribution around the landmarks diminishes compared to the average image intensity distribution.

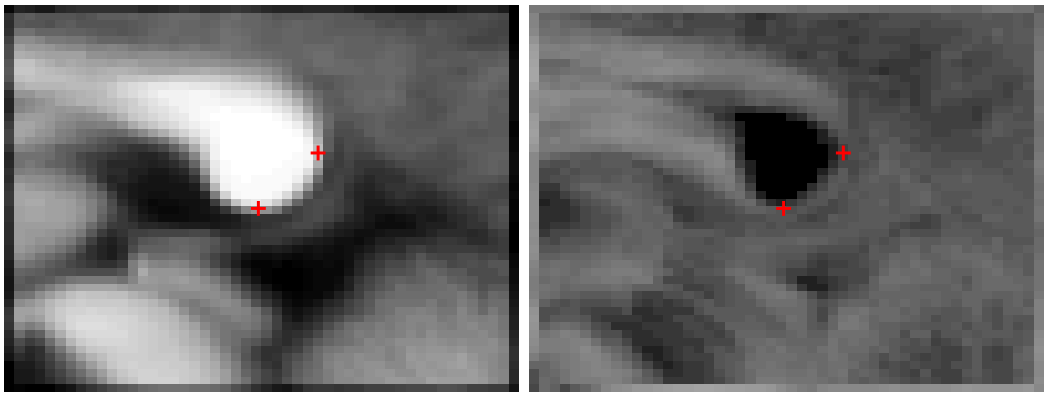
Landmark Detection

The prediction of the landmark location is performed on the testing set composed of 17 images. The likelihood is maximized by gradient ascent with respect to the landmark location according to equation (3.34). Recall though that in practice it is equivalent to maximize with respect to the landmark location and to the deformation parameter. The gradient ascent is initialized with the identity deformation, i.e. $y \rightarrow \bar{y}$. The size of the step is optimized by a line search algorithm at each iteration. Convergence is reached when the optimal step is 0. The Euclidean distance between the manual landmark and the estimated landmarks measures the performance of the algorithm. We compare the performance of the Deformable Intensity Model (DIM) with the detection by SSD. In both cases we use the same spline model for the deformation and find an optimum using a gradient method. Recall that while the intensity variance is constant throughout the SSD model, it varies at each pixel in the case of DIM. In addition, because the Jacobian is generally neglected in the estimation of the template, we compare the performance of DIM with normal estimation and the performance of DIM with approximated template estimation. We refer to this experiment as DIM-A. We make the same approximation for the estimation of the template in the case of SSD and denote that latter experiment with SSD-A. Initial refers to the distribution of the prediction error if we use the average location



(a) Average Intensity (b) Intensity Standard Deviation

Figure 3.2: Intensity Distribution **before** Registration: the red crosses represents the location of the landmarks: top-right SCC1, bottom-left SCC2. Images of the intensity distribution parameters learnt from the training set **before** registration

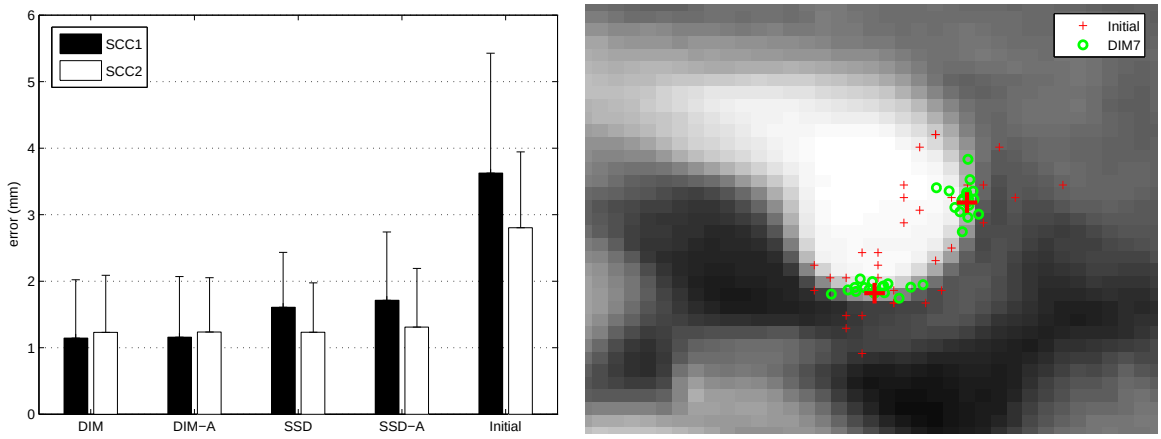


(a) Average Intensity: x_0 (b) Intensity Standard Deviation: τ_0

Figure 3.3: Estimated Intensity Template ($\sigma = 7$): the red crosses represents the location of the landmarks: top-right SCC1, bottom-left SCC2. Images of the intensity distribution parameters learnt from the training set **after** registration

of the landmarks in the training set as a prediction for all the new images. It assesses the variability of the landmark distribution before registration.

Figure 3.4(a) and Table 3.2 present the performance of the 5 predictors on the detection of SCC1 and SCC2. There exists a clear improvement between the initial error and the detection results obtained by each of the 4 detection methods. It is confirmed by the Wilcoxon test which rejects the equality hypothesis between each of the 4 predictors and Initial prediction error. The difference of performance between DIM and SSD is significant for SCC1 but not for SCC2. Recall though that SCC1 was located in the 3D volume while SCC2 is identified in the extracted 2D slice. It explains the lower variance of the landmark to begin with but also could explain why the predictors have the same performance on this landmark, which seem easier to detect. The statistical tests on the performance of DIM and DIM-A, and on the performance of SSD and SSD-A does not show a significant difference



(a) Prediction Error (b) Spatial Repartition of the Errors

Figure 3.4: **Left:** Bar plot representing the average error between the estimated and real position of SCC1 and SCC2. The error bar corresponds to the standard deviation of the error. **Right:** Spatial repartition of the error around the real location of the landmarks. The large red crosses represented the true location of the landmarks. The small black crosses represents the initial error for each image and the green circle the residual error after detection with DIM

	Prediction Error (mm)		Wilcoxon Signed Rank Test p-value				
	SCC1	SCC2	DIM	DIM-A	SSD	SSD-A	Initial
DIM	1.14 (0.88)	1.23 (0.86)	∅	0.8904	**0.0850**	**0.0850**	**0.0002**
DIM-A	1.16 (0.91)	1.24 (0.82)	0.9177	∅	**0.0582**	**0.0679**	**0.0001**
SSD	1.61 (0.83)	1.23 (0.74)	1.0000	1.0000	∅	0.8904	**0.0014**
SSD-A	1.71 (1.02)	1.31 (0.88)	0.8363	0.8633	0.9177	∅	**0.0034**
Initial	3.62 (1.80)	2.80 (1.14)	**0.0002**	**0.0002**	**0.0002**	**0.0004**	∅

Table 3.2: Statistical Comparison of Detection Performance. The left side of the table contains the mean and standard deviation of the prediction error (in mm) of SCC1 and SCC2 for each of the predictors, on a common testing set composed of 17 images. The righthand side of the table contains the p-value of the Wilcoxon Signed Rank Test for each couple of predictor. The p-values above the first diagonal of the table represent the test result for SCC1 and below the diagonal the p-value associated to the prediction error of SCC2. The values represented in bold correspond to the test that validates the existence of a difference for a tolerance level of $\alpha = 10$.

between computing the template parameter as a weighted sample or a simple sample.

Figure 3.4(b) represents the spatial repartition of the detection error of DIM around the real location of the landmarks. The error concentrates clearly around the true location showing a reduced variance after detection. It appears also that the error is oriented along the local edge of the image. Indeed both SCC1 and SCC2 are located on the edge of white matter and darker tissue in the image. The predicted position of the landmark is more precise in the direction orthogonal to the edge than along the edge.

3.6.3 Choice of the Kernel

In this section we investigate how the choice of the kernel influences the performance of the algorithm. The kernel is still Gaussian, but we vary its standard deviation: $\sigma = 3$, 5, 7 ,10 or 15 pixels. With a large variance, the number of pixels subject to a displacement is larger, therefore more pixels participate to the variation of the likelihood. It can be interpreted as increasing the size of the discriminative intensity pattern. We perform a set of experiments on both the Corpus Callosum and Hippocampus data sets.

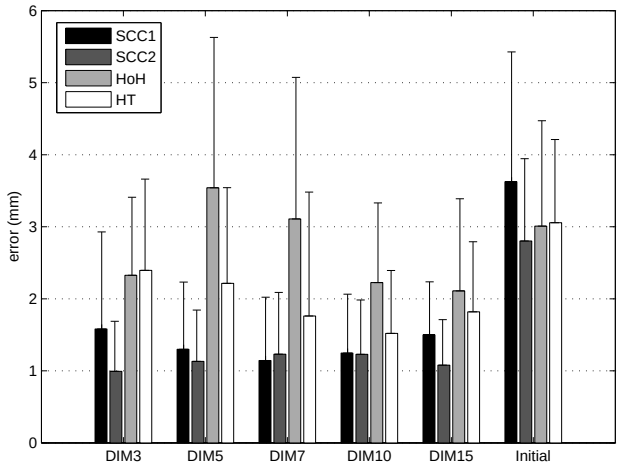


Figure 3.5: Performance of the detection algorithm using DIM for different choices of kernel standard deviation: 3, 5, 7, 10 and 15. The landmarks are detected by pair: SCC1 and SCC2, HoH and HT. Initial corresponds to the prediction error if we use only the average location of the landmarks from the training set. The error bar represents the standard deviation of the prediction error in the training set.

Figure 3.5 represents the performance of DIM with different values of the kernel variance. For most of the landmarks the best choice is $\sigma = 10$. While SCC2 is relatively stable across the different values of σ, the detection performance of HoH varies significantly depending on the kernel variance. This is due to the size of the intensity pattern learned around the landmark. By representing the prediction errors on the template, it is possible to visualize the type of intensity pattern that is around the estimated position and if there exist some direction of larger prediction error. Figure 3.6 illustrates the repartition of the prediction error in the case of the detection of HoH with DIM5. The prediction error is aligned with the lower edge separating the hippocampus from a white structure. This pattern disappears when the size of the kernel increases as shown by the error repartition of DIM10. Since the head of the hippocampus lies in a large gray region, for smaller kernel variance values, the intensity pattern is small and therefore does not capture enough information to be discriminative enough. In consequence in the absence of prior information, the detection algorithm is not specific enough and the prediction is translated along the lower contour of the hippocampus.

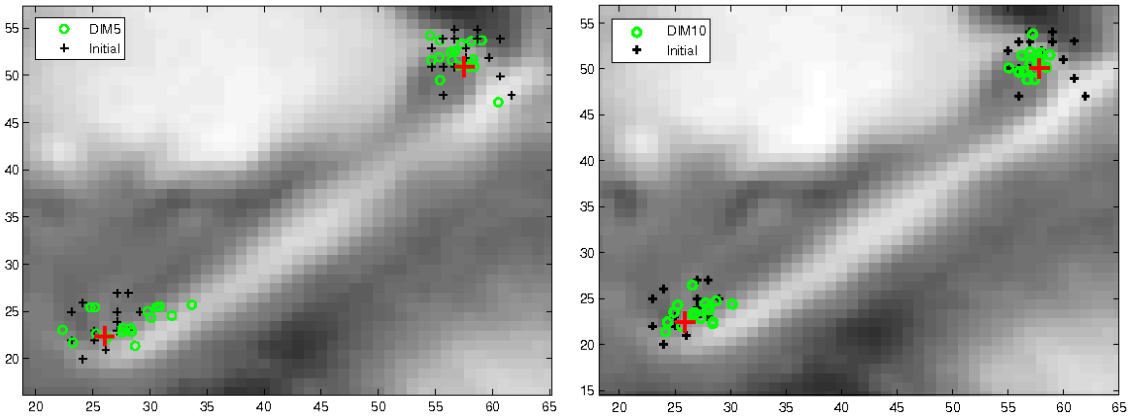


Figure 3.6: Distribution of the detection error for HoH and HT, represented on the template. The large red crosses represents the location of the landmarks, the black crosses the initial error and in green circles the detection error.

3.7 Chapter Conclusion

The Deformable Intensity Model is the simplest intensity matching model. It behaves similarly to SSD. However it illustrates well how a statistical model of the image can be used to derive both learning and testing algorithms. We will use the same principles in the following chapter, applied to other deformable image models.

Although it is very simple to use, the DIM faces an important limitation, which is that the intensity distribution in the image and in the template need to coincide. Indeed if we use the DIM on images whose intensity has not been normalized before, the prediction error increases from 1.15 mm for both SCC1 and SCC2 to respectively 1.95 and 1.77mm, with 1.74 and 1.12 mm of standard deviation. Beyond the degradation of the performance, it means that DIM can only be used from images coming from a same modality, which limits the range of applications.

Therefore in the following chapters we will propose some other deformable models which are robust to the change of intensity distribution. In Chapter 4, we present a model based on the distribution of edges in the image. In Chapter 5, we use a different strategy and propose a model in which the intensity distribution is image-specific.

CHAPTER 4

DEFORMABLE EDGE MODEL

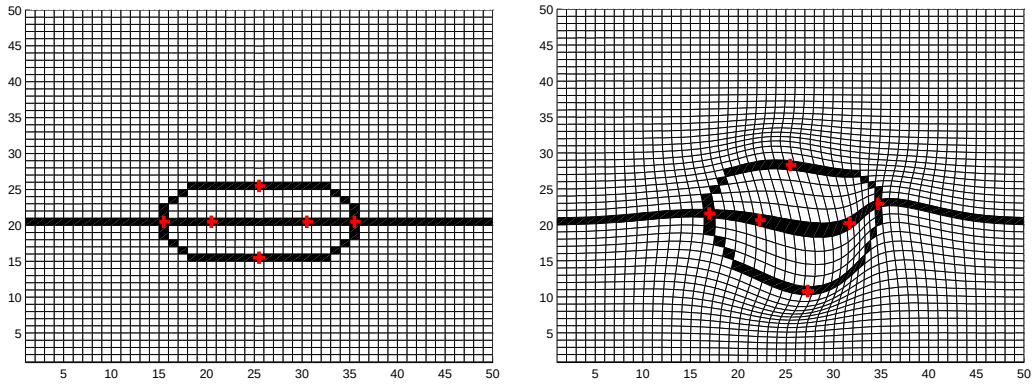
While the Deformable Intensity Model (DIM) is efficient for the detection of landmarks in image with normalized intensity distribution, as soon as the intensity from a new image differs from the learned template distribution, it fails at detecting accurately the landmark locations. A solution consists of building a model of an intensity-invariant image feature such as edges. We propose to use a simple edge detector based on local intensity comparison which therefore adapts to the local image intensity automatically. The probabilistic deformable model encodes the edges distribution in the image. In consequence it is not possible to generate full intensity images with that model but only contour images. Using the same principles as in the preceding chapter we derive an algorithm for landmark detection and test it on the detection of landmarks on synthetic images first and then on our database of medical images.

4.1 The Deformable Edge Model

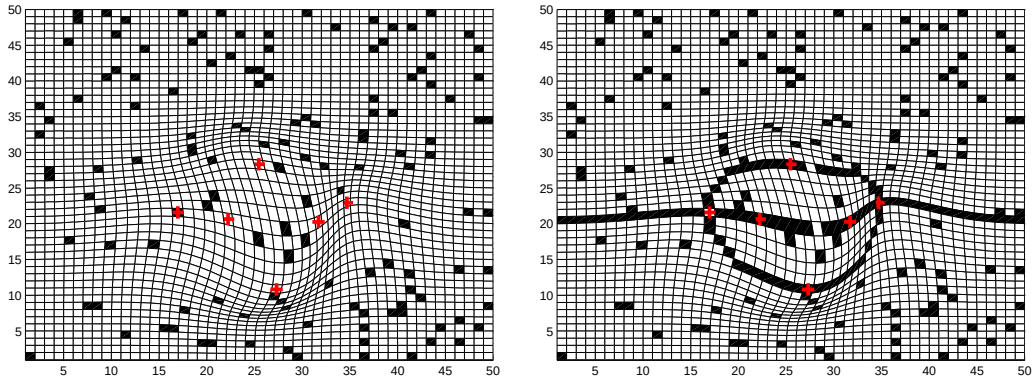
The spatial arrangement of the edges in an image is a cue that has been commonly used in image analysis specially for template-based image recognition. The binary edge image is generally obtained by filtering the original intensity image with an edge detector. We denote $x(s)$ the output of the edge detector, a binary random variable, which takes value 1 if an edge is detected at pixel s and 0 otherwise. We propose to model the repartition of the edges in an image using a statistical model based on a probabilistic deformable edge template. Similarly to the template of the DIM, the probabilistic deformable edge template encodes the probability at each location $t \in \Lambda_T$, a finite grid of $\mathbb{R}^d$, of observing an edge. We model the probability of observing an edge at a pixel by a Bernoulli distribution whose parameter depends on the location. The template is a function from Λ_T to $[0,1]$ and assigns to each location t the probability $\pi(1,t)$, which means that at t the edge distribution follows the Bernoulli distribution $\mathcal{B}(\pi(1,t))$. (We will sometimes refer to $\pi(0,t) = 1 - \pi(1,t)$). The location of the landmarks in the template is known and denoted by $\bar{y}$. Due to noise, the edge detector sometimes detects an edge in the background (false positive detection) or misses an edge (false negative detection). We model the noise effect as a binary channel which adds and removes edges. We introduce the binary noise-free image, denoted by z, which results from the deformation of the template. For all $s \in \Lambda$, $z(s)$ is a binary random variable which encodes the presence of an edge at s. Since the observed images x are always noisy, z is a hidden variable. The noise effect is modeled by:

$$\forall s \in \Lambda, \quad p(x(s) = 0 | z(s) = 1) = \rho \tag{4.1}$$

$$\forall s \in \Lambda, \quad p(x(s) = 1 | z(s) = 0) = \eta. \tag{4.2}$$



(a) Edge Template (b) Deformed Sample



(c) Noise (d) Final Image

Figure 4.1: Generating an edge image from a deformable edge template. 1) Sample from the probabilistic template 4.1(a) a random edge image (here identical to 4.1(a) because it is a deterministic template), 2) sample a set of landmarks location y from $p(y)$ and deform the edge image with the resulting deformation f_y to obtain the noise-free image 4.1(b), 3) Sample from the binary channel the switching pixels (here $\eta = \rho = 0.05$) 4.1(c), 4) Combine the noise-free image with the switching noise to obtain 4.1(b). The final image is resampled on a regular grid

The deformable model can be used to simulate images. Using the template, an edge image is sampled, i.e. at each $t \in \Lambda_T$ the presence of an edge is determined by drawing from the corresponding Bernoulli distribution $\mathcal{B}(\pi(1,t))$. Given the location of the landmarks in the final image, the edge image is deformed using the deformation f_y defined by the correspondence between the template landmarks $\bar{y}$ and the chosen image landmark locations y. The resulting deformed edge image is the noise-free image z. The final edge image is obtained by sampling the noise effect at each pixel $s \in \Lambda_T$ using the binary channel defined in (4.1) and (4.2). Figure 4.1 illustrates the simulation of images using the deformable edge model (DEM).

We assume conditional independence of the pixels $x(s)$ given $z(s)$, such that the model can be represented by the Bayesian graph of Figure 4.2.

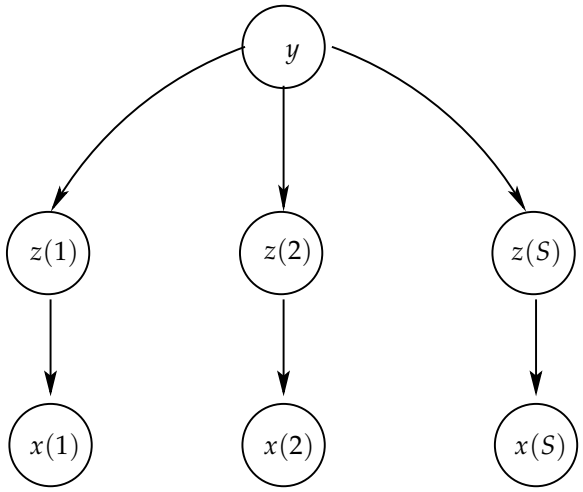


Figure 4.2: Bayesian network representing the Deformable Edge Model. y is the location of the landmarks and characterizes the geometry, $z(1), z(2), \cdots, z(S)$ are the noise-free edge variables at each pixel of the image and $x(1), x(2), \cdots, x(S)$ the presence or not of an edge in the observed image.

According to this model, the log-likelihood of an image x is:

$$\ell(x) = \ln p(y) + \sum_{s \in \Lambda} \ln p(x(s)|y). \tag{4.3}$$

Since $x(s)$ is a binary variable, we write the conditional distribution of $x(s)|y$:

$$p(x(s)|y) = x(s)p(x(s) = 1|y) + (1 - x(s))p(x(s) = 0|y). \tag{4.4}$$

The complementary events $\{x(s) = 1|y\}$ and $\{x(s) = 0|y\}$ are themselves decomposable as unions of events:

$$\{x(s) = 1|y\} = \{x(s) = 1 \cap z(s) = 1|y\} \cup \{x(s) = 1 \cap z(s) = 0|y\} \tag{4.5}$$

$$\{x(s) = 0|y\} = \{x(s) = 0 \cap z(s) = 1|y\} \cup \{x(s) = 0 \cap z(s) = 0|y\}. \tag{4.6}$$

We recall that using a deformable template model consists in assuming that the distribution parameter of $z(s) = 1|y$ is given by the template in $f_y^{-1}(s)$:

$$\forall s \in \Lambda, \quad p(z(s) = 1|y) = \pi(1, f_y^{-1}(s)), \quad p(z(s) = 0|y) = \pi(0, f_y^{-1}(s)). \tag{4.7}$$

Therefore, using the conditional independence assumption, (4.7) and the event decompositions (4.5) and (4.6),

$$\begin{aligned} p(x(s) = 1|y) &= p(x(s) = 1|z(s) = 1)p(z(s) = 1|y) + p(x(s) = 1|z(s) = 0)p(z(s) = 0|y) \\ &= (1 - \rho)\pi(1, f_y^{-1}(s)) + \eta\pi(0, f_y^{-1}(s)) \end{aligned} \tag{4.8}$$

$$\begin{aligned} p(x(s) = 0|y) &= p(x(s) = 0|z(s) = 0)p(z(s) = 0|y) + p(x(s) = 0|z(s) = 1)p(z(s) = 1|y) \\ &= (1 - \eta)\pi(0, f_y^{-1}(s)) + \rho\pi(1, f_y^{-1}(s)) \end{aligned} \tag{4.9}$$

It follows that the log-likelihood of an image is:

$$\ell(x) = \ln p(y) + \sum_{s\in\Lambda} \ln \Big\{ [(1-2\rho)x(s) + (1-\eta)]\, \pi(1, f_y^{-1}(s)) + [(2\eta-1)x(s) + \rho]\, \pi(0, f_y^{-1}(s)) \Big\} \tag{4.10}$$

A pixel contributes to increase the likelihood of an image if the observed value $x(s)$ has a large probability to be observed at the corresponding location in the template. The correspondence between the template and the image is given by the deformation f_y which is defined by the location of the landmarks in the image. Therefore the likelihood is a function of y. Consider the attachment term of the likelihood function, when $\eta = \rho = 0$:

$$\ell(x|y) = \sum_{s\in\Lambda} \ln \Big\{ x(s) \Big[\pi(1, f_y^{-1}(s)) - \pi(0, f_y^{-1}(s)) \Big] + \pi(0, f_y^{-1}(s)) \Big\}. \tag{4.11}$$

If $x(s) = 1$ and $\pi(1, f_y^{-1}(s)) \simeq 1$ or if $x(s) = 0$ and $\pi(1, f_y^{-1}(s)) \simeq 0$, the likelihood $p(x(s)|y)$ is close to 1, but if the observation does not correspond to the model, the likelihood tends to zero. The model counts the number of pixels whose observation correspond to the model. The noise parameters regularize the cost function by allowing mismatches between the image and the template up to a proportion equal to the amount of noise.

4.1.1 Edge Detection by Directional Intensity Comparison

In the large variety of possible edge detectors, we choose to use a simple edge detector based on local intensity comparisons [32]. The detector relies on the observation that the intensity variations are larger across than along an edge. The edge detector is therefore based on a non-parametric test, comparing the intensity variations across and along the tentative edge direction. The filter compares the intensity values of adjacent or neighboring pixels to s_0, the tested location, as depicted in Figure 4.3. If the intensity variation across the edge is the largest, the test is positive and an edge is detected. . For

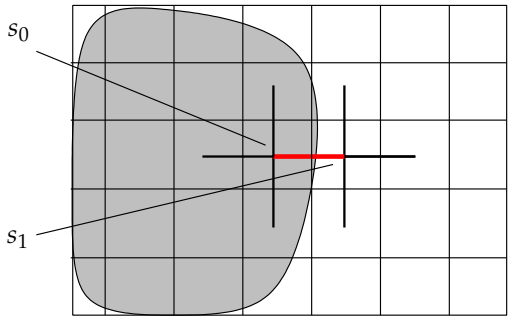


Figure 4.3: Edge Detection by Intensity Comparison. Given a grayscale image, the edge detector, represented here by the graph lying across the edge, compares the intensity along each of its segments. If the intensity difference along the central segment, represented in red, is the largest, an edge is detected in s_0.

example, in a region where the intensity is almost constant, an edge will be detected as soon as there exists a small variation. While in areas with larger intensity variations, the edge needs to be sharper and "steeper" to be detected. The detector's ability to adapt to the local range of intensity, together with its speed of computation, make it a simple

and robust detector, which does not rely on an arbitrary threshold. The definition of an edge depends on the scale at which the image is considered. Therefore a scale factor is introduced and corresponds to the distance between compared pixels. Large distances are more appropriate for coarser scale. Because of the presence of noise, the filter produces false positive and false negative responses. Varying the scale at which the detection is performed modifies the effect of the noise. At coarser scales the noise effect is reduced.

While in [32], the edge detector is applied at each pixel of the image in 4 directions with 2 orientations per direction, we choose to reduce the search at each pixel by simply aligning the tentative edge direction to the local intensity gradient. While in 2D images it does not make a crucial difference, in 3D images, the number of comparison would increase without this simplification.

Denoting $I(s)$ the intensity value at pixel s, the 2D edge detection algorithm works as follows:

Given 2 locations $s_0, s_1 \in \Lambda$ separated by a distance of d_{s_0,s_1}

1. Compute the normalized intensity gradient $\vec{w}_1 = \frac{\nabla I}{\|\nabla I\|}$, and $\vec{w}_2$ such that $(\vec{w}_1, \vec{w}_2)$ forms an orthonormal basis of $\mathbb{R}^2$,

2. Compare the intensity variation of the central clique $C_R \equiv |I(s_0) - I(s_1)|$, which is defined as the intensity difference between pixels s_0 and s_1, with the 6 lateral clique intensity differences (cf. Figure 4.3). Using the same notation:

$$C_{01} \equiv |I(s_0) - I(s_0 - d_{s_0,s_1} \cdot \vec{w}_1)|, \quad C_{11} \equiv |I(s_1) - I(s_1 + d_{s_0,s_1} \cdot \vec{w}_1)|$$
$$C_{02} \equiv |I(s_0) - I(s_0 - d_{s_0,s_1} \cdot \vec{w}_2)|, \quad C_{12} \equiv |I(s_1) - I(s_1 - d_{s_0,s_1} \cdot \vec{w}_2)|$$
$$C_{03} \equiv |I(s_0) - I(s_0 + d_{s_0,s_1} \cdot \vec{w}_2)|, \quad C_{13} \equiv |I(s_1) - I(s_1 + d_{s_0,s_1} \cdot \vec{w}_2)|$$

3. The presence of an edge at s_0 is determined as follows,

$$x(s_0) = \begin{cases} 1, & \text{if } \forall (i,j) \in \{0,1\} \times \{1,2,3\}, \quad C_R > C_{i,j}, \\ 0, & \text{if } \exists (i,j) \in \{0,1\} \times \{1,2,3\} : \quad C_{i,j} > C_R. \end{cases}$$

In 3D images, the intensity variations are considered along three orthogonal directions. Hence, denoting by $\vec{w}_1$ the unit vector following the local image gradient direction, we choose $\vec{w}_2$ and $\vec{w}_3$ such that $(\vec{w}_1, \vec{w}_2, \vec{w}_3)$ forms an orthonormal basis of $\mathbb{R}^3$. The central clique is compared to 10 lateral cliques determined by the unit vectors as depicted in Figure 4.4. If the intensity difference $|I(s_0) - I(s_1)|$ is larger than the intensity differences along all the other cliques, then $x(s_0) = 1$.

Figure 4.5 presents the output of the edge detector on 2D images of the Splenium of the Corpus Callosum, for different values of d_{s_0,s_1} and compares them with the edge images obtained with classical edge detectors. When the distance between pixels increases the sensibility to noise is reduced. A distance of 2 pixels seems to be a good choice for our images. On one hand it limits the noise effect and on the other hand it detects precisely the image contours. Notice that because the model is based on the distribution of the edges, only edge images can be generated from it.

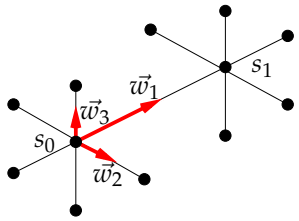
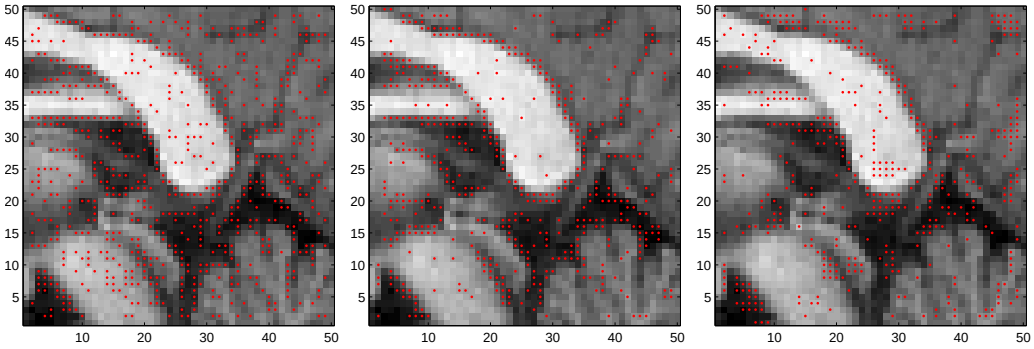
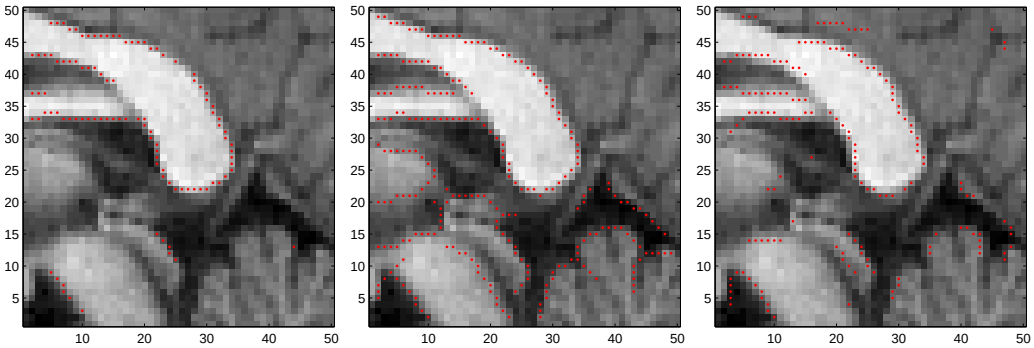


Figure 4.4: Edge Detector for 3D contour detection. Let s_0, s_1 be two voxels in the image grid, $(\vec{w}_3, \vec{w}_2, \vec{w}_1)$ an orthonormal basis of $\mathbb{R}^3$. If the intensity difference $|I(s_0) - I(s_1)|$ is larger than the intensity differences in the other cliques, then $x(s_0) = 1$.



(a) Intensity Comparison Edge Detector, From Left to Right: $d_{s_0,s_1} = 1$, 2 or 3 pixels



(b) Classical Edge Detectors, From Left to Right: Sobel's detector, Canny's detector, Zero-crossing detector

Figure 4.5: Comparison of some edge detectors

4.2 Model Selection

The model parameters are composed of the probabilistic template $(\pi(1,t), \forall t \in \Lambda_T)$ and the noise parameters η and ρ. The goal of the model selection is to use a training set of edge images, in which the location of the landmarks has been detected, to learn the parameters of the model $\theta = \{\pi(1,t), \forall t \in \Lambda_T; \eta, \rho\}$. The value of z, the noise-free image, is unobserved, also it is not possible to estimate directly all the parameters of the model. Indeed if the noise parameters were known, together with the location of the landmarks, the direct estimation as described for the deformable intensity model, would provide

an estimate of the probabilistic template. Similarly, would the template be known, the estimation of the noise parameters would be straightforward. But since both quantities are unknown we use the EM algorithm to estimate jointly the geometric parameters $\pi(1,t)$ for all $t \in \Lambda_T$ and the noise parameters η, ρ. The landmark distribution $p(y)$ can be estimated independently by any of the methods presented in 3.3.2.

4.2.1 Global Estimation by the EM Algorithm

As explained in 2.4.1, the EM algorithm consists in alternating between computing a lower bound of the likelihood and maximizing that bound. The expected log-likelihood of the joint distribution of the hidden variables $z(s)$ (or z if referring to the whole image) and the observed variable $x(s)$ is used as a lower bound of the log-likelihood.

The set of training images is composed of independent samples of the joint distribution $p(x,z|y)$. We denote θ the current value of the model parameters $\{\pi(1,t), \forall t; \eta, \rho\}$ (to be estimated) and θ' the estimate at the preceding iteration. Denoting by x_1^N the set of N images, y_1^N the set of landmark locations, z_1^N the set of noise-free edge images, the expected log-likelihood is:

$$\begin{aligned} Q(\theta,\theta') &= \mathbb{E}_z \left[\ln p_\theta(x_1^N, z_1^N | y_1^N) | x_1^N, y_1^N \right] \\ &= \sum_z \left[\sum_{i=1}^N \sum_{s\in\Lambda} \ln p_\theta(x^{(i)}(s), z^{(i)}(s) | y^{(i)}) \right] p_{\theta'}(z | x_1^N, y_1^N) \end{aligned} \tag{4.12}$$

Remark 4.1. *For the sake of keeping the notation simple, we use the following abbreviated notation:*

$$\sum_z \equiv \sum_{z^{(1)}(1)=0}^{1} \cdots \sum_{z^{(1)}(S)=0}^{1} \sum_{z^{(2)}(1)=0}^{N} \cdots \sum_{z^{(N)}(S)=0}^{1}$$

which is the sum over all the possible values of each noise-free image.

Changing the order of the sums

$$\begin{aligned} Q(\theta,\theta') &= \sum_{i=1}^N \sum_{s\in\Lambda} \sum_z \ln p_\theta(x^{(i)}(s), z^{(i)}(s) | y^{(i)}) p_{\theta'}(z | x_1^N, y_1^N), \\ &= \sum_{i=1}^N \sum_{s\in\Lambda} \sum_{z^{(i)}(s)} \ln p_\theta(x^{(i)}(s), z^{(i)}(s) | y^{(i)}) \sum_{z \setminus z^{(i)}(s)} p_{\theta'}(z | x_1^N, y_1^N), \\ &= \sum_{i=1}^N \sum_{s\in\Lambda} \sum_{z^{(i)}(s)} \ln p_\theta(x^{(i)}(s), z^{(i)}(s) | y^{(i)}) p_{\theta'}(z^{(i)}(s) | x_1^N, y_1^N). \end{aligned} \tag{4.13}$$

We decompose $Q(\theta,\theta')$ in different terms by developing the logarithm, using (4.7) and the conditional independence assumption:

$$Q(\theta,\theta') = \sum_{i=1}^N \sum_{s\in\Lambda} \sum_{z^{(i)}(s)=0}^{1} \ln \left[p_\theta(x^{(i)}(s) | z^{(i)}(s)) \pi(z^{(i)}(s), f_y^{-1}(s)) \right] p_{\theta'}(z^{(i)}(s) | x_1^N, y_1^N). \tag{4.14}$$

Using (4.8) and (4.9),

$$
\begin{aligned}
Q(\theta,\theta') &= \sum_{i=1}^{N}\sum_{s\in\Lambda} \ln\left[p_\theta(x^{(i)}(s)|z^{(i)}(s)=1)\pi(1,f_y^{-1}(s))\right] p_{\theta'}(z^{(i)}(s)=1|x_1^N,y_1^N) \\
&+ \sum_{i=1}^{N}\sum_{s\in\Lambda} \ln\left[p_\theta(x^{(i)}(s)|z^{(i)}(s)=0)\pi(0,f_y^{-1}(s))\right] p_{\theta'}(z^{(i)}(s)=0|x_1^N,y_1^N), \qquad (4.15)\\
&= \sum_{i=1}^{N}\sum_{s\in\Lambda} \ln\left[x(s)(1-\rho)+(1-x(s))\rho\right] p_{\theta'}(z^{(i)}(s)=1|x_1^N,y_1^N) \\
&+ \sum_{i=1}^{N}\sum_{s\in\Lambda} \ln\left[x(s)\eta+(1-x(s))(1-\eta)\right] p_{\theta'}(z^{(i)}(s)=0|x_1^N,y_1^N) \\
&+ \sum_{i=1}^{N}\sum_{s\in\Lambda}\sum_{j=0}^{1} \ln\left[\pi(j,f_{y^{(i)}}^{-1}(s))\right] p_{\theta'}(z^{(i)}(s)=j|x_1^N,y_1^N). \qquad (4.16)
\end{aligned}
$$

After initializing the parameters θ, the EM algorithm iterates the computation of $Q(\theta,\theta')$ and its maximization with respect to the model parameters.

4.2.2 Details of the E-step

The computation of $Q(\theta,\theta')$ relies on the expression of $p_{\theta'}(z^{(i)}(s)=1|x_1^N,y_1^N)$ for all i, and all s. Since $z^{(i)}(s)$ has only two possible values, it is enough to compute $p_{\theta'}(z^{(i)}(s)=1|x_1^N,y_1^N)$. It follows that $p_{\theta'}(z^{(i)}(s)=0|x_1^N,y_1^N)=1-p_{\theta'}(z^{(i)}(s)=1|x_1^N,y_1^N)$.

Proposition 1.

$$\forall i,\, s, \quad p_{\theta'}(z^{(i)}(s)|x_1^N,y_1^N) = p_{\theta'}(z^{(i)}(s)|x^{(i)}(s),y^{(i)})$$

Proof. In order to simplify the notations we choose $s=1$ and $i=1$,

$$
\begin{aligned}
p_{\theta'}(z^{(1)}(1),x_1^N|y_1^N) &= \sum_{z^{(1)}(2)}\cdots\sum_{z^{(N)}(S)} p_{\theta'}(z_1^N,x_1^N|y_1^N), \\
&= \sum_{z^{(1)}(2)}\cdots\sum_{z^{(N)}(S)} \prod_{i=1}^{N}\prod_{s=1}^{S} p_{\theta'}(z^{(i)}(s),x^{(i)}(s)|y^{(i)}), \\
&= p_{\theta'}(z^{(1)}(1),x^{(1)}(1)|y^{(1)}) \prod_{s=2}^{S} p_{\theta'}(z^{(1)}(s),x^{(1)}(s)|y^{(i)}) \prod_{i=2}^{N}\prod_{s=1}^{S} p_{\theta'}(x^{(i)}(s)|y^{(i)}).
\end{aligned}
$$

And on the other hand,

$$p_{\theta'}(x_1^N|y_1^N) = \prod_{i=1}^{N}\prod_{s=1}^{S} p_{\theta'}(x^{(i)}(s)|y^{(i)}).$$

Hence,

$$p_{\theta'}(z^{(1)}(1)|x_1^N,y_1^N) = \frac{p_{\theta'}(z^{(1)}(1),x^{(1)}(1)|y^{(1)})}{p_{\theta'}(x^{(1)}(1)|y^{(1)})} = p_{\theta'}(z^{(1)}(1)|x^{(1)}(1),y^{(1)}).$$

The same reasoning applies for all $s\in\Lambda$ and for all $i\in\{1,\cdots,N\}$. □

Using Bayes' rule and the parameters estimated at the preceding iteration, $\theta' = \{\eta', \rho'; \pi'\}$,

$$\begin{aligned} p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)}) &\propto x^{(i)}(s)(1-\rho')\pi'(1, f^{-1}_{y^{(i)}}(s)) + (1 - x^{(i)}(s))\rho'\pi'(1, f^{-1}_{y^{(i)}}(s)), \\ p_{\theta'}(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)}) &\propto x^{(i)}\eta'\pi'(0, f^{-1}_{y^{(i)}}(s)) + (1 - x^{(i)}(s))(1-\eta')\pi'(0, f^{-1}_{y^{(i)}}(s)). \end{aligned} \tag{4.17}$$

4.2.3 Details of the Noise Parameter Estimation

The maximization step consists of maximizing the function $Q(\theta, \theta')$ with respect to the new parameters $\theta = \{\eta, \rho; \pi(1,t), \forall t\}$. The joint maximization can be written as a set of independent optimizations.

$$\hat{\theta} = \arg\max_{\theta} Q(\theta, \theta') \Longleftrightarrow \begin{cases} \hat{\eta} = \arg\max_{\eta} Q(\theta, \theta') \\ \hat{\rho} = \arg\max_{\rho} Q(\theta, \theta') \\ \hat{\pi} = \arg\max_{\pi} Q(\theta, \theta') \end{cases} \tag{4.18}$$

We begin with the noise parameters. The maximum of ρ has a closed form solution. The derivative of the Q-function with respect to ρ is:

$$\frac{\partial}{\partial \rho} Q(\theta, \theta') = \sum_{i=1}^{N} \sum_{s \in \Lambda} \frac{1 - 2x^{(i)}(s)}{x^{(i)}(s)(1-\rho) + (1 - x^{(i)}(s))\rho} p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)}),$$

Since $x^{(i)}(s)$ can take only two values:

$$\frac{1 - 2x^{(i)}(s)}{x^{(i)}(s)(1-\rho) + (1 - x^{(i)}(s))\rho} = \begin{cases} \frac{1}{\rho - 1}, & \text{if } x^{(i)}(s) = 1 \\ \frac{1}{\rho}, & \text{if } x^{(i)}(s) = 0 \end{cases} \tag{4.19}$$

Hence,

$$\begin{aligned} &\sum_{i=1}^{N} \sum_{s \in \Lambda} \left[(1 - x^{(i)}(s)) \frac{1}{\rho} + x^{(i)}(s) \frac{1}{\rho - 1} \right] p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)}) = 0, \\ \Longleftrightarrow &\sum_{i=1}^{N} \sum_{s \in \Lambda} \left[-\rho + (1 - x^{(i)}(s)) \right] p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)}) = 0, \\ \Longleftrightarrow &\hat{\rho} = \frac{\sum_{i=1}^{N} \sum_{s \in \Lambda} \left(1 - x^{(i)}(s) \right) p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)})}{\sum_{i=1}^{N} \sum_{s \in \Lambda} p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)})} \end{aligned} \tag{4.20}$$

Recall from (4.1) that ρ is defined as the probability that an edge is present in the noise-free image but missing in the observed image. In (4.20), the denominator is the expected number of edges in the noise-free images. The numerator is the number of edge missing due to the noise in the training images. Hence the ratio is a natural estimator of the probability of missing an edge.

Similarly, we maximize the Q-function with respect to η,

$$\begin{aligned}
&\frac{\partial}{\partial \eta} Q(\theta, \theta') = 0 \\
&\iff \sum_{i=1}^{N} \sum_{s \in \Lambda} \frac{2x^{(i)}(s) - 1}{x^{(i)}(s)\eta + (1 - x^{(i)}(s))(1 - \eta)} p_{\theta'}(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)}) = 0,
\end{aligned} \tag{4.21}$$

Since $x(s)$ takes only two values,

$$\frac{2x^{(i)}(s) - 1}{x^{(i)}(s)\eta + (1 - x^{(i)}(s))(1 - \eta)} = \begin{cases} \frac{1}{\eta}, & \text{if } x^{(i)}(s) = 1 \\ \frac{1}{\eta - 1}, & \text{if } x^{(i)}(s) = 0 \end{cases} \tag{4.22}$$

$$\iff \sum_{i=1}^{N} \sum_{s \in \Lambda} \left[(1 - \eta) x^{(i)}(s) - \eta (1 - x^{(i)}(s)) \right] p_{\theta'}(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)}) = 0,$$

$$\iff \hat{\eta} = \frac{\sum_{i=1}^{N} \sum_{s \in \Lambda} x^{(i)}(s) p_{\theta'}(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)})}{\sum_{i=1}^{N} \sum_{s \in \Lambda} p_{\theta'}(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)})}. \tag{4.23}$$

According to (4.2), η is defined as the probability to observe noisy edges, i.e. to observe an edge because of the noise. In (4.23), the denominator is the expected number of edge-free pixels and the numerator is the number of edges due to noise in the training set. Therefore the ratio is an estimate of the probability of observing a noisy edge.

4.2.4 Details of the Edge Template Estimation

The template estimation consists in finding the parameter of the Bernoulli distribution at each $t \in \Lambda_T$ which encodes the probability to observe an edge at a pixel t, given the landmark location. Only the third term of the Q-function (4.15) depends on the template distribution. The expression (4.15) though is defined as a discrete sum over the pixels s of the training images. Since each image results from a specific deformation of the template, a fixed location s in the image support corresponds to different locations in the template, depending on the image-specific registering transformation $f_{y^{(i)}}$. We therefore perform the approximated integral change of variable, already presented in Chapter 3. After the change of variable, the optimization is defined on the template support and each pixel optimization can be performed independently. (In practice, because the image are actually observed on a finite grid, we need to interpolate it.) The optimization consists in finding for all $t \in \Lambda_T$ the parameter $\pi(j, t)$ that maximizes:

$$\sum_{i=1}^{N} \sum_{t \in \Lambda_T} \sum_{j=0}^{1} \ln \left[\pi(j, t) \right] p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) | J_{f_{y^{(i)}}}(t) |. \tag{4.24}$$

$\pi(j, t)$ does not depend on the image i, hence we modify the order of the sums:

$$\sum_{t \in \Lambda_T} \sum_{j=0}^{1} \ln \pi(j, t) \sum_{i=1}^{N} p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) | J_{f_{y^{(i)}}}(t) |. \tag{4.25}$$

Therefore,

$$\forall t, \forall j, \quad \hat{\pi}(j,t) \propto \sum_{i=1}^{N} p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) |J_{f_{y^{(i)}}}(t)|. \tag{4.26}$$

Notice that the Jacobian of the registering deformation weights the samples in the computation of the template estimate.

4.3 Landmark Detection by Local Edge Matching

In the detection algorithm, the purpose is to predict y in a new image, using the model selected during learning $(\hat{p}(y), \hat{\eta}, \hat{\rho}, \ln \hat{\pi}(1,t), \forall t)$. Similarly to what was done for the probabilistic deformable intensity model, the estimated position of the landmarks is obtained by a gradient algorithm maximizing the log-likelihood of a new image, with respect to y:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \left[\ell(x|y; \hat{\eta}, \hat{\rho}, \hat{\pi}) + \hat{p}(y) \right]. \tag{4.27}$$

The conditional log-likelihood function of a testing image is:

$$\ell(x|y; \hat{\eta}, \hat{\rho}, \hat{\pi}) \simeq \sum_{s \in \Lambda} \ln \left\{ (2x(s) - 1) \left[(1 - \hat{\eta} - \hat{\rho}) \hat{\pi}(1, f_y^{-1}(s)) - \left(\frac{1}{2} - \hat{\eta} \right) \right] + \frac{1}{2} \right\} \tag{4.28}$$

Neglecting the prior information, we compute the MLE:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} p(x|y). \tag{4.29}$$

In (4.28), only the template value depends on y. Similarly to the prediction using the Deformable Intensity Model, f_y^{-1} appears in the likelihood expression. To avoid computing its derivative, we perform the approximated integral change of variable with $t = f_y^{-1}(s)$:

$$\ell(x|y; \hat{\eta}, \hat{\rho}, \hat{\pi}) \simeq \sum_{t \in \Lambda_T} \ln \left\{ (2\, x\left(f_y(t)\right) - 1) \left[(1 - \hat{\eta} - \hat{\rho}) \hat{\pi}(1,t) - \left(\frac{1}{2} - \hat{\eta} \right) \right] + \frac{1}{2} \right\} |J_{f_y}(t)| \tag{4.30}$$

After the change of variable the intensity and the Jacobian become functions of y. Therefore the derivative is:

$$\begin{aligned}
&\frac{\partial \ell}{\partial y}(x|y; \hat{\eta}, \hat{\rho}, \hat{\pi}) \\
&= \sum_{t \in \Lambda_T} \frac{\partial x}{\partial y}(f_y(t)) \left[2\, x(f_y(t)) - 1 + \frac{1}{2\left[(1 - \hat{\eta} - \hat{\rho}) \hat{\pi}(1,t) - \frac{1}{2} - \hat{\eta} \right]} \right]^{-1} |J_{f_y}(t)| \\
&+ \sum_{t \in \Lambda_T} \ln \left\{ (2\, x\left(f_y(t)\right) - 1) \left[(1 - \hat{\eta} - \hat{\rho}) \hat{\pi}(1,t) - \left(\frac{1}{2} - \hat{\eta} \right) \right] + \frac{1}{2} \right\} \frac{\partial |J_{f_y}(t)|}{\partial y}.
\end{aligned} \tag{4.31}$$

Denoting $\frac{\partial}{\partial c_{l'}}$ the l'-th cartesian derivative and $f_y^{(l')}$ the l'-th component of the deformation, the derivative of the edge image is:

$$\frac{\partial x}{\partial y_{kl}}(f_y(t)) = \sum_{l'=1}^{d} \langle \frac{\partial x}{\partial c_{l'}}(f_y(t)), \frac{\partial f_y^{(l')}}{\partial y_{kl}}(t) \rangle.$$

The gradient ascent is coupled with a line search procedure to optimize the step size at each iteration.

Algorithm 4.2 summarizes the estimation of the Probabilistic Edge Deformable Model.

4.4 Deformable Edge Model with Image-specific Noise Parameters

It may happen that the noise level depends on the image. In this case instead of considering η and ρ as global parameters, they are modeled as image-specific parameters $\eta^{(i)}$ and $\rho^{(i)}$, taking their values between 0 and 1. The template, i.e. the conditional distribution of z given y remains common to all the images.

The training phase still requires an EM algorithm as the noise-free images remain unobserved and the model parameters unknown. The main difference occurs in the testing phase of the algorithm. It used to be a simple gradient method maximizing the likelihood with respect to y. In the case of the image-specific noise, the likelihood of a new image needs to be maximized with respect to y but now the image-specific noise parameters ρ, η are also unknown. In order to solve this joint estimation problem, we use an EM algorithm. We present briefly the changes in the landmark detection algorithm resulting from the model changes.

4.4.1 Model Estimation

The model estimation algorithm remains essentially the same, except that the image noise parameters are estimated independently for each image, $\theta = \{\forall i, \eta^{(i)}, \rho^{(i)}; \forall t, \pi(1,t)\}$.

Without entering in the details of the computation, and denoting θ' the estimate of the parameters at the preceding iteration, the Q-function becomes:

$$Q(\theta, \theta') = \sum_{i=1}^{N} \sum_{s \in \Lambda} \sum_{z^{(i)}(s)} \ln p_\theta(x^{(i)}(s), z^{(i)}(s) | y^{(i)}) p_{\theta'}(z^{(i)}(s) | x^{(i)}(s), y^{(i)}). \tag{4.32}$$

The E-step of the EM algorithm is performed by computing for each image i and at each pixel s the posterior distribution:

$$\begin{aligned} p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)}) &\propto \left[x^{(i)}(s)(1 - 2\rho^{(i)\prime}) + \rho^{(i)\prime} \right] \pi'(1, f_{y^{(i)}}^{-1}(s)), \\ p_{\theta'}(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)}) &\propto \left[x^{(i)}(s)(2\eta^{(i)\prime} - 1) + (1 - \eta^{(i)\prime}) \right] \pi'(0, f_{y^{(i)}}^{-1}(s)). \end{aligned} \tag{4.33}$$

In the M-step the noise estimation is now image specific:

$$\forall i \qquad \hat{\eta}^{(i)} = \frac{\sum_{s \in \Lambda} x^{(i)}(s) p_{\theta'}(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)}}{\sum_{s \in \Lambda} p_{\theta'}(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)})}, \tag{4.34}$$

$$\forall i \qquad \hat{\rho}^{(i)} = \frac{\sum_{s \in \Lambda} \left(1 - x^{(i)}(s)\right) p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)})}{\sum_{s \in \Lambda} p_{\theta'}(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)})}. \tag{4.35}$$

Algorithm 4.2 Probabilistic Deformable Edge Model

LEARNING

Let (x_1^N, y_1^N) be a training set and $\theta = \{\eta, \rho, \pi(1,t), \forall t \in \Lambda_T\}$ the model parameters

Initialize the model parameters η, ρ, and $\pi(1,t), \forall t \in \Lambda_T$,
Iterate until convergence

- **E-step:**

$$\forall i, \forall s, \qquad p_\theta(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)}) \propto \left[x^{(i)}(s)(1-2\rho) + \rho \right] \pi(1, f_{y^{(i)}}^{-1}(s)),$$
$$\forall i, \forall s, \qquad p_\theta(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)}) \propto \left[x^{(i)}(s)(2\eta - 1) + (1-\eta) \right] \pi(0, f_{y^{(i)}}^{-1}(s)).$$

- **M-step:**
  - Update the noise parameters

$$\eta \leftarrow \frac{\sum_{i=1}^N \sum_{s \in \Lambda} x^{(i)}(s) p_\theta(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)})}{\sum_{i=1}^N \sum_{s \in \Lambda} p_\theta(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)})},$$
$$\rho \leftarrow \frac{\sum_{i=1}^N \sum_{s \in \Lambda} \left(1 - x^{(i)}(s)\right) p_\theta(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)})}{\sum_{i=1}^N \sum_{s \in \Lambda} p_\theta(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)})},$$

  - Update the template,

$$\forall t, \quad \pi(1,t) \propto \sum_{i=1}^N |J_{f_{y^{(i)}}}(t)| p_\theta(z^{(i)}(f_{y^{(i)}}(t)) = 1 | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}).$$

TESTING

Let x be a testing image and (η, ρ, π) the parameters learnt during training,

Initialize with $y \leftarrow \bar{y}$
Iterate until convergence

- **Compute** the gradient,

$$\frac{\partial \ell}{\partial y}(x|y;\eta,\rho,\pi) \leftarrow \sum_{t \in \Lambda_T} \frac{\partial x}{\partial y}(f_y(t)) \left[2\,x(f_y(t)) - 1 + \frac{1}{2\left[(1-\eta-\rho)\pi(1,t) - \frac{1}{2} - \eta\right]} \right]^{-1} |J_{f_y}(t)|$$
$$+ \sum_{t \in \Lambda_T} \ln \left\{ \left[2\,x\left(f_y(t)\right) - 1 \right] \left[(1-\eta-\rho)\pi(1,t) - \left(\frac{1}{2} - \eta\right) \right] + \frac{1}{2} \right\} \frac{\partial |J_{f_y}(t)|}{\partial y}.$$

- **Optimize** the step size,

$$a \leftarrow \arg\max_{a \in \mathbb{R}^+} \ell \left(x | y + a \frac{\partial \ell(x|y;\eta,\rho,\pi)}{\partial y}; \eta, \rho, \pi \right),$$

- **Follow** the gradient direction,

$$y \leftarrow y + a \cdot \frac{\partial \ell(x|y;\eta,\rho,\pi)}{\partial y}$$

The update of the template is unchanged and given by,

$$\forall j,t, \qquad \hat{\pi}(j,t) \propto \sum_{i=1}^{N} |J_{f_{y^{(i)}}}(t)| p_{\theta'}(z(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}). \tag{4.36}$$

4.4.2 Landmark Detection

On a new image, it is necessary to estimate the noise parameters $\tilde{\theta} = \{\eta, \rho\}$ in addition of the landmark location y. The template does not need to be estimated since it was learnt during the training phase. The EM algorithm is used to perform the optimization. Denoting by $y', \tilde{\theta}'$ the estimates of the landmark locations and of the noise parameters at the preceding iteration, the expected log-likelihood to compute and maximize is:

$$\begin{aligned} & Q(\tilde{\theta}, \tilde{\theta}') \\ & = \sum_{s \in \Lambda} \left\{ \ln \left[x(s)(1-2\rho) + \rho \right] + \ln \pi(1, f_y^{-1}(s)) \right\} p_{\tilde{\theta}'}(z(s) = 1 | x(s), y') \\ & + \sum_{s \in \Lambda} \left\{ \ln \left[x(s)(2\eta - 1) + (1-\eta) \right] + \ln \pi(0, f_y^{-1}(s)) \right\} p_{\tilde{\theta}'}(z(s) = 0 | x(s), y'). \end{aligned} \tag{4.37}$$

The E-step is identical to the one of the training algorithm (4.33), except that we work with a single image at a time and with a fixed template. The M-step is composed of the estimation of the noise parameters which again is similar to the training algorithm. $\hat{\eta}$ is obtained with (4.34) and $\hat{\rho}$ with (4.35). There are no closed form solutions for the maximization of Q with respect to y, but since it is enough to improve the value of Q at each iteration to increase the log-likelihood, the maximization in y is replaced by a step in the direction of the Q function gradient.

$$\frac{\partial Q(\tilde{\theta}, \tilde{\theta}')}{\partial y} = \frac{\partial}{\partial y} \sum_{s \in \Lambda} \sum_{j=0}^{1} \left[\ln \hat{\pi}(j, f_y^{-1}(s)) \right] p_{\tilde{\theta}'}(z(s) = j | x(s), y') \tag{4.38}$$

We perform the usual approximated integral change of variable, $t = f_y^{-1}(s)$ to write the cost function on the template support:

$$\frac{\partial Q(\tilde{\theta}, \tilde{\theta}')}{\partial y} \simeq \sum_{t \in \Lambda_T} \sum_{j=0}^{1} \ln \hat{\pi}(j,t) \frac{\partial}{\partial y} \left\{ p_{\tilde{\theta}'}(z(f_y(t)) = j | x(f_y(t)), y') |J_{f_y}(t)| \right\}. \tag{4.39}$$

After the change of variable, the Jacobian and the posterior distribution depend on y.

Deriving the Posterior Distribution

Because of the change of variable, the posterior distribution depends on the position of y. Propagating the change of variable into the expression of the posterior distribution (4.33),

$$\begin{aligned} & p_{\theta'}(z(f_y(t)) = 1 | x(f_y(t)), y') \propto \left[x(f_y(t))(1-2\rho') + \rho' \right] \pi'(1, f_{y'}^{-1} \circ f_y(t)), \\ & p_{\theta'}(z(f_y(t)) = 0 | x(f_y(t)), y') \propto \left[x(f_y(t))(2\eta' - 1) + (1-\eta') \right] \pi'(0, f_{y'}^{-1} \circ f_y(t)). \end{aligned} \tag{4.40}$$

y' is the estimated landmark location at the preceding iteration, therefore the composition of the transformation and its inverse $f_{y'}^{-1} \circ f_y$ is not necessarily the identity. Hence one

needs to compute the derivative of $\pi'(1, f_{y'}^{-1} \circ f_y(t))$ with respect to y. The resulting calculation is quite complicated.

One numerical solution consists in approximating $p_{\tilde{\theta}'}(z(s) = j|x(s), y')$ as a function of s and j, pre-computed during the E-step, such that its derivative is:

$$\frac{\partial p_{\tilde{\theta}'}}{\partial y_{kl}}(z(f_y(t)) = j|x(f_y(t)), y') \simeq \frac{\partial p_{\tilde{\theta}'}(j, f_y(t))}{\partial y_{kl}} = \sum_{l'=1}^{d} \langle \frac{\partial p_{\tilde{\theta}'}}{\partial c_{l'}}(j, f_y(t)), \frac{\partial f_y^{(l')}}{\partial y_{kl}}(t) \rangle,$$

denoting $\frac{\partial}{\partial c_{l'}}$ the l'-th cartesian derivative and $f_y^{(l')}$ the lth component of the deformation f_y.

With that approximation the derivative of $Q(\tilde{\theta}, \tilde{\theta}')$ with respect to y becomes:

$$\frac{\partial Q(\tilde{\theta}, \tilde{\theta}')}{\partial y_{kl}}$$
$$\simeq \sum_{t \in \Lambda_T} \sum_{j=0}^{1} \ln \pi(j, t) \left[|J_{f_y}(t)| \sum_{l'=1}^{d} \langle \frac{\partial p_{\tilde{\theta}'}}{\partial c_{l'}}(j, f_y(t)), \frac{\partial f_y^{(l')}}{\partial y_{kl}}(t) \rangle + p_{\tilde{\theta}'}(j, f_y(t)) \frac{\partial |J_{f_y}(t)|}{\partial y_{kl}} \right] \quad (4.41)$$

Modification of the EM algorithm While the derivation of $p_{\tilde{\theta}'}$ is quite complicated, the expression of the derivative of the log-likelihood is tractable. Both the expression of the log-likelihood (4.30) and of the expected log-likelihood (4.37) requires the derivation of the Jacobian and of the image. In addition to these quantities the expected likelihood involves the computation of the derivative of $p_{\tilde{\theta}'}$ while the likelihood does not.

Using the generic notation for the EM algorithm, we recall that the Q-function is defined such that:

$$\ln p_\theta(x) - \ln p_{\theta'}(x) = Q(\theta, \theta') - Q(\theta', \theta') + DL(p_{\theta'}(z|x), p_\theta(x)), \quad (4.42)$$

with $DL(\cdot, \cdot)$ the Kullback-Leibler divergence, which is a positive quantity. The goal of the EM algorithm is to provide a tractable way to maximize the likelihood. We denote x the observations, z the hidden variable, θ_1, θ_2 the two model parameters and θ_1', θ_2' their estimates at the preceding iteration. We propose to modify the M-step of the EM algorithm by combining the maximization of the Q-function and the maximization of the likelihood function (cf Alg. 4.3).

Using the same notations as before, it can be proved that,

Theorem 4.1. *$\forall \theta_1' \in \Theta_1, \theta_2' \in \Theta_2$, by choosing $\hat{\theta}_1, \hat{\theta}_2$ as described in Alg. 4.3,*

$$\ln p_{\{\hat{\theta}_1, \hat{\theta}_2\}}(x) \geq \ln p_{\{\theta_1', \theta_2'\}}(x)$$

Proof. According to Eq.(4.42), choosing $\hat{\theta}_1$ that maximizes $Q(\theta_1, \theta_2; \theta_1', \theta_2')$ in θ_1 leads to $\ln p_{\{\hat{\theta}_1, \theta_2\}} \geq \ln p_{\{\theta_1', \theta_2'\}}$. Since in addition for any θ_2, $\hat{\theta}_2$ is such that $p\{\hat{\theta}_1, \hat{\theta}_2\}(x) \geq p\{\hat{\theta}_1, \theta_2\}(x)$ $\Rightarrow \ln p_{\{\hat{\theta}_1, \hat{\theta}_2\}}(x) \geq \ln p_{\{\theta_1', \theta_2'\}}(x)$. □

Therefore the Modified EM algorithm can be used in lieu of the EM algorithm and the likelihood increases at each iteration.

Algorithm 4.3 Modified EM algorithm

Starting from some initial values of the model parameters: $\theta = \{\theta_1, \theta_2\}$, iterate until convergence:

E-step: Posterior distribution
Given the preceding estimate of the parameters $\theta' = \{\theta'_1, \theta'_2\}$ find

$$p_{\{\theta'_1,\theta'_2\}}(z|x) \leftarrow \frac{p_{\{\theta'_1,\theta'_2\}}(x|z)p_{\{\theta'_1,\theta'_2\}}(z)}{\sum_z p_{\{\theta'_1,\theta'_2\}}(x|z)p_{\{\theta'_1,\theta'_2\}}(z)}$$

M-step: Maximization
Update the model parameters,

$$\hat{\theta}_1 = \arg\max_{\theta_1\in\Theta_1} Q(\theta_1, \theta_2; \theta'_1, \theta'_2), \quad \hat{\theta}_2 = \arg\max_{\theta_2\in\Theta_2} \ln p_{\{\hat{\theta}_1,\theta_2\}}(x).$$

Application to the Landmark Detection with the Deformable Edge Model Coming back to the Deformable Edge Model, we identify $\theta_1 \equiv \{\eta, \rho\}$ and $\theta_2 \equiv y$. The maximization step consists in finding $\tilde{\theta} = \{\eta, \rho\}$ and y such that the Q-function is maximal. While the maximization with respect to η and ρ is unchanged, we replace the maximization of Q with respect to y, by the maximization of the log-likelihood function:

$$(\hat{\eta}, \hat{\rho}) = \arg\max_{\eta,\rho} Q(\tilde{\theta}, y; \tilde{\theta}', y') \quad \text{and} \quad \hat{y} = \arg\max_{y} \ln p_{\{\hat{\eta},\hat{\rho}\}}(x|y). \tag{4.43}$$

The maximization of the likelihood with respect to y does not admit a closed form solution. Therefore the maximization is carried out by gradient ascent:

$$\begin{aligned}
&\frac{\partial \ell}{\partial y}(x|y;\eta,\rho,\pi) \\
&= \sum_{t\in\Lambda_T} \frac{\partial x}{\partial y}(f_y(t)) \left[2\,x(f_y(t)) - 1 + \frac{1}{2\left[(1-\hat{\eta}-\hat{\rho})\pi(1,t) - \frac{1}{2} - \hat{\eta}\right]}\right]^{-1} |J_{f_y}(t)| \\
&+ \sum_{t\in\Lambda_T} \ln\left\{\left[2\,x\left(f_y(t)\right) - 1\right]\left[(1-\hat{\eta}-\hat{\rho})\pi(1,t) - \left(\frac{1}{2} - \hat{\eta}\right)\right] + \frac{1}{2}\right\} \frac{\partial |J_{f_y}(t)|}{\partial y}.
\end{aligned} \tag{4.44}$$

This is the same expression as for the prediction of landmarks using the Deformable Edge Model with global noise parameters (4.31), except that here it is computed at each iteration of the EM algorithm with the current estimate of the noise parameters.

Algorithm 4.4 summarizes the algorithm proposed in the case of the Deformable Edge Model with image-specific noise levels.

4.5 Detection Results

4.5.1 Synthetic Experiment

In order to generate a set of synthetic images, it is enough to choose a template, some noise parameters and the landmark location. In our experiment, we choose a deterministic

Algorithm 4.4 Deformable Edge Model:

LEARNING

Let (x_1^N, y_1^N) be a training set, $\theta = \{\forall i, \eta^{(i)}, \rho^{(i)}\}$ the set of noise parameters of image i and, $\{\pi(1,t), \forall t \in \Lambda_T\}$ the template

Initialize $\forall i$, $\eta^{(i)}, \rho^{(i)}$, and $\pi(1,t), \forall t \in \Lambda_T$,
Iterate until convergence

- **E-step:**

$$\forall i, \forall s, \quad p_\theta(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)}) \propto \left[x^{(i)}(s)(1-2\rho^{(i)}) + \rho^{(i)}\right] \pi(1, f_{y^{(i)}}^{-1}(s)),$$
$$\forall i, \forall s, \quad p_\theta(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)}) \propto \left[x^{(i)}(s)(2\eta^{(i)}-1) + (1-\eta^{(i)})\right] \pi(0, f_{y^{(i)}}^{-1}(s)).$$

- **M-step:**
  - Update the noise parameters for each image i

$$\eta^{(i)} \leftarrow \frac{\sum_{s\in\Lambda} x^{(i)}(s) p_\theta(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)})}{\sum_{s\in\Lambda} p_\theta(z^{(i)}(s) = 0 | x^{(i)}(s), y^{(i)})},$$
$$\rho^{(i)} \leftarrow \frac{\sum_{s\in\Lambda} \left(1 - x^{(i)}(s)\right) p_\theta(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)})}{\sum_{s\in\Lambda} p_\theta(z^{(i)}(s) = 1 | x^{(i)}(s), y^{(i)})},$$

  - Update the template,

$$\forall j, t, \quad \pi(j,t) \propto \sum_{i=1}^{N} |J_{f_{y^{(i)}}}(t)| p_\theta(z^{(i)}(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}).$$

TESTING

Let x be a testing image, π the parameters learnt during training, $\tilde{\theta} = \{\eta, \rho\}$ the unknown noise parameters, y the unknown landmark location

Initialize η, ρ and $y \leftarrow \bar{y}$
Iterate until convergence

- **E-step:**

$$\forall s, \quad p_{\tilde{\theta}}(z(s) = 1 | x(s), y) \propto [x(s)(1-2\rho) + \rho]\, \pi(1, f_y^{-1}(s)),$$
$$\forall s, \quad p_{\tilde{\theta}}(z(s) = 0 | x(s), y) \propto [x(s)(2\eta-1) + (1-\eta)]\, \pi(0, f_y^{-1}(s)).$$

- **M-step:**
  - **Update** the noise parameters

$$\eta \leftarrow \frac{\sum_{s\in\Lambda} x(s) p_{\tilde{\theta}}(z(s) = 0 | x(s), y)}{\sum_{s\in\Lambda} p_{\tilde{\theta}}(z(s) = 0 | x(s), y)} \quad \text{and} \quad \rho \leftarrow \frac{\sum_{s\in\Lambda} (1 - x(s))\, p_{\tilde{\theta}}(z(s) = 1 | x(s), y)}{\sum_{s\in\Lambda} p_{\tilde{\theta}}(z(s) = 1 | x^{(i)}(s), y)},$$

  - **Compute** the gradient direction $\frac{\partial \ell}{\partial y}(x | y; \eta, \rho, \pi)$ from (4.44).
  - **Update** the location of the landmarks,

$$y \leftarrow y + a \cdot \frac{\partial \ell(x|y;\eta,\rho,\pi)}{\partial y}, \quad \text{with} \quad a \leftarrow \arg\max_{a\in\mathbb{R}^+} \ell\left(x | y + a\frac{\partial \ell(x|y;\eta,\rho,\pi)}{\partial y}; \eta, \rho, \pi\right),$$

Noise	Model Estimation			Landmark Detection Error (mm)			
	$\hat{\eta}$	$\hat{\rho}$	Entropy	LEFT	RIGHT	TOP	BOTTOM
0.01	0.00	0.02	19	0.30 (0.29)	0.29 (0.23)	0.95 (0.56)	0.91 (0.50)
0.05	0.00	0.04	57	0.73 (0.48)	0.67 (0.41)	1.13 (0.56)	1.02 (0.63)
0.10	0.03	0.07	113	0.62 (0.44)	0.66 (0.45)	1.06 (0.61)	1.01 (0.61)
0.15	0.05	0.08	184	0.69 (0.40)	0.77 (0.54)	1.21 (0.67)	1.03 (0.58)
0.25	0.07	0.13	390	1.03 (0.61)	0.93 (0.62)	1.25 (0.63)	1.23 (0.69)

Table 4.1: Synthetic Experiment Results

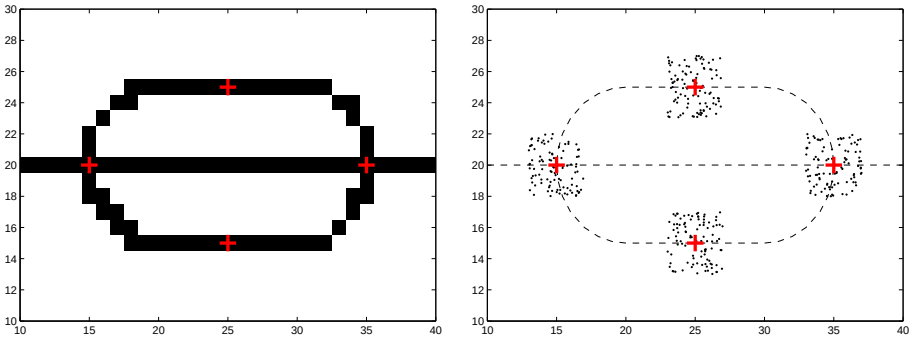
template (50×50 pixels), representing an ellipse intersected by a horizontal line. The noise parameters vary between 0.01 and 0.25 depending on the experiment. We define 4 landmarks in the template, on the ellipse contour. Each coordinate of the landmark displacement is sampled from the Uniform distribution on $[-2 : 2]$ pixels. We use the Gaussian spline as deformation model with a standard deviation of 5 pixels.The deformation model is based on a Gaussian kernel with $\sigma = 5$.

In this experiment, the noise parameters are common to all the images. They both vary between 0.01 and 0.25. For simplicity we choose them equal in this experiment but do not enforce it in the estimation of the model parameters. 50 random images are generated for the training of the model. Figure 4.7 presents few examples of random images sampled from the model, for different level of noise. In the training by EM, we initialize the noise parameters by $\eta = \rho = 0.10$ and the template using the average of the training images. The average consists simply in computing the proportion of edges observed at each pixel in the training set of images, without registration.

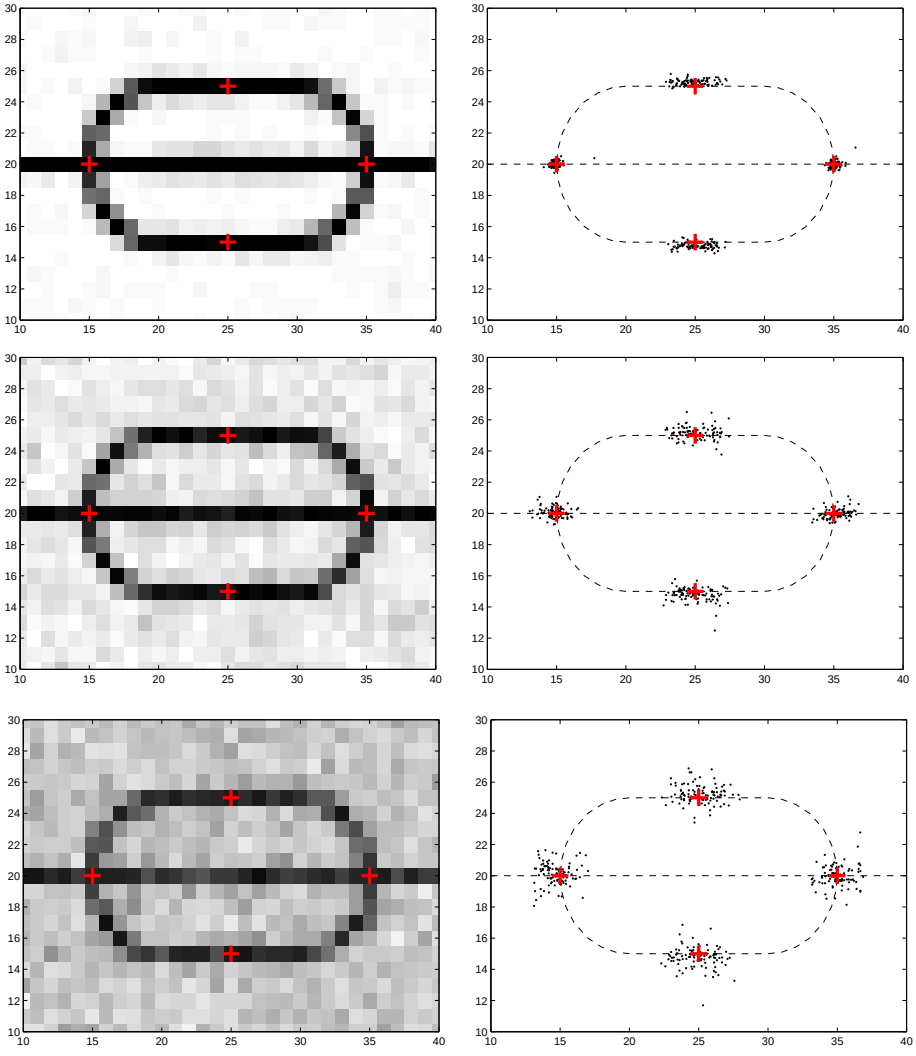
The detection algorithm is tested on 100 independent random images, generated with the same model as the training image. The learnt model is used to predict the location of the landmarks in the testing images. We use a gradient method to optimize the likelihood function, starting from the location of the landmark in the template $\bar{y}$. Figure 4.7 presents the visual results on 3 images. The red crosses correspond to the true landmark locations, the blue crosses represent the initialization of the gradient ascent and the green crosses correspond to the location predicted by DEM.

Table 4.5.1 presents the results of the synthetic experiment with various level of noise. The leftmost side of the table contains the estimated noise parameters. It is noticeable that both noise parameters are underestimated and that the entropy of the template increases when the level of noise increases. It shows that the model does not identify properly the different sources of noise, the one resulting from the geometric variation and the one coming from the additional noise. If the number of images increases, for a fixed level of noise, the estimation of the parameters is improved (this is not reported here but was observed during the experiments).

The repartition of the prediction error and the estimated template are represented in Figure 4.6. When the amount of noise in the training images increases, as we observed in Table 4.5.1, the noise parameter tend to be underestimated and therefore the template contains some residual edges which correspond to the noisy edges not captured by the noise parameters. Looking at the repartition of the prediction error in the testing set, one can make several comments. First, for all level of noise represented (1, 10 and 25



(a) Left: Real Template, Right: Initial Error



(b) Left Column: Estimated Template, Right Column: Prediction Error

Figure 4.6: Estimated Template and Prediction Error. Left Column: Image Template, black represents a probability close to 1 to observe an edge while white represents a probability close to 0. The red crosses represent the location of the landmarks in the template. Right Column: Localization Error, each black dot represents the localization error for one image, represented with respect to the average location of the landmarks (red crosses).

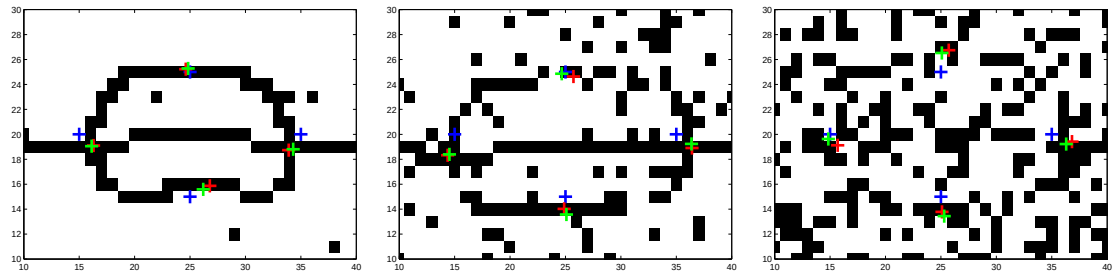


Figure 4.7: From Left to Right: Examples of simulated images with respective levels of noise 0.01 0.10 and 0.25. The Red crosses represent the ground truth of the landmarks location, the blue crosses the starting point of the optimization and the green crosses the estimated position of the landmarks.

%), the localization error is significantly reduced by the prediction algorithm. Second, the prediction error increases with the amount of noise. This is due to local minima in which the gradient ascent gets trapped. Finally, the repartition of the error around each landmark shows well that the prediction error of TOP and BOTTOM is oriented along the ellipse edge. The error repartition of the other landmarks RIGHT and LEFT is apparently spherical. These observations are valid for all levels of noise represented in Figure 4.6. Since the algorithm uses only edges as cues to locate the landmarks, the detection precision increases if there exists a rich distinctive edge pattern around the landmark. Since LEFT and RIGHT are located at the intersection of the ellipse and the horizontal line, there exists a lot of local information to detect them. On the contrary for the TOP and BOTTOM landmarks, the local information is reduced to a locally horizontal edge, which does not provide any information about the horizontal localization of the landmarks.

This shows that the gradient descent algorithm gets trapped in some local minima of the likelihood function, specially when the noise level increases.

4.5.2 Detection of a Landmark in Real Images

We use the Deformable Edge Model (DEM) to detect the SCC1 and SCC2 in the data set 2D-SCC. The images were first filtered with the intensity-comparison edge detector presented in this chapter. We use a scale factor of 2 pixels. The filtered images are represented in Figure 4.5. We use a Gaussian spline deformation model with standard deviation $\sigma = 7$. We detect simultaneously SCC1 and SCC2. We use 30 images for training and 17 images for the test.

Table 4.5.2 summarizes the results obtained on this data set with DEM. Initialization refers to the initialization of the algorithm and the localization error of the landmarks before detection. *Full training* refers to the DEM algorithm with global noise parameters. *Different Noise Training* corresponds to the DEM algorithm with image specific noise parameters. The noise estimates in this case represent the average estimated value in the 17 testing images. Finally since it is difficult to judge whether the amount of noise estimated is correct, we performed a set of experiments summarized in *Partial Training* in which the value of the noise parameters is fixed manually before training. This experiment was repeated with different values of the noise parameters.

	Model Estimation			Prediction Error (pixel)	
	$\hat{\eta}$	$\hat{\rho}$	entropy	SCC 1	SCC 2
Initialization	0.10	0.10	205	3.62 (1.80)	2.82 (1.20)
Full Training	0.02	0.07	345	2.79 (2.37)	1.72 (1.68)
Different Noise Training	0.03	0.13	353	3.54 (2.95)	1.99 (1.85)
Partial Training	0.05	0.05	301	2.94 (2.41)	2.11 (1.69)
	0.10	0.10	248	2.31 (2.32)	1.91 (1.57)
	0.15	0.15	196	2.75 (2.05)	1.80 (1.27)
	0.25	0.25	111	2.92 (2.22)	1.90 (1.42)

Table 4.2: Performance on 2D-SCC

Figure 4.8 depicts the template learnt in some of the experiments. The top left image represents the initialization of the learning algorithm, which is the average edge image obtained from the 30 training images. The top right image corresponds to the template learn in the *Full Training* experiment. During learning the images are locally registered using the landmark correspondence. Locally all the edges are superimposed and locally the probability to observe an edge is close to 1. The probability of missing an edge is estimated to 0.07 while the probability to observe an edge due to noise is 0.02. The noise parameters seem to be underestimated, because we observed a significant amount of noisy edges and the edge detector does miss some edges. In terms of performance, the average error is reduced for both SCC1 and SCC2, the standard deviation though is very large. This is explained by the sparseness of the edges in the image and the usage of the gradient method. In the *Partial Training* experiments the amount of noise is not learnt but manually chosen. The results are comparable to the ones obtained by the *Full Training* method. The average prediction is improved but there exists a large variance in the results. When we fix the amount of noise in the model higher than the one learnt in the other experiment, we notice that the entropy of the template decreases which means that some of the edges are now modeled by the noise parameters.

4.6 Chapter Conclusion

The main purpose of the model presented in this chapter was to build a statistical model on the repartition of edges in the image. By doing so it produces an edge matching algorithm that can be used for the detection of landmarks. If the edge detector performance are not altered by the change of intensity distribution in the image, the resulting model is invariant to the change of variables. To evaluate the model and the derived landmark detection algorithm, the model was tested on synthetic images first and then on the detection of SCC1 and SCC2 in the 2D-SCC data set. Although the algorithm detection precision in the synthetic data set is at most of the order of 1 pixel with up to 25 percents of noise, the performance on real images does not match this performances. We believe the lack of performance on real images comes from the sparseness of the edges on the images which is not appropriate for gradient methods.

In this chapter, we have reused the modeling principles presented in Chapter 3 and applied them on the DEM. Because the edge images are noisy, we introduced a hidden variable which represents the noise-free edge model. Due to this hidden variable, the

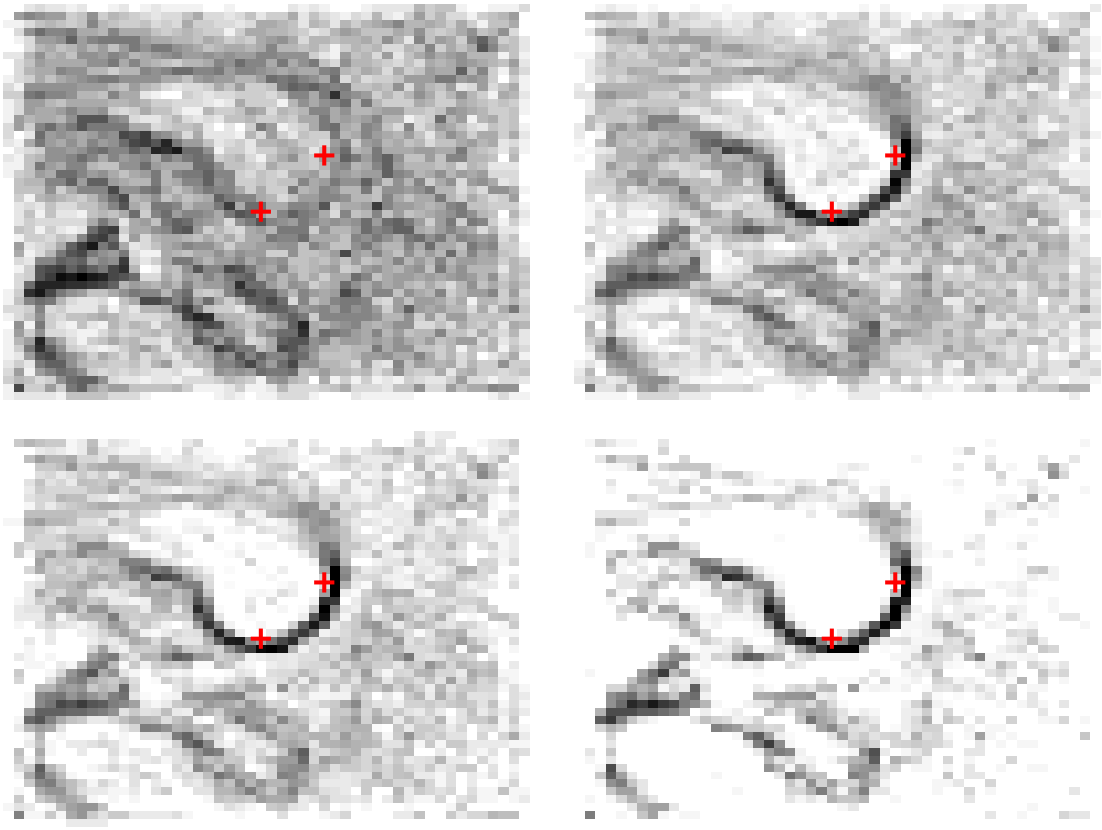


Figure 4.8: Estimated Template based on 30 images of the Splenium of the Corpus Callosum, with SCC1 and SCC2, represented by the red crosses. Top Left: Initialization edge distribution obtained by averaging the training images without registration. (Entropy=205 bits) Top Right: Template obtained by joint estimation ($\hat{\eta} = 0.02$ and $\hat{\rho} = 0.07$, Entropy=345 bits). Bottom Left: Template obtained with fixed noise parameters (0.10), (Entropy=248 bits), Bottom Right: Template obtained with fixed noise parameters (0.25), (Entropy=111 bits).

straightforward estimation methods presented in the preceding chapter do not hold in this case. Instead the EM algorithm or an approximated EM algorithm was used to learn the model but also for the detection of landmarks.

CHAPTER 5

TISSUE-BASED DEFORMABLE INTENSITY MODEL

In the preceding chapters, we proposed two deformable models, the Deformable Intensity Model (DIM) and the Deformable Edge Model (DEM). We have seen in Chapter 3 that the performance of DIM are affected by variability in the image intensity. Not only a lack of robustness, the model lacks of generality, as for example it is not possible to use combined information from different image modalities. While the DEM partially resolved these issues, it comes with important limitations. It does not allow us to sample grayscale images but only edge images. In addition because the edge images contain sparse information, the optimization by gradient descent is less accurate and leads to larger estimation errors.

Therefore we propose in this chapter a deformable model of the image intensity, which relies on the assumption that the segmentation of each image comes from a common deformable template. This model combines the advantages of DIM and DEM: it is a generative model and it is able to deal with change of intensity distribution across images. Since the segmentation of the images is unknown, it is necessary to use an iterative estimation algorithm such as the EM algorithm.

In this chapter, we first review the classical model for image segmentation, which we combine in T-DIM to a deformable tissue model. We first present the model in details and use it for the detection of anatomical landmarks.

5.1 Previous Work: Image Segmentation

In [80] a simple generative model of the image intensity was proposed to perform MR image segmentation. The joint probability of the image intensity and the tissue type is modeled as a mixture of Gaussian distributions. In order to use such a model one needs to assume that the intensities at a pixel depends only on the tissue type at this pixel. This is a strong assumption as clearly the intensity value of two neighboring pixel is correlated beyond sharing the same tissue type, this is what produces smooth images. However this is a common assumption, because it allows one to express the joint probability of the image intensity x and the tissue type z as a product over each pixel $s \in \Lambda$:

$$\begin{aligned} p(x,z) &= \prod_{s\in\Lambda} p(x(s),z(s)), \\ &= \prod_{s\in\Lambda} p(x(s)|z(s))p(z(s)), \end{aligned}$$

with $x(s)$ the random variable of the image intensity at pixel s and $z(s)$ the random discrete variable representing the tissue type at voxel s. The segmentation of the image is obtained

by maximizing its log-likelihood:

$$\ell(x) = \sum_{s \in \Lambda} \ln \sum_{j=1}^{J} p(x(s)|z(s) = j)p(z(s) = j),$$

with J the number of tissue types in the image. The conditional probability of the intensity given the tissue type $p(x(s)|z(s))$ is modeled as a Gaussian distribution $\mathcal{N}(\mu(j), \sigma^2(j))$, while $p(z(s) = j)$ represents the proportions of each tissue type in the image.

The Gaussian mixture model was a pioneer approach in intensity-based MR image segmentation. Because both the models parameters and the segmentation are unknown, Wells proposed in [80] to use the EM algorithm [19] to maximize the joint likelihood of the image and the segmentation. The EM algorithm alternates between the estimation of the model parameters and of the tissue posterior distribution. Many extensions of this model have been proposed to adapt the Gaussian mixture model to the specific challenges encountered in brain image segmentation. MR images are commonly affected by a field bias that makes the intensity distribution vary depending on the location in the image. A solution proposed in [48] consists in adding a parameter encoding the effect of the bias field at each pixel in the image. Another issue results from the coarse resolution of MR images in comparison with the brain structures which leads to mixed pixels, whose intensity results from the mixture of two tissue types. The simplest solution consists in increasing the number of tissue types in order to represent this type of pixels [46] too. Furthermore, while it is neglected in the classical Gaussian mixture model, the correlation between voxels is important in the image and some authors worked on incorporating this correlation in the form of a spatial tissue prior or atlas [26]. Finally, a family of hierarchical mixture models [64] has been proposed to perform precise brain image segmentation. In this approach, not only the image intensity is modeled as a mixture of Gaussian distribution but also each tissue type is modeled as a mixture of Gaussian, allowing one to distinguish substructures based on intensity variation even if they originally appeared in the same tissue type. One of the challenges in this type of model is to select the appropriate number of mixture components. Classical likelihood penalization methods can be used to select the number of mixture components.

Several issues arise from this statistical model. The lack of spatial correlation in the noise structure is a major drawback since it prevents us from generating smooth images by sampling from the model. This drawback is common to most of the intensity-based methods due to the common assumption of the independence of the image intensity given the hidden variables. In addition, the method inherits from the drawbacks of the EM algorithm, which can be trapped in local optima, depends on the initialization, and is computationally intense for the segmentation of large 3D images.

Many competing approaches have been proposed for image segmentation using for example machine learning techniques (e.g.[76, 58]), but also non-rigid registration to an atlas, optimization of a level function. We will not discuss these approaches in details here, as our main interest is to build a joint probability between the image and the hidden tissue type and not necessarily to provide an accurate segmentation.

5.2 A Complete Generative Model

The probabilistic Deformable Intensity Model assumes that the intensity range is common to all the images. It is often the case that the intensity distribution varies significantly between images, depending on the patient and on the scanner settings at the time of the acquisition. Scaling the image intensity would be a solution but still requires some manual adjustments because of the presence of pixels with very high intensity, which perturb classical intensity equalization algorithms. Therefore instead of introducing a manual step in the preprocessing of the image, we propose to build a model able to deal with the intensity variability. It also allows us to work with images from different modalities, i.e. in which a same tissue appears with different intensity distributions. We propose the Tissue-based Deformable Intensity Model (T-DIM), in which we assume that while the intensity distribution of a same tissue type varies depending on the image, the spatial arrangement of the tissues is common to all the images up to some deformation, parametrized by the displacement of the landmarks. Therefore we propose to build a probabilistic deformable model on the tissue-types instead of modeling directly the intensity values.

5.2.1 The Generative Model

Let us specify the notation for this model. We denote as before x and y the random real vectors representing respectively the intensity vector of an image and the vectors of the K landmark locations. x takes values in $\mathbb{R}^S$ and y takes values in $\mathbb{R}^{dK}$. Let z be a discrete random vector representing the image segmentation. $z(s)$ is the tissue type at voxel s and takes values in $\{1, \dots, J\}$, J the number of tissues. Since the segmentation of the image is unknown, z is a hidden variable. Finally, we introduce u a discrete random variable which characterizes the photometry. u takes value in $\{1, \cdots, U\}$ a set of possible photometric models. For example, it represents different acquisition settings, such as high contrast, low contrast, darker or brighter image, or even the image modality. Since we have no information about the specific acquisition procedure, this is also a hidden variable.

Figure 5.1 illustrates with a Bayesian network the complete generative model of an image. The model can be used to generate images as follows: first draw from the landmark distribution $p(y)$ a random landmark location y and from the photometric parameters distribution $p(u)$ a set of photometric parameters u. Then, given the location of the landmarks and using the deformable model $p(z|y)$, sample a segmented image, i.e. sampling the tissue type $z(s)$ at each voxel. Finally, given the photometric parameters u and the segmented image, assign an intensity value $x(s)$ to each pixel of the image domain by sampling from $p(x(s)|z(s), u)$.

The following assumptions were made in order to build the model represented by the graph of Figure 5.1. The intensity at a pixel s is assumed to be independent from the intensity at the other pixels, given the corresponding tissue type $z(s)$ and the photometric parameters u. We also assume that the intensity $x(s)$, given the tissue type $z(s)$ and the photometry u is independent from the location of the landmarks. Finally we assume that the tissue type $z(s)$ is independent from the tissue type at the other pixels, given the location of the landmarks y.

Remark 5.1. *The different random variables of the generative model have different roles. The*

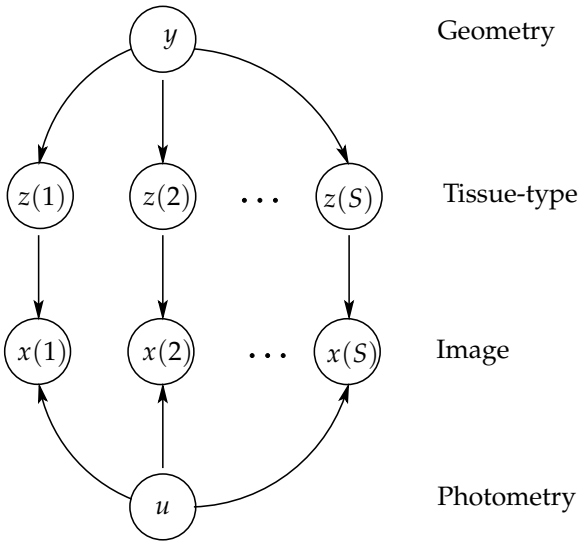


Figure 5.1: Bayesian Network representing the Deformable Tissue-Based Intensity Model. y is the location of the landmarks and characterizes the geometry, $z(1), z(2), \cdots, z(S)$ are the tissue-type at different locations in the image and $x(1), x(2), \cdots, x(S)$ the corresponding intensity. u characterizes the photometry.

intensity variables, $x(s)$, i.e. the images, are always observed. The landmark locations y are observed in the training set but need to be estimated in the testing set. Finally, the tissue-type variables $z(s)$ and the photometric variable u are never observed, neither in the training images nor in the testing ones.

The training set is composed of N images on which the landmarks have been located, $\left((x^{(1)}, y^{(1)}), \cdots, (x^{(N)}, y^{(N)})\right)$. The training set is assumed to be an independent sample of the joint distribution $p(x, y, z, u)$, in which both the segmentation z and the photometry u are missing.

Using the Bayesian network of Figure 5.1, the joint distribution can be written as:

$$p(x, y, z, u) = p(u)p(y)p(x|z, u)p(z|y). \tag{5.1}$$

Using the conditional independence assumption described above:

$$p(x, y, z, u) = p(u)p(y) \prod_{s \in \Lambda} p(x(s)|z(s), u)p(z(s)|y). \tag{5.2}$$

Therefore the log-likelihood of an image $\ell(x)$ for the T-DIM is:

$$\begin{aligned} \ell(x) = \ln p(x) &= \ln \sum_y \sum_z \sum_u p(x, y, z, u), \\ &= \ln \sum_y \sum_u p(u)p(y) \sum_z \prod_{s \in \Lambda} p(x(s)|z(s), u)p(z(s)|y), \\ &= \ln \sum_y \sum_u p(u)p(y) \prod_{s \in \Lambda} \sum_{j=1}^{J} p(x(s)|z(s) = j, u)p(z(s) = j|y). \end{aligned} \tag{5.3}$$

Hence, to ultimately compute the MLE of the landmark location $\hat{y} = \arg\max_y \ell(x)$, it is necessary to learn the model $\ell(x)$ by estimating the probability distributions involved in the likelihood function (5.3). Briefly speaking the four terms to be estimated are:

- the **prior distribution of the landmarks**, $p(y)$: since y is observed in the training set, it can be estimated from the training data;
- the **prior on the photometry**, $p(u)$: u is unobserved thus it needs to be estimated during training;
- the **photometric model**, $p(x(s)|z(s),u)$: it is modeled as a Gaussian distribution $\mathcal{N}(\mu(j,u),\sigma^2(j,u))$. The parameters of the Gaussian distributions have to be learnt during training;
- the **geometric model**, $p(z(s)|y)$: We assume that the images arise from a common probabilistic deformable tissue model $\pi(j,t), \forall t \in \Lambda_T, \forall j$. At each $t \in \Lambda_T$ the tissue type probability is modeled by a point mass function, $\sum_J \pi(j,t) = 1$. Therefore the conditional distribution $p(z(s) = j|y)$ at s is given by the point mass function in $f_y^{-1}(s)$: $\pi(j, f_y^{-1}(s))$. Learning the geometric model is equivalent to estimating the deformable template π using the set of training images.

The different parts of the model will be studied in details in the following subsections. We will start with the core pieces of the model: the geometry and the photometry.

5.2.2 Deformable Tissue Model

The geometry of the image is modeled by a deformable tissue model. It means that the distribution of the tissue types in an image is given by their distribution at the corresponding location in the template, using the image-specific deformation to set the correspondences between the template and the image. The probabilistic template is a function which assigns to each node t of a finite grid $\Lambda_T \subset \mathbb{R}^d$, a point mass function $\pi(j,t), 1 \leq j \leq J$, such that $\sum_{j=1}^{J} \pi(j,t) = 1$. The template definition is extended to a bounded domain of $\mathbb{R}^d$ by linear interpolation.

The location of the landmarks is fixed in the template $\bar{y}$, such that given a family of deformations $\mathcal{F}$, there exists a unique bijective deformation $f_{y^{(i)}} \in \mathcal{F}$ which maps the template onto the image under the constraint that $f_{y^{(i)}}(\bar{y}) = y^{(i)}$. To simplify, we assume that the template domain is exactly mapped to the image domain by $f_{y^{(i)}}$.

In the deformable model setting, the tissue types are assumed to follow a common distribution across the registered images. Since the registering deformation is characterized by the landmark correspondences, the geometry is in practice encoded by the location of the landmarks. If there are only few landmarks, it is likely that the registration will be precise around the landmarks but potentially inaccurate at further distance. This aspect is taken care of by defining a probabilistic template, able to encode the post-registration geometry variations better than a deterministic template. Figure 5.2 illustrates the deformable model of the image in the case of the tissue-based model.

Using a deformable model consists in assuming that the spatial distribution of the tissue types given the landmark location follows the distribution given in the template

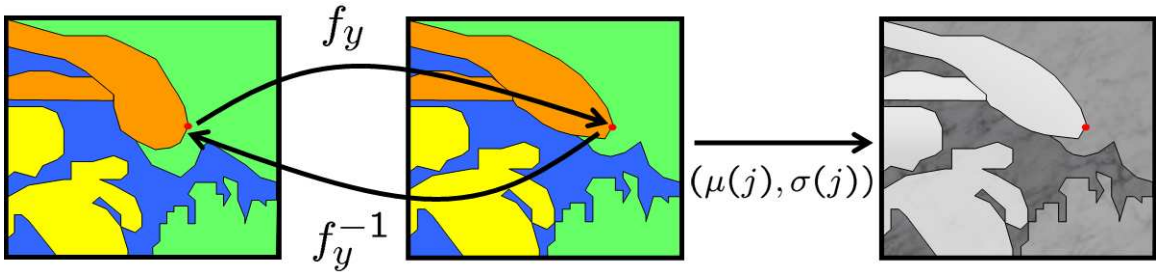


Figure 5.2: Probabilistic Tissue-based Deformable Model,. **Left to Right:** a random segmentation sampled from the template distribution, the deformed segmentation, the gray scale image

at the corresponding location:

$$\forall s \in \Lambda_i, \quad p(z^{(i)}(s) = j|y) = \pi(j, f^{-1}_{y^{(i)}}(s)). \tag{5.4}$$

5.2.3 Photometric Model

Often in medical imaging, anatomically different tissues appear in different intensity ranges. It is particularly true in the brain images for which 3 anatomically distinct tissues can be easily identified. The 3 tissue type intensity distributions are modeled as a mixture of Gaussian distributions as it is commonly done in brain segmentation methods. We make the same simplifying assumptions as in [19]: the intensity value at a pixel s depends only on the tissue type $z(s)$ and the global photometric variable u. It is assumed that the intensity distribution, given the tissue type, depends neither on the location in the image nor on the landmark location. In practice this assumption neglects the bias field of the scanner, which creates some inhomogeneity in the intensity distribution at the scale of the whole image. In different regions of the brain a same tissue may appear at somewhat different intensities. The effect should not be neglected to perform image segmentation, but since we are interested in detecting some specific points and use the pixel value in a relatively small neighborhood around the landmarks, it is reasonable to neglect the bias effect in our case.

Remark 5.2. *Even though segmenting the images is not the major aim of the illustrative application, it is still necessary to estimate the intensity distribution correctly as the matching relies on the implicit segmentation of the image.*

Given an image x and the photometric variable u,

$$\forall s \in \Lambda, \forall j \in \{1, \cdots, J\}, \quad p(x(s)|z(s) = j, u) = g(x(s); \mu(j,u), \sigma^2(j,u)), \tag{5.5}$$

with $g(x(s); \mu(j,u), \sigma^2(j,u))$ the value of the Gaussian probability density function of parameters $\mu(j,u), \sigma^2(j,u)$, taken in $x(s)$. While the model is similar to the mixture model used in image segmentation, the estimation of the Gaussian distribution parameters is coupled with the estimation of the geometry as the proportions of each tissue type comes from the deformable model.

The likelihood function of an image using the Tissue-based Deformable Intensity Model is:

$$\ell(x, y; \mu, \sigma^2, \pi) = \ln \sum_y \sum_u p(u)p(y) \prod_{s\in\Lambda} \sum_{j=1}^{J} g(x(s); \mu(j,u), \sigma^2(j,u))\pi(j, f_y^{-1}(s)). \quad (5.6)$$

5.2.4 Prior on the Landmark Locations

Since the landmark locations are observed in the training set, the estimation of $p(y)$ is performed independently from the estimation of the rest of the model. The same methods as the ones detailed for the DIM can be applied, cf. paragraph 3.3.2.

5.2.5 Prior on the Photometry

u is assumed to be a discrete variable, representing different acquisition methods. We model its distribution as a point mass function $p(u)$. Contrarily to the landmark locations, the photometry variable is not observed in the training set. Thus, its marginal distribution needs to be learnt during the training phase, simultaneously with the geometric model and the photometric parameters.

5.3 Model Selection

As usual the purpose of model selection is to estimate the model parameters using the training set of landmarked images. The T-DIM, as described in section 5.2, is a complete generative model of the joint distribution of image intensity x, the landmark location y, the tissue type or image segmentation z and the photometry u. Both x and y are observed in the training set but z and u are missing. We recall the expression of the joint distribution:

$$p(x, y, z, u) = p(u)p(y) \prod_{s\in\Lambda} p(x(s)|z(s), u)p(z(s)|y). \quad (5.7)$$

The model parameters are composed of the geometric parameters: $\pi(j,t), \forall j, \forall t$, the photometric parameters $\mu(j,u), \sigma^2(j,u), \forall j, \forall u$ and the marginal distribution of the photometric variable $p(u)$. Since the model parameters, the image segmentation and the photometric parameters are unknown and need to be estimated jointly, we propose to use the EM algorithm to perform the model selection. Because y is observed in the training set we work on the conditional model $x|y$.

5.3.1 Complete Model Estimation by the EM Algorithm

Expected log-likelihood

The expected log-likelihood is the expectation of the joint log-likelihood with respect to the posterior law of the hidden variables:

$$\ln p(x|y) = \sum_{s\in\Lambda} \ln \sum_{j=1}^{J} \pi(j, f_y^{-1}(s)) \sum_u g(x(s); \mu(j,u), \sigma^2(j,u))p(u). \quad (5.8)$$

We denote $\theta = \{\pi(j,t), \forall j, \forall t; \mu(j,u), \sigma^2(j,u), \forall j, \forall u\}$ and θ' the parameters value at the preceding iteration. The expected log-likelihood of an image is defined by:

$$\begin{aligned} Q(\theta,\theta') &= \mathbb{E}_{z,u}\left[\ln p_\theta(x,z,u|y)|x,y\right], \\ &= \sum_{z(1)=1}^{J} \cdots \sum_{z(S)=1}^{J} \sum_{u} \left[\sum_{s\in\Lambda} \ln p_\theta(x(s),z(s)|u,y) p_\theta(u)\right] p_{\theta'}(z,u|x,y), \\ &= \sum_{z(1)=1}^{J} \cdots \sum_{z(S)=1}^{J} \sum_{u} \left[\sum_{s\in\Lambda} \ln p_\theta(x(s),z(s)|u,y) p_\theta(u)\right] p_{\theta'}(u|x,y) \prod_{s\in\Lambda} p_{\theta'}(z(s)|u,x,y). \end{aligned} \tag{5.9}$$

Switching the order of the sums, we notice that the sum of $J^S \times U$ terms reduces to a sum of $J \times S \times U$ terms:

$$\begin{aligned} &Q(\theta,\theta') \\ &= \sum_{s\in\Lambda} \sum_{z(s)=j}^{J} \sum_{u} \left[\ln p_\theta(x(s),z(s)=j|u,y) p_\theta(u)\right] p_{\theta'}(u|x,y) \sum_{z\setminus z(s)} \prod_{s\in\Lambda} p_{\theta'}(z(s)=j|u,x,y), \\ &= \sum_{s\in\Lambda} \sum_{z(s)=j}^{J} \sum_{u} \left[\ln p_\theta(x(s),z(s)=j|u,y) p_\theta(u)\right] p_{\theta'}(u|x,y) p_{\theta'}(z(s)=j|u,x,y), \\ &= \sum_{s\in\Lambda} \sum_{z(s)=j}^{J} \sum_{u} \left[\ln p_\theta(x(s)|z(s)=j,u) p_\theta(z(s)=j|y) p_\theta(u)\right] p_{\theta'}(z(s)=j,u|x,y). \end{aligned} \tag{5.10}$$

Using the modeling assumptions, (5.10) becomes a sum of three terms:

$$\begin{aligned} &Q(\theta,\theta') = Q_1(\theta,\theta') + Q_2(\theta,\theta') + Q_3(\theta,\theta'), \\ &= \sum_{s}\sum_{j}\sum_{u} \left[\ln g\left(x(s);\mu(j,u),\sigma^2(j,u)\right)\right] p_{\theta'}(z(s)=j|u,x,y) p_{\theta'}(u|x,y) \\ &+ \sum_{s}\sum_{j}\sum_{u} \left[\ln \pi(j, f_y^{-1}(s))\right] p_{\theta'}(z(s)=j|u,x,y) p_{\theta'}(u|x,y) \\ &+ \sum_{s}\sum_{j}\sum_{u} \left[\ln p_\theta(u)\right] p_{\theta'}(z(s)=j|u,x,y) p_{\theta'}(u|x,y). \end{aligned} \tag{5.11}$$

We generalize to a training set of N images, denoting respectively $x_1^N, y_1^N, u_1^N, z_1^N$ the set of images, landmark locations, photometric variables, and segmentations:

$$\begin{aligned} &Q(\theta,\theta') \\ &= \sum_{i=1}^{N} \sum_{s}\sum_{j}\sum_{u} \left[\ln g\left(x^{(i)}(s);\mu(j,u^{(i)}),\sigma^2(j,u^{(i)})\right)\right] p_{\theta'}(z^{(i)}(s)=j|u^{(i)},x_1^N,y_1^N) p_{\theta'}(u^{(i)}|x_1^N,y_1^N) \\ &+ \sum_{i=1}^{N} \sum_{s}\sum_{j}\sum_{u} \left[\ln \pi(j, f_{y^{(i)}}^{-1}(s))\right] p_{\theta'}(z^{(i)}(s)=j|u^{(i)},x_1^N,y_1^N) p_{\theta'}(u^{(i)}|x_1^N,y_1^N) \\ &+ \sum_{i=1}^{N} \sum_{s}\sum_{j}\sum_{u} \left[\ln p_\theta(u^{(i)})\right] p_{\theta'}(z^{(i)}(s)=j|u^{(i)},x_1^N,y_1^N) p_{\theta'}(u^{(i)}|x_1^N,y_1^N). \end{aligned} \tag{5.12}$$

Details of the E-step

It consists in computing the posterior distribution of the hidden variables given the data x_1^N and the landmarks y_1^N. The expected log-likelihood can be further simplified, using the following proposition:

Proposition 2.

$$\forall s \in \Lambda, \forall i \in \{1, \cdots, N\}, p_{\theta'}(z^{(i)}(s)|x_1^N, y_1^N, u^{(i)}) = p_{\theta'}(z^{(i)}(s)|x^{(i)}(s), y^{(i)}, u^{(i)}).$$

Proof. To simplify the notations we choose $s = 1$ and $i = 1$,

$$\begin{aligned}
A &= p(z^{(1)}(1), u^{(1)}, x_1^N|y_1^N) \\
&= \sum_{z^{(1)}(2)} \cdots \sum_{z^{(1)}(S)} \sum_{z^{(2)}(1)} \cdots \sum_{z^{(N)}(S)} \sum_{u^{(2)}} \cdots \sum_{u^{(N)}} p(z_1^N, x_1^N, u_1^N|y_1^N), \\
&= \sum_{z^{(1)}(2)} \cdots \sum_{z^{(1)}(S)} \sum_{z^{(2)}(1)} \cdots \sum_{z^{(N)}(S)} \sum_{u^{(2)}} \cdots \sum_{u^{(N)}} \prod_{i=1}^{N} p(u^{(i)}) \prod_{s=1}^{S} p(z^{(i)}(s), x^{(i)}(s)|y^{(i)}, u^{(i)}), \\
&= p(u^{(1)}) \sum_{z^{(1)}(2)} \cdots \sum_{z^{(1)}(S)} \sum_{z^{(2)}(1)} \cdots \sum_{z^{(N)}(S)} \prod_{i=1}^{N} \prod_{s=1}^{S} p(z^{(i)}(s), x^{(i)}(s)|y^{(i)}, u^{(i)}), \\
&= p(u^{(1)}) p(z^{(1)}(1), x^{(1)}(1)|y^{(1)}, u^{(1)}) \prod_{s=2}^{S} p(x^{(1)}(s)|y^{(1)}, u^{(1)}) \prod_{i=2}^{N} \prod_{s=1}^{S} p(x^{(i)}(s)|y^{(i)}, u^{(i)}).
\end{aligned}$$

$$\begin{aligned}
B &= p(x_1^N, u^{(1)}|y_1^N) \\
&= \sum_{z^{(1)}(1)} \cdots \sum_{z^{(N)}(S)} \sum_{u^{(2)}} \cdots \sum_{u^{(N)}} p(z_1^N, x_1^N, u_1^N|y_1^N), \\
&= p(u^{(1)}) \sum_{z^{(1)}(1)} \cdots \sum_{z^{(N)}(S)} \prod_{s=1}^{S} \prod_{i=1}^{N} p(z^{(i)}(s), x^{(i)}(s)|y^{(i)}, u^{(i)}), \\
&= p(u^{(1)}) \prod_{i=1}^{N} \prod_{s=1}^{S} p(x^{(i)}(s)|y^{(i)}, u^{(i)}).
\end{aligned}$$

$$\frac{A}{B} = \frac{p(z^{(1)}(1), x^{(1)}(1)|y^{(1)}, u^{(1)})}{p(x^{(1)}(1)|y^{(1)}, u^{(1)})} = p(z^{(1)}(1)|x^{(1)}(1), y^{(1)}, u^{(1)}).$$

This is true for any s and any i. □

Using Proposition 2, the Bayes' formula and the modeling assumptions:

$$\forall s \in \Lambda, p_{\theta'}(z^{(i)}(s), u^{(i)}|x_1^N, y_1^N) = p_{\theta'}(z^{(i)}(s)|x^{(i)}(s), y^{(i)}, u^{(i)}) p_{\theta'}(u^{(i)}|x^{(i)}, y^{(i)}), \quad (5.13)$$

Using the set of parameters $\theta' \equiv \{\forall u, \forall j, \mu'(j,u), \sigma'^2(j,u); \forall t, \forall j, \pi'(j,t)\}$ and the distribution $p_{\theta'}(u)$ learnt at the preceding iteration, the posterior distribution becomes:

$$\begin{aligned}
p_{\theta'}(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)}, u^{(i)}) &\propto p_{\theta'}(x^{(i)}(s)|z^{(i)}(s) = j, u^{(i)}) p_{\theta'}(z^{(i)}(s) = j|y^{(i)}) \\
&\propto g(x^{(i)}(s); \mu'(j, u^{(i)}), \sigma'^2(j, u^{(i)})) \pi'(j, f_{y^{(i)}}^{-1}(s)), \quad (5.14)
\end{aligned}$$

$$p_{\theta'}(u^{(i)}|x^{(i)},y^{(i)}) \propto p_{\theta'}(u^{(i)}) \prod_{s\in\Lambda}\left[\sum_j p_{\theta'}(x^{(i)}(s)|z^{(i)}(s)=j,u^{(i)})p_{\theta'}(z^{(i)}(s)=j|y^{(i)})\right],$$
$$\propto p_{\theta'}(u^{(i)}) \prod_{s\in\Lambda}\left[\sum_j g(x^{(i)}(s);\mu'(j,u^{(i)}),\sigma'^2(j,u^{(i)}))\pi'(j,f^{-1}_{y^{(i)}}(s))\right] \quad (5.15)$$

The posterior distribution is computed for each image i, each tissue type j, at each location s and for each photometric model u.

Details of the M-step

The maximization of the Q-function (5.12) can be decomposed in three independent maximization problems:

$$\forall j,\forall u, \quad \max_{\mu(j,u),\sigma^2(j,u)} \sum_i\sum_s \left[\ln g\left(x^{(i)}(s);\mu(j,u),\sigma^2(j,u)\right)\right] p_{\theta'}(z(s)=j,u|x^{(i)},y^{(i)}) \quad (5.16)$$

$$\forall s, \quad \max_{\pi} \sum_i\sum_j\sum_u \left[\ln \pi(j,f^{-1}_{y^{(i)}}(s))\right] p_{\theta'}(z(s)=j,u|x^{(i)},y^{(i)}) \quad (5.17)$$

$$\max_{p_\theta(u)} \sum_i\sum_u\sum_s\sum_j [\ln p_\theta(u)]\, p_{\theta'}(z(s)=j,u|x^{(i)},y^{(i)}) \quad (5.18)$$

For the proposed model each maximization admits a closed form solution. The solution for the photometric parameters are:

$$\forall u,j, \quad \hat{\mu}(j,u) = \frac{\sum_i\sum_s x^{(i)}(s)p_{\theta'}(z(s)=j|x^{(i)}(s),y^{(i)},u)p_{\theta'}(u|x^{(i)},y^{(i)})}{\sum_i\sum_s p_{\theta'}(z(s)=j|x^{(i)}(s),y^{(i)},u)p_{\theta'}(u|x^{(i)},y^{(i)})}, \quad (5.19)$$

$$\forall u,j, \quad \hat{\sigma}^2(j,u) = \frac{\sum_i\sum_s (x^{(i)}(s)-\mu'(j,u))^2 p_{\theta'}(z(s)=j|x^{(i)}(s),y^{(i)},u)p_{\theta'}(u|x^{(i)},y^{(i)})}{\sum_i\sum_s p_{\theta'}(z(s)=j|x^{(i)}(s),y^{(i)},u)p_{\theta'}(u|x^{(i)},y^{(i)})}. \quad (5.20)$$

The number of photometric intensity values U and the number of Gaussian distributions J used to describe the intensity variation is manually fixed before learning the model parameters. If $U < N$, several images may contribute to the estimation of the photometric parameters corresponding to the intensity model u. Their contribution to the estimation of the parameters is weighted using the posterior probability of the intensity model for each image. The images that are unlikely to come from the intensity model u will not contribute to the estimation of $\mu(j,u),\sigma^2(j,u)$.

The solution of the maximization (5.18) is:

$$\hat{p}_\theta(u) \propto \sum_i p_{\theta'}(u|x^{(i)},y^{(i)}),$$
$$\propto p_{\theta'}(u)\sum_i\prod_{s\in\Lambda}\sum_j g(x^{(i)}(s);\mu'(j,u),\sigma'^2(j,u))\pi'(j,f^{-1}_{y^{(i)}}(s)). \quad (5.21)$$

The point mass function representing the distribution of the photometric variable u is adjusted at each iteration, adding some weight to the intensity models that explain the best the observed data. A normalization ensures that the result is a point mass distribution.

The template update comes from the maximization of (5.17). Since each image i comes from a specific deformation of the template, the template value used as a proportion coefficient for a fixed location $s \in \Lambda$ differs depending on the image and more specifically on the deformation. To overcome this difficulty the sum over each image is approximated using the approximated integral change of variable: $s = f_{y^{(i)}}(t)$ (cf. Chapter 3 for the details of this change of variable).

$$\begin{aligned}
&\sum_i \sum_s \sum_j \left[\ln \pi(j, f_{y^{(i)}}^{-1}(s))\right] \sum_u p_{\theta'}(z(s) = j, u | x^{(i)}, y^{(i)}), \\
&\simeq \sum_i \sum_j \sum_{t \in \Lambda_T} [\ln \pi(j,t)] \sum_u p_{\theta'}(z(f_{y^{(i)}}(t)), u | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) |J_{f_{y^{(i)}}}(t)|, \\
&= \sum_j \sum_{t \in \Lambda_T} \ln \pi(j,t) \sum_i \sum_u p_{\theta'}(z(f_{y^{(i)}}(t)), u | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) |J_{f_{y^{(i)}}}(t)|
\end{aligned} \tag{5.22}$$

Exchanging the two first sums, the maximization can be performed at each template location $t \in \Lambda_T$ independently:

$$\forall t, \forall j, \quad \hat{\pi}(j,t) \propto \sum_i \sum_u p_{\theta'}(z(f_{y^{(i)}}(t)) = j, u | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) |J_{f_{y^{(i)}}}(t)|. \tag{5.23}$$

The update is a weighted average of the posterior probabilities of each tissue type at each location t. The contributions of the images are weighted by the local Jacobian value. Images whose grid locally contracts ($|J| < 1$) during the registration have a smaller contribution than images whose grid expands ($|J| > 1$) locally. In region with no grid deformation ($|J| = 1$), the update consists exactly in computing the average proportions of the different tissue types. Notice that while the change of variable leads to an important simplification of the maximization, it becomes necessary to use some interpolation method on the image support. Indeed, even though the statistical model is defined at the observed data values of the image, the final update expression is defined on the template grid, such that the intensity value $x^{(i)}(f_{y^{(i)}}(t))$ may not be observed in practice.

5.4 Prediction of the Landmark Location

The prediction problem consists of locating y in a new image x, using the model learnt previously in the training phase. The specificity of the tissue-based model is that the tissue $z(s)$ at each location is unknown, since the segmentation of the image is not given. Using the aforementioned model, the log-likelihood of a new image is given by:

$$\ell(x,y) = \ln p(y) + \sum_{s \in \Lambda} \ln \sum_u p(u) \sum_j g(x(s); \mu(j,u), \sigma^2(j,u)) \pi(j, f_y^{-1}(s)). \tag{5.24}$$

The maximum likelihood estimator is used to predict the location of the landmarks in the new image. The model parameters $\{\forall j, \forall u, \mu(j,u), \sigma^2(j,u); \forall j, \forall t, \pi(j,t)\}$ and the marginal distributions $p(u)$ and $p(y)$ were learnt during the training phase. Therefore we can optimize directly the likelihood function, using a gradient method since the maximum does not have a simple closed form expression.

Because the likelihood involves the computation of the inverse deformation, we perform the approximated integral change of variable $t = f_y^{-1}(s)$. The expression becomes:

$$\ell(x,y) \simeq \ln p(y) + \sum_{t\in\Lambda_T} \left[\ln \sum_u p(u) \sum_j g(x(f_y(t)); \mu(j,u), \sigma^2(j,u))\pi(j,t) \right] |J_{f_y}(t)|. \quad (5.25)$$

After the change of variable, the intensity function $x(f_y(t))$ and the Jacobian depend on the location of the landmarks. The gradient computation gives:

$$\begin{aligned}
&\frac{\partial \ell(x,y)}{\partial y} \\
&= \sum_{t\in\Lambda_T} |J_{f_y}(t)| \frac{\partial x(f_y(t))}{\partial y} \cdot \frac{\sum_u p(u) \sum_j g(x(f_y(t)); \mu(j,u), \sigma^2(j,u))\pi(j,t) \frac{\mu(j,u)-x(f_y(t))}{\sigma^2(j,u)}}{\sum_u p(u) \sum_j g(x(f_y(t)); \mu(j,u), \sigma^2(j,u))\pi(j,t)} \\
&+ \sum_{t\in\Lambda_T} \left[\ln \sum_u p(u) \sum_j g(x(f_y(t)); \mu(j,u), \sigma^2(j,u))\pi(j,t) \right] \frac{\partial |J_{f_y}(t)|}{\partial y} + \frac{\partial p(y)}{\partial y} \cdot \frac{1}{p(y)} \quad (5.26)
\end{aligned}$$

Starting at $y = \bar{y}$, the gradient ascent is coupled to a line search to determine at each iteration the optimal step size.

Remark 5.3. *In order to compute the maximum likelihood estimate, one needs to integrate over the hidden variables z and u. Because both hidden variables are discrete, the computation can be carried out. However if the number of levels of any of these variables increases, the computational cost increases significantly. It can become an important limitation, specially with rather large images. If the distribution of one of these variables is continuous, one still needs to integrate with respect to the hidden variable. There is no easy way to compute this integral. A common numerical approximation assumes that the distribution of the hidden variables is a Dirac function at the current estimated mode. This approximation is equivalent to model u as a nuisance parameter. Another solution proposed in [1] consists in estimating the distribution by a Monte Carlo Markov Chain approximation and using the Stochastic Approximation of the EM algorithm to solve the optimization problem. The computational cost of such a procedure is quite large and would prevent from working with large images.*

Algorithm 5.5 summarizes the algorithm associated to the complete generative model.

5.4.1 Combining Segmentation and Registration

Two main strategies have been proposed in brain MRI segmentation. The first set of methods is essentially pixel-wise and assigns based on intensity for example each pixel individual to one of the objects to be segmented. This type of approaches, pioneered by [19, 80], can be used as in [46] to perform precise segmentation. The competing template-based approach aims at warping a segmented image or an atlas onto the image to be segmented, or the opposite to deform the image so that it looks similar to the template. This approach allows to define regions that span different intensity.

The T-DIM belongs to a new set of models combining image segmentation and template-based registration. If the images are registered onto each other, the T-DIM boils down to a simple mixture Gaussian model (for all i, $f_{y^{(i)}}$ is the identity). Similarly, if the image

Algorithm 5.5 Deformable Tissue-based Intensity Model

LEARNING

Let (x_1^N, y_1^N) be a training set, $\theta = \{\pi(j,t), \forall j, \forall t; \mu(j,u), \sigma^2(j,u), \forall j, \forall u\}$ the set of photometric and geometric parameters, and $p_\theta(u)$ the distribution of the photometric variable.

Initialize $\forall j, \forall u$, $\mu(j,u), \sigma^2(j,u), \pi(j,t), \forall j, \forall t \in \Lambda_T$, and $p_\theta(u)$.
Iterate until convergence

- **E-step:** $\forall j, \forall u, \forall i, \forall s$, compute $p_\theta(z(s) = j, u | x^{(i)}, y^{(i)})$:

$$p_\theta(z(s) = j | x^{(i)}(s), y^{(i)}, u) \propto g(x^{(i)}(s); \mu(j,u), \sigma^2(j,u)) \pi(j, f^{-1}_{y^{(i)}}(s))$$

$$p_\theta(u | x^{(i)}, y^{(i)}) \propto p_\theta(u) \prod_{s \in \Lambda} \sum_j g(x^{(i)}(s); \mu(j,u), \sigma^2(j,u)) \pi(j, f^{-1}_{y^{(i)}}(s))$$

- **M-step:**
  - Update the photometric parameters,

$$\forall j, u, \qquad \mu(j,u) \leftarrow \frac{\sum_i \sum_s x^{(i)}(s) p_\theta(j, u | x^{(i)}, y^{(i)})}{\sum_i \sum_s p_\theta(j, u | x^{(i)}, y^{(i)})},$$

$$\forall j, u, \qquad \sigma^2(j,u) \leftarrow \frac{\sum_i \sum_s \left(x^{(i)}(s) - \mu(j,u) \right)^2 p_\theta(j, u | x^{(i)}, y^{(i)})}{\sum_i \sum_s p_\theta(j, u | x^{(i)}, y^{(i)})},$$

  - Update the distribution of the photometric model

$$\forall u, \quad p_\theta(u) \propto \sum_i p_\theta(u | x^{(i)}, y^{(i)}),$$

  - Update the template estimate,

$$\forall j, t, \quad \pi(j,t) \propto \sum_i |J_{f_{y^{(i)}}}(t)| \sum_u p_\theta(z(s) = j, u | x^{(i)}, y^{(i)}).$$

TESTING

Let x be a testing image and $\forall t, \forall j, \pi(j,t), \forall j, \forall u, \mu(j,u), \sigma^2(j,u)$, $p(u)$ the parameters and distributions learnt during training,

Initialize $y \leftarrow \bar{y}$
Iterate until convergence

- **Compute** the gradient direction $\frac{\partial \ell(x,y)}{\partial y}$ using (5.26),
- **Find** the optimal step size,

$$a \leftarrow \arg\max_{a \in \mathbb{R}^+} \ell \left(x, y + a \frac{\partial \ell(x,y)}{\partial y} \right),$$

- **Update** the location of the landmarks,

$$y \leftarrow y + a \cdot \frac{\partial \ell(x,y)}{\partial y}.$$

segmentation is known, the model boils down to a template-based registration model. The combined model is aimed at performing simultaneous segmentation and registration of images. In the practical example we present, the registration is only local since the purpose is to detect landmarks. Recent efforts have been made to perform the registration of the image onto the atlas and the image segmentation simultaneously, using combined intensity- and registration-based models, see e.g.[61, 4, 27, 60, 79]. Notice though that the common objective of these methods is to perform segmentation while in our case, we are more interested in the registration result and the segmentation is a by-product of the registration algorithm. In the latter cited work, the template or the atlas is given and was obtained from complete segmented images. In our work, the template is estimated from the training set which is only composed of images in which few landmarks have been located.

5.5 Image-specific Photometric Parameters

In the preceding model, the images are modeled as samples of the joint distribution $p(x, y, z, u)$. The learning phase allows us to estimate this joint distribution and thus, if desired, to generate random images. The model relies on a fixed number of photometric models U, learnt during training. Because u is modeled as a hidden variable, one needs to integrate with respect to u in order to optimize the log-likelihood. This leads to a computationally involved gradient expression (5.26). The choice of the number of possible photometric models is balanced between reducing the computational load and capturing the training image variability. Whichever the number of values of u, if the new image intensity distribution does not correspond to the intensity distribution in the training set, the detection of landmarks will be prone to errors.

5.5.1 Parameter versus Hidden Variable

One way to address these concerns is to model u as a nuisance parameter rather than as a hidden variable. In our case it makes sense to model it this way, because the intensity parameters may vary tremendously between images. In terms of likelihood, modeling u as a nuisance parameter means that it is enough to work with the conditional distribution:

$$\begin{aligned} \ln p(x, y|u) &= \ln p(y) + \ln \sum_z p(x, z|y, u), \\ &= \ln p(y) + \sum_{s\in\Lambda} \ln \sum_{z(s)} p(x(s)|z(s), u) p(z(s)|y). \end{aligned} \tag{5.27}$$

During training, the problem is reduced to estimating on one hand the landmark distribution and on the other hand the conditional joint probabilities $p(x|z, u)$ and $p(z|y)$. As for the testing algorithm, the predicted landmark location is obtained by optimizing the image and the landmark likelihood $p(x, y|u)$, with respect to y and the nuisance parameters $\mu(j, u), \sigma^2(j, u)$. The joint estimation is carried out by the EM algorithm.

5.5.2 Model Estimation by the EM Algorithm

Expected log-likelihood

We recall the expression of the joint probability of an image:

$$\ln p(x,y|u) = \ln p(y) + \sum_{s\in\Lambda} \ln \sum_{j=1}^{J} \pi(j, f_y^{-1}(s)) g(x(s); \mu(j,u), \sigma^2(j,u)). \tag{5.28}$$

Using the same reasoning as in (5.3.1), we write the expected log-likelihood of a sample of N images for which the location of the landmarks y has been identified. We denote x_1^N the set of N images and use similar notations for the set of landmark locations y_1^N, segmentations z_1^N. We denote θ the model parameters (π) and the nuisance parameters (μ, σ), θ' their estimate at the preceding iteration,

$$\begin{aligned} Q(\theta,\theta') &= \mathbb{E}_z \left[\ln p_\theta(x_1^N, y_1^N, z_1^N) | x_1^N, y_1^N, u_1^N \right], \\ &= \sum_i \ln p(y^{(i)}) \\ &+ \sum_i \sum_s \sum_j \left[\ln p_\theta(x^{(i)}(s), z^{(i)}(s) = j | y^{(i)}, u^{(i)}) \right] p_{\theta'}(z^{(i)}(s) = j | x_1^N, y_1^N, u_1^N). \end{aligned} \tag{5.29}$$

Plugging the expression of the log-likelihood (5.28) into (5.29) and developing the logarithm, $Q(\theta,\theta')$ is a sum of three terms:

$$\begin{aligned} Q(\theta,\theta') &= Q_1(\theta,\theta') + Q_2(\theta,\theta') + Q_3(\theta,\theta'), \\ &= \sum_i \ln p(y^{(i)}) && (5.30) \\ &+ \sum_i \sum_s \sum_j \left[\ln g\left(x^{(i)}(s); \mu(j,u), \sigma^2(j,u) \right) \right] p_{\theta'}(z^{(i)}(s) = j | x_1^N, y_1^N, u_1^N) && (5.31) \\ &+ \sum_i \sum_s \sum_j \left[\ln \pi(j, f_{y^{(i)}}^{-1}(s)) \right] p_{\theta'}(z^{(i)}(s) = j | x_1^N, y_1^N, u_1^N). && (5.32) \end{aligned}$$

Details of the E-step

Similarly to Proposition 2,

$$\forall s \in \Lambda, \forall i \in \{1, \cdots, N\}, \quad p_{\theta'}(z^{(i)}(s) | x_1^N, y_1^N, u_1^N) = p_{\theta'}(z^{(i)}(s) | x^{(i)}(s), y^{(i)}, u^{(i)}).$$

The E-step consists in computing the posterior distribution of the tissue type for each image, each tissue, and at each location, using the parameters learnt at the preceding iteration.

$$\begin{aligned} p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}, u^{(i)}) &\propto p_{\theta'}(x^{(i)}(s) | z^{(i)}(s) = j, u^{(i)}) p_{\theta'}(z^{(i)}(s) = j | y^{(i)}) \\ &\propto g(x^{(i)}(s); \mu'(j,u^{(i)}), \sigma'^2(j,u^{(i)})) \pi'(j, f_{y^{(i)}}^{-1}(s)), && (5.33) \\ p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}) &\propto g(x^{(i)}; \mu'(j,i), \sigma'^2(j,i)) \pi'(j, f_{y^{(i)}}^{-1}(s)). && (5.34) \end{aligned}$$

Since the nuisance parameters are learnt on each image, we simplify the notation of the Gaussian distribution parameters. Instead of denoting $u^{(i)}$ the nuisance parameters, we use $\mu(j,i)$ and $\sigma(j,i)$ and denote the posterior probability $p_{\theta'}(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)}, u^{(i)})$ by $p_{\theta'}(z^{(i)}(s) = j|x^{(i)}(s), y^{(i)})$.

Details of the M-step

The M-step consists in maximizing each term of $Q(\theta, \theta')$ with respect to $p(y)$, $\pi(j,t)$, $\mu(j,i)$, $\sigma^2(j,i)$ for all $i \in \{1, \cdots, N\}$, $j \in \{1, \cdots, J\}$ and for all $t \in \Lambda_T$. The maximization of (5.30) is in fact simply learning the marginal distribution of the landmarks, and does not depend on the preceding estimate of the parameters. Therefore it can be done independently as proposed in (3.3.2). The maximization of (5.32) admits closed form solutions, such that the update of the photometric parameters are:

$$\forall i,j, \quad \hat{\mu}(j,i) = \frac{\sum_s x^{(i)}(s) p_{\theta'}(z(s) = j|x^{(i)}(s), y^{(i)})}{\sum_s p_{\theta'}(z(s) = j|x^{(i)}(s), y^{(i)})}, \tag{5.35}$$

$$\forall i,j, \quad \hat{\sigma}^2(j,i) = \frac{\sum_i \sum_s (x^{(i)}(s) - \mu'(j,i))^2 p_{\theta'}(z(s) = j|x^{(i)}(s), y^{(i)})}{\sum_s p_{\theta'}(z(s) = j|x^{(i)}(s), y^{(i)})}. \tag{5.36}$$

Notice that contrarily to the update expression in the complete generative model (5.19), the update is computed using the intensities observed in image i only.

The update of the template does not change compared to the complete generative model, except that there is no need to sum over all possible values of u:

$$\forall t,j, \quad \pi(j,t) \propto \sum_i p_{\theta'}(z(f_{y^{(i)}}(t)) = j|x^{(i)}(f_{y^{(i)}}(t)), y^{(i)})|J_{f_{y^{(i)}}}(t)|. \tag{5.37}$$

5.5.3 Landmark Detection

We propose to use the Maximum Likelihood Estimator to predict the location of the landmarks. Denoting $\tilde{\theta}$ the set of nuisance parameters:

$$\begin{aligned} \{\hat{y}, \hat{\tilde{\theta}}\} &= \arg\max_{y,\tilde{\theta}} \ln p(y) + \ln p_{\tilde{\theta}}(x|y), \\ &= \arg\max_{y,\tilde{\theta}} \ln p(y) + \sum_s \ln \sum_j p_{\tilde{\theta}}(x(s)|z(s) = j) p(z(s) = j|y). \end{aligned} \tag{5.38}$$

The Modified EM algorithm, introduced in Chapter 4, can be used to solve this estimation problem. The Q-function is:

$$\begin{aligned} &Q(\tilde{\theta}, y;, \tilde{\theta}', y') \\ &= \mathbb{E}_z \left[\ln p_{\tilde{\theta}}(x, y, z)|x, y\right], \\ &= \sum_s \sum_j \left[\ln g(x(s); \mu(j), \sigma^2(j))\, \pi(j, f_y^{-1}(s))\, p(y)\right] p_{\tilde{\theta}'}(z(s) = j|x(s), y'). \end{aligned} \tag{5.39}$$

During the E-step the posterior distribution of each tissue type j is computed at each pixel s using the preceding iteration estimates:

$$\forall j, \forall s, \quad p_{\tilde{\theta}'}(z(s) = j | x(s), y') \propto g(x(s); \mu'(j), \sigma'^2(j)) \pi(j, f_{y'}^{-1}(s)). \tag{5.40}$$

The M-step is composed of the update of the photometric parameters, and a gradient ascent of the likelihood function with respect to y. The update of the photometric parameters $\hat{\tilde{\theta}} = (\hat{\mu}(j), \hat{\sigma}^2(j), 1 \leq j \leq J)$ is the same as in the training phase (5.35) and (5.36).

The optimization with respect to y is performed on the likelihood function, using the updated values of the nuisance parameters:

$$\begin{aligned} \hat{y} &= \arg\max_y \ln p_{\hat{\tilde{\theta}}}(x, y) \\ &= \ln p(y) + \sum_{s \in \Lambda} \ln \sum_{j=1}^{J} \pi(j, f_y^{-1}(s)) g(x(s); \hat{\mu}(j), \hat{\sigma}^2(j)). \end{aligned} \tag{5.41}$$

Since the inverse deformation f_y^{-1} appears in the likelihood function, we perform the approximated integral change of variable as described in Chapter 3:

$$\hat{y} = \arg\max_y \left\{ \ln p(y) + \sum_{t \in \Lambda_T} |J_{f_y}(t)| \ln \sum_{j=1}^{J} \pi(j, t) g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j)) \right\}. \tag{5.42}$$

The gradient of the likelihood function can be written analytically:

$$\begin{aligned} \frac{\partial \ell(x, y; \hat{\tilde{\theta}})}{\partial y} &= \frac{\partial p(y)}{\partial y} \cdot \frac{1}{p(y)} + \\ &+ \sum_{t \in \Lambda_T} |J_{f_y}(t)| \frac{\partial x(f_y(t))}{\partial y} \sum_{j=1}^{N} \frac{\hat{\mu}(j) - x(f_y(t))}{\hat{\sigma}^2(j)} \cdot \frac{\pi(j, t) g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j))}{\sum_{j=1}^{N} \pi(j, t) g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j))} \\ &+ \sum_{t \in \Lambda_T} \frac{\partial |J_{f_y}(t)|}{\partial y} \ln \sum_{j=1}^{J} \pi(j, t) g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j)). \end{aligned} \tag{5.43}$$

The gradient expression is similar to the expression of the gradient of the complete generative model (5.26) except for the sum over all possible values of u. In terms of computation the gradient expression is less complex, but the optimization method requires to loop on the estimation of the photometric parameters as well.

Algorithm 5.6 summarizes the training and testing algorithms derived from the Tissue-based Deformable Intensity Model when the photometry is encoded as a nuisance parameter.

5.6 Decoupling Photometry and Geometry

In this section we introduce mostly for comparison, a sequential algorithm that can be used for landmark detection. Even though this algorithm does not fully reflect the structure of the image model as it neglects the connection between the geometry and the photometry, it provides an easy and efficient way to initialize the joint algorithms. The main assumption to carry out the decoupling model estimation is that the photometry of each image is independent from the geometry and that its estimation can be performed for each image independently before learning the template.

Algorithm 5.6 Tissue-based Deformable Intensity Model (Nuisance Parameters)

LEARNING

Let (x_1^N, y_1^N) be a training set, $\theta = \{\forall j, \forall i, \mu(j,i), \sigma^2(j,i); \forall j, \forall t, \pi(j,t)\}$ the set of photometric and geometric parameters.

Initialize $\forall j, \forall i,\ \mu(j,i), \sigma^2(j,i)$, and $\forall j, \forall t \in \Lambda_T, \pi(j,t)$
Iterate until convergence

- **E-step:** compute:
$$\forall j, \forall i, \forall s, \quad p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}) \propto g(x^{(i)}(s); \mu(j,i), \sigma^2(j,i)) \pi(j, f_{y^{(i)}}^{-1}(s))$$
- **M-step:**
  - Update the photometric parameters,
$$\forall j, i, \qquad \mu(j,i) \leftarrow \frac{\sum_s x^{(i)}(s) p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)})}{\sum_s p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)})},$$
$$\forall j, i, \qquad \sigma^2(j,i) \leftarrow \frac{\sum_s \left(x^{(i)}(s) - \mu(j,i)\right)^2 p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)})}{\sum_s p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)})},$$
  - Update the template estimate,
$$\forall j, t, \quad \pi(j,t) \propto \sum_i |J_{f_{y^{(i)}}}(t)| p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}).$$

TESTING

Let x be a testing image of unknown photometric parameters $\tilde{\theta} = (\mu(j), \sigma^2(j), 1 \leq j \leq J)$ and π the parameters learnt during training,

Initialize $\forall j, \mu(j), \sigma^2(j)$ and $y \leftarrow \bar{y}$
Iterate until convergence

- **E-step:**
$$\forall j, \forall s, \qquad p_{\tilde{\theta}}(z(s) = j | x(s), y) \propto g(x(s); \mu(j), \sigma^2(j)) \pi(j, f_y^{-1}(s)).$$
- **M-step:**
  - **Update** the photometric parameters
$$\forall j, \qquad \mu(j) \leftarrow \frac{\sum_s x(s) p_{\tilde{\theta}}(z(s) = j | x(s), y)}{\sum_s p_{\tilde{\theta}}(z(s) = j | x(s), y)},$$
$$\forall j, \qquad \sigma^2(j) \leftarrow \frac{\sum_s \left(x^{(i)}(s) - \mu(j)\right)^2 p_{\tilde{\theta}}(z(s) = j | x(s), y)}{\sum_s p_{\tilde{\theta}}(z(s) = j | x(s), y)},$$
  - **Compute** the gradient direction $\frac{\partial \ell}{\partial y}(x, y; \tilde{\theta})$ from (5.43).
  - **Update** the location of the landmarks,
$$y \leftarrow y + a \cdot \frac{\partial \ell(x, y | \tilde{\theta})}{\partial y}, \quad \text{with} \quad a \leftarrow \arg\max_{a \in \mathbb{R}^+} \ell\left(x, y + a \frac{\partial \ell(x, y; \tilde{\theta})}{\partial y}; \tilde{\theta}\right),$$

5.6.1 Model Description

The image model is identical to the Tissue-based Deformable Intensity Model with nuisance parameters, which means that the log-likelihood function is:

$$\ell(x, y; \mu, \sigma^2, \pi) = \ln p(y) + \sum_{s \in \Lambda} \ln \sum_{j=1}^{J} \pi(j, f_y^{-1}(s)) g(x(s); \mu(j), \sigma(j)^2). \tag{5.44}$$

The intensity distribution is modeled as a mixture of Gaussian distributions, which can be learned from each image. Therefore in the training algorithm, the joint optimization is approximated by a 2-step maximization scheme:

1. For each image i, estimate the photometric parameters $\mu(j,i), \sigma^2(j,i)$,
2. Learn the model parameters $\pi(j,t)$, using the training images and the learnt photometric parameters.

As for the testing algorithm, it is also assumed that the geometry and photometry parameters can be estimated independently. Given a new image x

1. Estimate its photometric parameters $\mu(j), \sigma^2(j)$,
2. With π the template learnt during training, find the location of the landmarks that maximizes the likelihood (5.44).

5.6.2 Model Selection

Photometric Model Estimation

The intensity of the image x is modeled as a mixture of Gaussian distributions, assuming conditional independence of the voxels. This is the same model as for image segmentation. We use the EM algorithm to estimate the photometric parameters $(\mu(j,i), \sigma^2(j,i))$ and the mixture proportions $\alpha(j,i)$ for each image independently:

$$p(x^{(i)}) = \prod_{s \in \Lambda_i} \sum_{j=1}^{J} g(x^{(i)}(s), \mu(j,i), \sigma^2(j,i)) \alpha(j,i), \text{ with } \sum_j \alpha(j,i) = 1. \tag{5.45}$$

Denoting $\theta = (\mu(j,i), \sigma^2(j,i), \alpha(j,i), \forall i, \forall j)$, the Q-function is:

$$\begin{aligned} Q(\theta, \theta') = & \sum_s \sum_j \left[\ln g \left(x^{(i)}(s); \mu(j,i), \sigma^2(j,i) \right) \right] p_{\theta'}(z(s) = j | x^{(i)}(s)) \\ & + \sum_s \sum_j \left[\ln \alpha(j,i) \right] p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s)). \end{aligned} \tag{5.46}$$

The posterior distribution of the tissue type is given by

$$\forall j, s, \quad p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s)) \propto g(x^{(i)}(s); \mu'(i,j), \sigma'^2(i,j)) \alpha'(j,i). \tag{5.47}$$

The maximization with respect to the photometric parameters has a closed form solution:

$$\forall j, \quad \hat{\alpha}(j,i) \propto \sum_s p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s)), \tag{5.48}$$

$$\forall j, \quad \hat{\mu}(j,i) = \frac{\sum_s x^{(i)}(s) p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s))}{\sum_s p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s))}, \tag{5.49}$$

$$\forall j, \quad \hat{\sigma}^2(j,i) = \frac{\sum_s (x^{(i)}(s) - \mu'(j,i))^2 p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s))}{\sum_s p_{\theta'}(z^{(i)}(s) = j | x^{(i)}(s))}. \tag{5.50}$$

Estimating the Geometry

The geometry is estimated in a second independent step, consisting of finding the proportions of each tissue types for all $t \in \Lambda_T$. Since the tissue type is unobserved, we use an EM algorithm at each $t \in \Lambda_T$ to deal with the missing variable z. The corresponding Q-function is:

$$Q(\pi, \pi') = \sum_{i=1}^{N} \sum_{s \in \Lambda} \sum_{j=1}^{J} \left[\ln g(x^{(i)}(s); \hat{\mu}(j,i), \hat{\sigma}^2(j,i)) \pi(j, f_{y^{(i)}}^{-1}(s)) \right] p_{\pi'}(z(s) = j | x^{(i)}(s), y^{(i)}). \tag{5.51}$$

We use the Gaussian parameters previously learnt and perform the usual approximated integral change of variable $t = f_{y^{(i)}}^{-1}(s)$:

$$Q(\pi, \pi') \simeq \sum_{i=1}^{N} \sum_{t \in \Lambda_T} \sum_{j=1}^{J} |J_{f_{y^{(i)}}}(t)| \ln g(x^{(i)}(f_{y^{(i)}}(t)); \hat{\mu}(j,i), \hat{\sigma}^2(j,i)) \pi(j,t) \times p_{\pi'}(z(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}). \tag{5.52}$$

In the E-step, we compute the posterior distribution of $z(f_{y^{(i)}}(t))$:

$$\begin{aligned} \forall i, \forall j, \forall t, \quad & p_{\pi'}(z(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) \\ & \propto g(x^{(i)}(f_{y^{(i)}}(t)); \mu(j,i), \sigma^2(j,i)) \pi'(j,t), \end{aligned} \tag{5.53}$$

The M-step has a closed form solution in π:

$$\forall t, \forall j, \quad \pi(j,t) \propto \sum_{i=1}^{N} |J_{f_{y^{(i)}}}(t)| p_{\pi'}(z(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}). \tag{5.54}$$

Intuitively, the algorithm consists in registering the images first. Then, based on the N observations at a fixed location t, it estimates the proportion of each tissue type, using the photometric parameters learnt during the photometry learning step. The Jacobian of the registering deformation weights the pixels depending upon the local deformation of the grid. Notice though that since the images are defined on a finite grid, the intensity values needed to estimate the template are not necessarily observed. We use linear interpolation to overcome this problem in practical applications.

5.6.3 Landmark Detection

Similarly to the training, the testing algorithm is composed of 2 independent steps: the photometry estimation and the detection of the landmarks.

Photometric Model Estimation

The estimation of the photometric parameters on a new image x is obtained via the EM algorithm exactly as it was performed for the image of the training set (cf. 5.6.2), using the same number of Gaussian distributions. We denote $\hat{\mu}(j)$ and $\hat{\sigma}^2(j)$ the estimated parameters.

Landmark Location Estimation

Once the photometry model has been estimated, the problem reduces to finding y that maximizes

$$\ell(x,y) = \ln p(y) + \sum_{s\in\Lambda} \ln \sum_{j=1}^{J} g(x(s);\hat{\mu}(j),\hat{\sigma}^2(j))\pi(j,f_y^{-1}(s)),$$

or,

$$\ell(x,y) \simeq \ln p(y) + \sum_{t\in\Lambda_T} |J_{f_y}(t)| \ln \sum_{j=1}^{J} g(x(f_y(t));\hat{\mu}(j),\hat{\sigma}^2(j))\pi(j,t), \tag{5.55}$$

after the change of variable $t = f_y^{-1}(s)$.

The gradient with respect to y is computed as follows:

$$\begin{aligned}\frac{\partial \ell(x,y)}{\partial y} &= \frac{\partial p(y)}{\partial y}\cdot\frac{1}{p(y)} + \\ &+ \sum_{t\in\Lambda_T} |J_{f_y}(t)| \frac{\partial x(f_y(t))}{\partial y} \sum_{j=1}^{J} \frac{\hat{\mu}(j) - x(f_y(t))}{\hat{\sigma}^2(j)} \cdot \frac{\pi(j,t)g(x(f_y(t));\hat{\mu}(j),\hat{\sigma}^2(j))}{\sum_{j=1}^{J}\pi(j,t)g(x(f_y(t));\hat{\mu}(j),\hat{\sigma}^2(j))} \\ &+ \sum_{t\in\Lambda_T} \frac{\partial |J_{f_y}(t)|}{\partial y} \ln \sum_{j=1}^{J} \pi(j,t)g(x(f_y(t));\hat{\mu}(j),\hat{\sigma}^2(j)).\end{aligned} \tag{5.56}$$

The gradient expression (5.56) is identical to the gradient in the case of a joint model with nuisance parameters, (5.43). A steepest gradient ascent is used to find a local maximum of the likelihood function, optimizing the step at each iteration using a line search algorithm. Because the optimization is performed sequentially, the gradient ascent stops as soon as a local maximum has been found. The whole decoupled detection algorithm corresponds to one iteration of the joint estimation algorithm detailed in Algorithm 5.6.

While the disjoint model allows fast computations by separating the optimization in two independent maximizations, the solution does not need to coincide with the maximum of the joint optimization problem.

Algorithm 5.7 summarizes the learning algorithm, while Algorithm 5.8 summarizes the detection of the landmarks in a new image.

Algorithm 5.7 Decoupled Model: Learning

LEARNING

Let (x_1^N, y_1^N) be a training set, $\forall i, \theta_i = \{\forall j, \mu(j,i), \sigma^2(j,i), \alpha(j,i)\}$ the set of photometric parameters of image i, $\{\pi(j,t), \forall j, \forall t \in \Lambda_T\}$ the template.

Photometry Estimation: Initialize θ_i and iterate until convergence,

- **E-step:** Update the posterior distribution,

$$\forall i, \forall j, \forall s \quad p(z^{(i)}(s) = j | x^{(i)}(s)) \propto g(x^{(i)}(s); \mu(j,i), \sigma^2(j,i)) \alpha(j,i),$$

- **M-step:**
  - Update the proportions,

$$\forall j, \quad \alpha(j,i) \propto \sum_s p_\theta(z^{(i)}(s) = j | x^{(i)}(s))$$

  - Update the photometric parameters,

$$\forall i, \forall j, \quad \mu(j,i) \leftarrow \frac{\sum_s x^{(i)}(s) p_\theta(z^{(i)}(s) = j | x^{(i)}(s))}{\sum_s p_\theta(z^{(i)}(s) = j | x^{(i)}(s))},$$
$$\forall j, \quad \sigma^2(j,i) \leftarrow \frac{\sum_s (x^{(i)} - \mu(j,i))^2(s) p_\theta(z^{(i)}(s) = j | x^{(i)}(s))}{\sum_s p_\theta(z^{(i)}(s) = j | x^{(i)}(s))}.$$

Geometry Estimation: Initialize $\pi(j,t), \forall j, \forall t$ and iterate until convergence,

- E-step: Update the posterior distribution,

$$\forall j, \forall t, \quad p_\pi(z(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}) \propto g(x^{(i)}(f_{y^{(i)}}(t)); \mu(j,i), \sigma^2(j,i)) \pi(j, f_{y^{(i)}}(t)),$$

- M-step: Update the estimation of the template,

$$\forall j, \forall t, \quad \pi(j,t) \propto \sum_i |J_{f_{y^{(i)}}}(t)| p_\pi(z(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}).$$

5.7 Experiments

In the following experiments we present some detection results first on the 2D-SCC database which contains 2D sagittal slices extracted from 47 different individuals. The position of SCC1 and SCC2 is given by an expert. We use 30 images for training and 17 images for testing. (We use the same testing images as in Chapters 3 and 4). We also present some results on the 3D-SCC for the detection of SCC1. Since the T-DIM models the intensity distribution of each image as a nuisance parameters, it is not necessary to normalize the image intensity as we did before using DIM in Chapter 3. Figure 5.3 presents some instances of testing images together with their respective histograms of intensities to emphasize the intensity differences.

We use the same deformation model as before, i.e. a Gaussian spline with σ fixed

Algorithm 5.8 Decoupled Model: Prediction

TESTING

Let x be a testing image of unknown photometric parameters $\theta = (\mu(j), \sigma^2(j), \alpha(j))$, $1 \leq j \leq J$,

Photometry Estimation: Initialize θ and iterate until convergence,

- **E-step:** Update the posterior distribution,

$$\forall j, \forall s \quad p(z(s) = j | x(s)) \propto g(x(s); \mu(j), \sigma^2(j)) \alpha(j),$$

- **M-step:**
  - Update the proportions,

$$\forall i, \forall j, \quad \alpha(j) \propto \sum_s p_\theta(z(s) = j | x(s))$$

  - Update the photometric parameters,

$$\forall j, \quad \mu(j) \leftarrow \frac{\sum_s x(s) p_\theta(z(s) = j | x(s))}{\sum_s p_\theta(z(s) = j | x(s))},$$
$$\forall i, \forall j, \quad \sigma^2(j) \leftarrow \frac{\sum_s (x - \mu(j))^2(s) p_\theta(z(s) = j | x(s))}{\sum_s p_\theta(z(s) = j | x(s))}.$$

Landmark Detection: Initialize $y \leftarrow \bar{y}$ and iterate until convergence

- **Compute** the gradient direction $\frac{\partial \ell}{\partial y}(x, y)$ from (5.56).
- **Update** the location of the landmarks,

$$y \leftarrow y + a \cdot \frac{\partial \ell(x, y)}{\partial y}, \quad \text{with} \quad a \leftarrow \arg\max_{a \in \mathbb{R}^+} \ell \left(x, y + a \frac{\partial \ell(x, y)}{\partial y} \right),$$

manually. We will present results for different values of σ ranging between 3 and 15 pixels.

We also need to chose a number of tissue types to be used in the probabilistic deformable template. The brain is usually modeled by 3 major tissues: the Cerebro-Spinal Fluid (CSF), the Gray Matter (GM) and the White Matter (WM). The number of tissues will vary depending whether one wants to model separately mixed pixels, or if one works on a subset of the whole image in which few of the tissue types are actually present. We will therefore vary the number of tissue types from 2 to 5 depending on the experiment.

5.7.1 Template Estimation

In the following experiments, we use the estimation and testing algorithm described in Algorithm 5.6. Both the estimation of the model parameters and the prediction of the landmark locations are obtained by an EM algorithm, which may fall in local maxima of the likelihood function. We will compare the joint algorithm and the decoupled algorithm in terms of likelihood maximization and in terms of performance for the detection of SCC1.

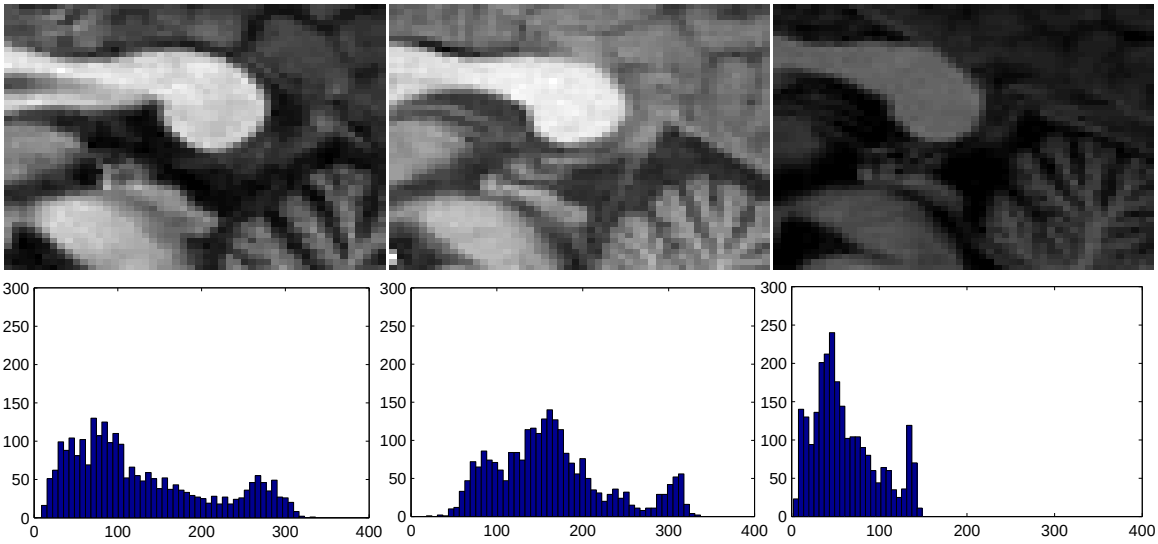


Figure 5.3: **Top**: 3 testing images of 2D-SCC. Each image represents a region of the sagittal view of the plane containing the corpus callosum. **Bottom**: Intensity histograms of the corresponding grayscale images.

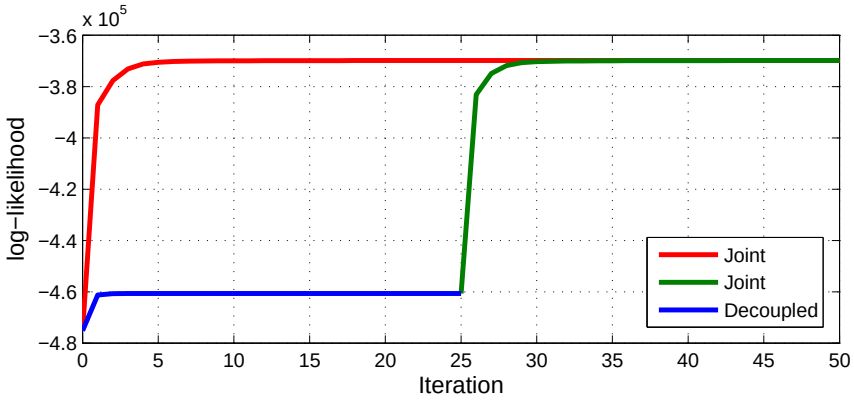


Figure 5.4: Evolution of the likelihood function during learning. The RED curve represents the evolution of the likelihood by joint optimization. The BLUE curve represents the likelihood evolution when using the decoupled algorithm and finally the GREEN curve represents the evolution of the likelihood when using the joint algorithm, initializing with the template estimate given by the Decoupled algorithm.

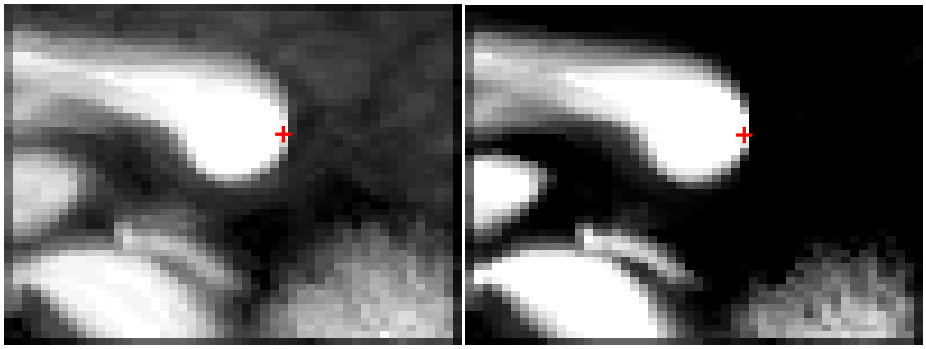


Figure 5.5: Estimated Templates in the case of T2-DIM (2 tissue types). We represent the probability at each pixel to observe the brighter tissue. White represents a probability close or equal to 1 and Black represents a probability close or equal to 0. The different shades of gray represent intermediate probabilities. The red cross shows the location of the landmark SCC1. **Left:** Template estimated by the decoupled algorithm, **Right:** Template estimated by the joint algorithm.

Figure 5.4 presents the evolution of the likelihood of the training set composed of 30 images of 2D-SCC during learning for a model with 2 tissue types. The deformation is modeled by a Gaussian spline of standard deviation $\sigma = 10$. The template is initialized by a Uniform distribution at each pixel. The initialization of the photometric parameters is obtained by estimating a set of Gaussian parameters on each image independently. We compare the likelihood evolution when using the joint optimization as described in Algorithm 5.6 and the decoupled algorithm 5.7. In only few iterations both the joint algorithm and the decoupled optimization converge, except that the decoupled optimization seems to be in a local maximum of the likelihood. If after convergence of the decoupled algorithm, ones uses the joint optimization, initialized at the current estimate, the likelihood exits the local maximum and reaches that same maximum as with the joint algorithm. The initialization of this experiment depends on the photometric parameters of each images, themselves obtained by the EM algorithm. The observed behavior was reproduced in 10 independent experiments. Figure 5.5 illustrates the template obtained by the decoupled and joint experiment at convergence. The template estimated by joint optimization is sharper than the one obtained by decoupled optimization. For example in the top right part of the template there is a region in which there exists a mixed probability to observe a dark or a bright tissue type. By coupling the estimation of the template and of the photometric parameters, the latter are more precisely adjusted using the current estimate of the template as prior information.

5.7.2 Detection Performance

We now present the performance of the detection algorithm on SCC1 and SCC2. We compare the initial localization error of the landmarks, the distance between $\bar{y}$ the position in the template and the expert location, with the prediction error of the detection algorithm, defined as the Euclidean distance between the predicted landmark and the ground-truth as defined by an expert. To assess the advantage of the joint optimization compared to the decoupled optimization, we performed 4 experiments using either the joint algorithm or

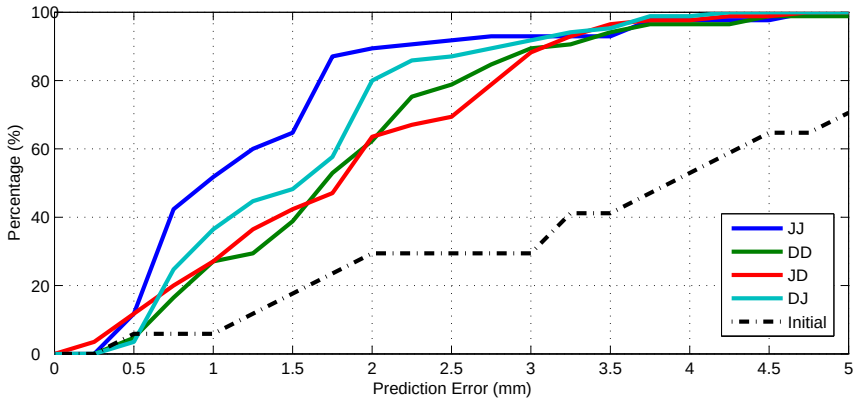


Figure 5.6: Repartition of the prediction error on the set of 17 testing images (5 estimates per images). We compare 4 algorithms composed of a learning and testing phases, joint J or decoupled D, to the initial repartition of the landmark localization error.

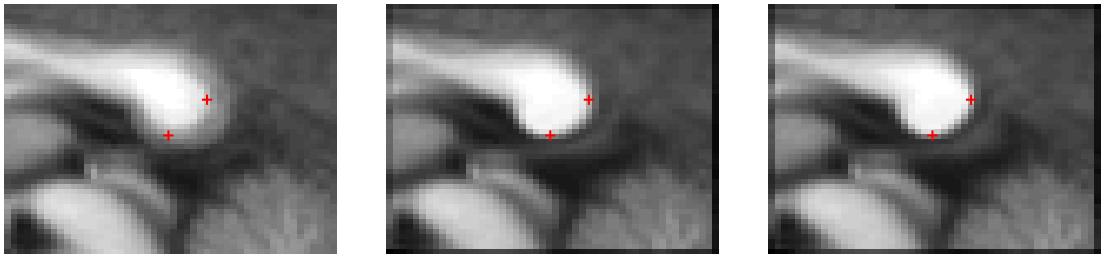
Alg.	Performance (mm)	Statistical Significance			
		JJ	DD	JD	DJ
JJ	1.23 (0.91)	∅			
DD	1.80 (0.84)	<**0.0001**	∅		
JD	1.79 (1.06)	**0.0001**	0.9466	∅	
DJ	1.55 (0.84)	**0.0007**	0.1225	0.1776	∅
Initial	3.62 (1.80)	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**

Table 5.1: Prediction performance for each algorithm. p-value associated to the Wilcoxon test comparing the average of the algorithm results.

the decoupled algorithm for training and testing. We denote JJ the experiment in which we use the joint algorithm both for training and testing, similarly DD for the decoupled algorithm. The experiment with joint learning but decoupled testing is denoted by JD, while DJ denotes the opposite,i.e. decoupled learning but joint detection. To deal with the effect of the random initialization of both the learning and testing algorithm each experiment is performed 5 times on each of the 17 testing images.

First, we quantify the effect of the random initialization on the prediction performance. We assume that the random effect is a additive Gaussian noise, we denote by e_i the prediction error of an image and by $\bar{e}_i$ the average error for that image. Using 85 samples (5×17), the estimated distribution of $E = e - \bar{e}$ has 0 mean and 0.3mm standard deviation. In conclusion, depending on the initialization, we obtained a variation of 0.3mm, which sounds reasonable given that the image resolution is 1mm.

Figure 5.6 presents the repartition of the prediction error for the experiments JJ, DD, JD and DJ. All these methods improve the localization of the landmarks, but this is the joint method that achieves the best performance with 50% of the landmarks detected with less than 1mm of error. Table 5.1 confirms these observations and shows that there exists a statistically significant difference between JJ and the other algorithms (using a Wilcoxon test). The average error of JJ is a bit larger than a pixel. Given that that the pixel resolution is 1mm, this is a satisfactory precision.



(a) Before Registration (b) Automatic Registration (c) Expert Registration

Figure 5.7: Testing Image Registration. Each subfigure represent the pixel-by-pixel intensity average of the 17 testing images. The red crosses represent the landmark locations $\bar{y}$. Subfigure (a) is computed before detecting the landmarks, i.e. the images have only been globally aligned to Talairach's atlas. Before computing the average image depicted in Subfigures (b) and (c), the images were registered to the template based on the landmark correspondences, using a Gaussian spline deformation ($\sigma = 7$). In (b) the correspondences are set using the automatic landmarks while in (c) we use the manual landmarks.

Figure 5.7 represents the "average" images obtained before registration, when the registration is performed using the automatic landmarks and when the registration is based on the landmarks located manually. We use the same model for registration as for the prediction, i.e. the Gaussian spline deformation with $\sigma = 7$. If the images are well registered the corresponding structures should coincides and therefore the average image should be sharp. When the image are misaligned, a blur appears in the image. The average image around the landmark is much sharper after registration, and there is little differences between the average image obtained using the automatic landmarks or the manual landmarks. It shows that the precision of the detection is adequate for registering images based on automatic landmarks.

5.7.3 Combining Registration and Segmentation

Although the main purpose of T-DIM in our application is to locate landmarks by learning and locating characteristic pattern in the image, the algorithm also provides us with a segmentation and a local registration of the image. The image segmentation is obtained by assigning each pixel to the tissue with the highest probability. The template acts as a prior information on the tissue type. Locating the landmarks in a new image is equivalent to finding the locally best deformation from the template to the image. The algorithm can be seen as a combined method to perform registration and segmentation. For a new image, the algorithm is initialized with the photometric parameters estimated by a simple EM without spatial prior. During the joint optimization, the grid of the template is deformed so that the tip of the corpus callosum be well segmented. Simultaneously the template provides a prior information for segmentation, which modifies the photometric parameter estimates. The segmentation of the corpus callosum is in practice an easy task because the intensity differs from the rest of the image. The interest of the algorithm is to provide simultaneously a segmentation and a registration, which is not the case of the simple EM. Figure 5.8 compares the segmentation before and after optimization on three

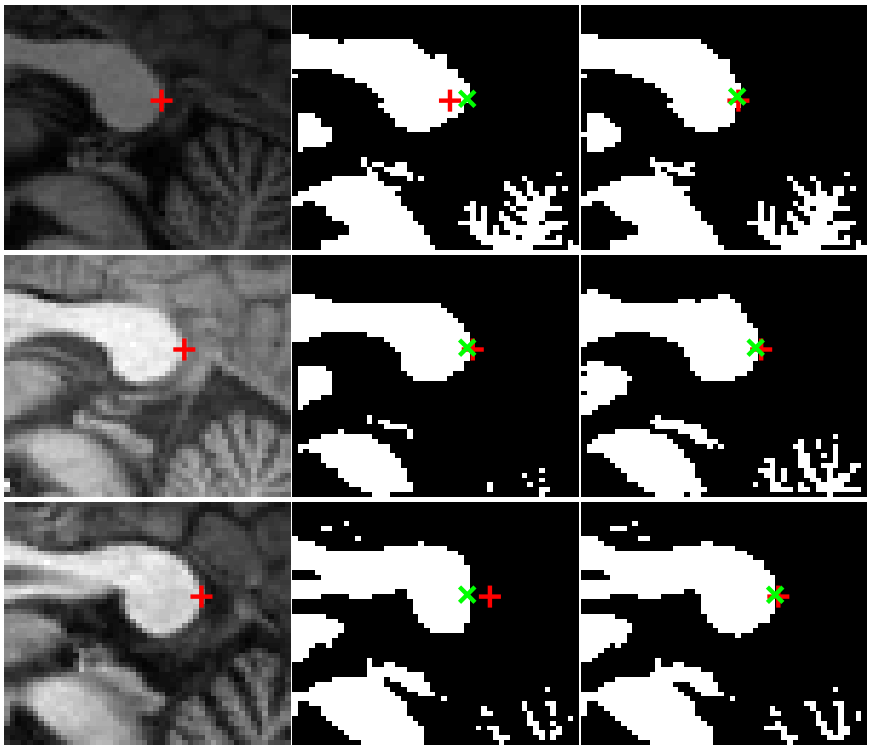


Figure 5.8: Combining Registration and Segmentation. Each line represents an image of the training set. The left most image depicts the original grayscale image and the position of the landmarks given by the expert. The middle column corresponds to the segmentation obtained when using the learnt template as a spatial prior and the photometric parameters used for the initialization of the joint algorithm. The registering deformation used to combine the template and the image is the identity. The red cross represents the expert location and the green cross the tentative location of the landmarks. In the rightmost column, the segmentation is obtained using the estimated deformation to register the template to the image, and using the optimized photometric parameters. The changes are mostly noticeable in the region of the landmark. The green cross represents the predicted location of the landmark, the red cross shows the location marked by the expert.

images from the testing set.

5.7.4 Choice of the Parameters

The T-DIM model requires to set by hand two parameters: J the number of tissue types and σ the standard deviation of the Gaussian kernel used to model the image deformation. By increasing the number of tissue types, on one hand it is expected that the precision of the model and maybe the performance increase, but on the other hand the number of parameters increases. The size of the Gaussian kernel standard deviation is related to the spread of the deformation. If σ is small then the deformation is local (potentially not invertible) and in consequence all pixels at further distance from the landmarks are not affected by the deformation, they are thus not contributing to the likelihood variations. If the standard deviation increases, more pixels are subject to the deformation. It increases the size of the tissue pattern used for detection. It is therefore expected that when the local

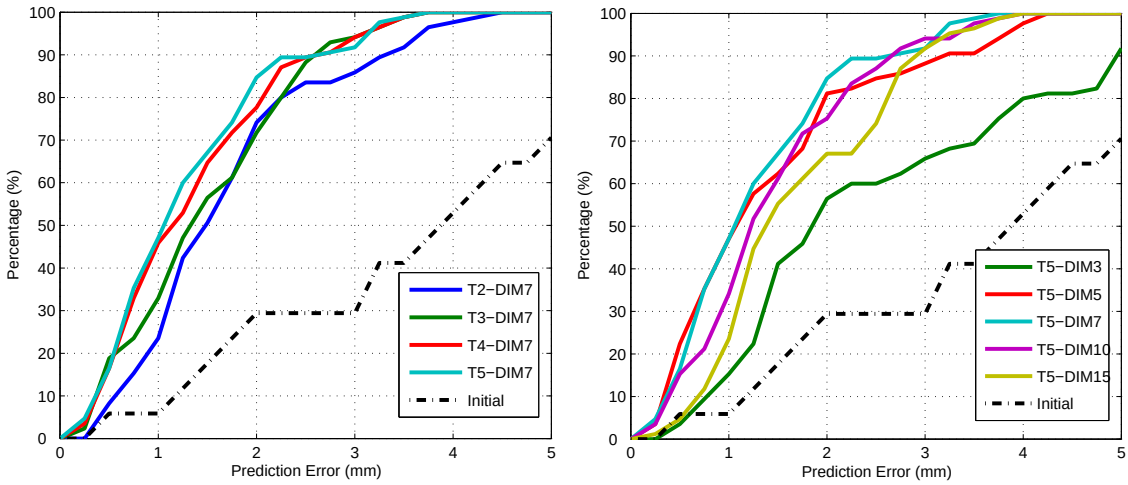
pattern around a landmark is not discriminative enough, the specificity of the algorithm will increase with the size of the pattern. We already observed this phenomenon in the experiments of Chapter 3.

In order to test the effect of J and σ on the performance, we test the algorithm on the detection of SCC1 and SCC2, with J varying between 2 and 5 and with σ varying between 3 and 15 pixels. Similarly to the preceding experiments, the detection is performed 5 times for each images with random initialization. The lowest error for SCC1 is 1.26 mm (0.85 mm) with $J = 5, \sigma = 7$ and for SCC2, 1.04 mm (0.58 mm) with $J = 5, \sigma = 5$. These numerical results are comparable to the performance obtained with DIM, cf. Table 3.2. Recall that T-DIM contrarily to to DIM, does not require any preprocessing of the image such as intensity normalization. Figure 5.9(a) represents the repartition of the prediction error for different values of the parameters in the case of SCC1. Similar results were obtained for SCC2. We conclude from this experiment that in the case of SCC, the precision increases when the number of tissues in the model increases. For the 2D-SCC database, the best results were achieved for σ is between 5 and 7. The optimal choice of the kernel is related to the amount of information contained around the landmark, but also on the specificity of the pattern learnt. If the landmark lives in a rich region of the image, we can predict that a small kernel will be enough but if the local intensity pattern is less distinctive, a larger kernel will be needed to achieve comparable performance.

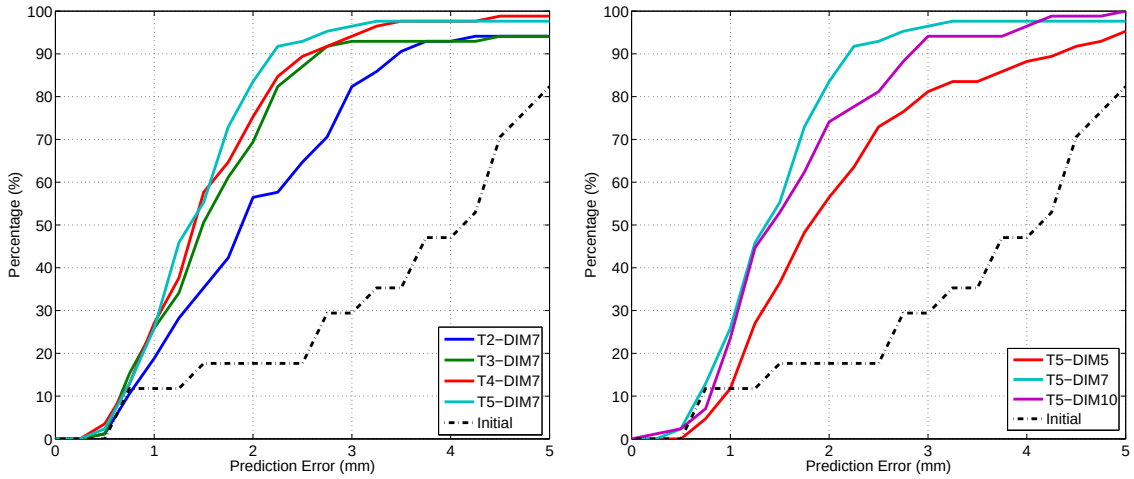
We repeated the experience on 3D-SCC for the detection of SCC1. (Since SCC2 is defined in 2D only, we did not use it in this experiment). The number of tissues varies from 2 to 5 and the Gaussian kernel parameter from 5 to 10. The experiment is repeated 5 times on each images of the training set. In order to reduce the computational load, in this experiment we compute the likelihood variations using a neighborhood of the landmark of diameter equal to σ. The best performance were achieved for $J = 5$ and $\sigma = 7$. The prediction error is in average 1.48 mm with a standard deviation of 0.82 mm. Before detection the localization error was 3.66 mm (1.69 mm). Figure 5.9(b) represents the error repartition.

5.8 Chapter Conclusion

Even though the T-DIM model is more complicated than the precedent models due to the presence of hidden variables and nuisance parameters, it is still possible to derive an intuitive algorithm to perform landmark detection and more generally for medical image analysis. All the optimization methods are directly derived from the modeling assumption. The algorithm depends only on two parameters, the choice of the kernel and the number of tissue types. Contrarily to DIM, this model can handle intensity variations and even the simultaneous analysis of images acquired with different protocols or image modalities. Furthermore the model allows us to perform a joint segmentation-registration. Notice that we did not provide any manually segmented image to the system, but only few points correspondences.



(a) Repartition of the prediction error of SCC1 in 2D



(b) Repartition of the prediction error of SCC1 in 3D

Figure 5.9: We use the notation T5-DIM7 for example to refers to the T-DIM algorithm with $J = 5$ and $\sigma = 7$. *Initial* in all the graphs represents the repartition of the error before detecting the landmarks. **Left:** Error repartition when the number of tissues varies. **Right:** Error repartition when the standard deviation of the kernel varies.

CHAPTER 6

DEFORMABLE OBJECT FOR MEDICAL IMAGING

In order to perform non-rigid registration of images, it is commonly assumed that the whole image is subject to a deformation. In the previous deformable models, we have made the same assumption and modeled an image as the result of a random deformable template of an image or a segmentation. Since we are specifically interested in locating landmarks, the deformation model was chosen with a local support only. While it allowed us to reduce the computational load, it does not handle well cases in which part of the deformation is affine. For example if a whole structure is translated, one needs to increase the support of the deformation in order to displace with minimal distortion the structure on which only few landmarks are located. In this chapter we propose a different approach to medical image registration and model an image as the superimposition of a deformable object and a background image. In this model the deformation acts on the object but not on the background image. In this chapter we first look at the specificities of this approach. Then we demonstrate how the deformable object model (DO) can be coupled to either the deformable intensity model (DIM) or the tissue-based deformable intensity model (T-DIM) to perform landmark detection in brain MRI.

6.1 Deformable Object Model

The modeling principle used in the Deformable Object approach is fundamentally different from usual techniques in medical imaging. As presented in Chapter 2, usually the image is modeled as the result of a deformation of a template, while the Deformable Object approach models an image as a deformable object on top of a background image. The main advantage of this formulation is that it is possible to handle both rigid and non-rigid deformations while keeping the cost function computation to a finite domain of the image, rather than introducing distortions at the limit of the deformable domain or approximating arbitrarily the optimization function. This approach is somewhat similar to the idea used for face tracking in video sequences. The face is a deformable object while the background remains unchanged even though the object is translated in front of the background. While it seems at first crude to make such a modeling assumption for medical imaging, this model is quite useful in this field of applications too. Indeed, it is often the case that the image is first globally registered and then that a non-rigid deformation model is used for refine the registration result in a region of interest. This is the case with brain images, that are first aligned for example to the Talairach grid and then locally registered one onto another using non-rigid deformations. The local deformation should not deteriorate the global alignment, even though locally the structure might have been translated.

6.1.1 Model Description

We denote x the image which maps each point of a finite lattice $\Lambda \subset \mathbb{R}^d$ to an intensity value in $\mathbb{R}$. Let y be the location of K landmarks, i.e. a vector of $\mathbb{R}^{dK}$. Because the images are observed on a finite grid of $\mathbb{R}^d$, we use linear interpolation to define the image intensity value on a bounded domain of $\mathbb{R}^d$ containing Λ, that we denote by Ω. In addition we define Λ_T and Ω_T respectively the finite lattice and subspace of $\mathbb{R}^d$ on which is defined the template. We define $\bar{y}$ the landmark reference location in Λ_T. We denote by f_y a bijection from Λ_T to Λ satisfying the landmark matching condition, i.e. such that $\bar{y}$ is mapped onto y. We define $(\Lambda_T^o, \Lambda_T^b)$ a partition of Λ_T. We denote by $\Lambda^o(y) = f_y(\Lambda_T^o)$ the image of the object domain by the transformation f_y and $\Lambda^b(y) = f_y(\Lambda_T^b)$ the image of the background. $(\Lambda^o(y), \Lambda^b(y))$ is an object-background partition of Λ. We propose to build a statistical model of the joint distribution of the image intensity x and the landmark location y. In the deformable object model it is assumed that the distribution of the image intensity given the location of the landmarks, follows a different law depending whether the pixel belongs to the object or to the background:

$$\forall s \in \Lambda, \quad p(x_s|y) = \begin{cases} p_o(x_s), \text{ if } f_y^{-1}(s) \in \Lambda^o, \\ p_b(x_s), \text{ if } f_y^{-1}(s) \in \Lambda^b. \end{cases}$$

The log-likelihood of an image is a sum of two terms:

$$\begin{aligned} \ell(x, y) &= \ln p(y) + \sum_{s \in \Lambda} \ln p(x_s|y) \\ &= \ln p(y) + \sum_{s \in \Lambda^o(y)} \ln p_o(x_s|y) + \sum_{s \in \Lambda^b(y)} \ln p_b(x_s|y). \end{aligned} \tag{6.1}$$

We assume that the background intensity model does not depend on the location of the landmarks. In addition, by adding and subtracting the sum $\sum_{s \in \Lambda^o(y)} \ln p_b(x(s))$, the likelihood becomes:

$$\ell(x, y) = \ln p(y) + \sum_{s \in \Lambda^o(y)} \ln \frac{p_o(x_s|y)}{p_b(x_s)} + \sum_{s \in \Lambda} \ln p_b(x_s). \tag{6.2}$$

The sum of the background probability at each pixel of the image, does not depend on the landmark location y, therefore the third term of (6.2) can be ignored in the maximization of the likelihood with respect to y. It reduces the computation to the finite domain $\Lambda^o(y)$ without any assumption needed to be made on the support of f_y.

Sampling images from the DO model consists first in drawing from $p(y)$ a location for the landmarks. At each pixel of the image grid, an intensity value is sampled from p_o or p_b depending whether the pixel belongs to the deformed object $\Lambda^o(y)$ or to the background $\Lambda^b(y)$.

6.1.2 Choice of the Deformation

Because the deformable object only is subject to the effect of the deformation, it is possible to work with deformations with large support. In order to keep parameterizing the

deformation by the landmark displacements, we still use a spline-based based deformation model but now it is possible to use either kernel with local support or incorporating an affine transformation. We presented some examples of such kernels in Chapter 1: the Affine-Gaussian kernel and the Thin-Plate Spline. The deformation is parameterized by the vectors of coefficients $\beta \in \mathbb{R}^{dK}$. Denoting by $\bar{y}$ and y respectively the location of the landmarks in the template and in the image, we define $f_y : \mathbb{R}^d \to \mathbb{R}^d$ the spatial deformation of the form:

$$\forall t \in \Lambda_T, \quad f_y(t) = \sum_{k=1}^{K} \beta_k \kappa(t, \bar{y}_k), \text{ such that } f_y(\bar{y}) = y. \tag{6.3}$$

The choice of the kernel determines the nature of the deformation and its support. The algorithm though stays unchanged whatever choice of kernel is made.

6.1.3 A Toy Example

We first describe a toy example on which we compare the performance of several methods for finding correspondences between images. We build random images, composed of a background and a 5-by-5 square at a random location, indexed by $y \in \Lambda$. We denote $\Lambda^o(y)$ the set of pixels belonging to the square when it is located in y and symmetrically we denote by $\Lambda^b(y)$ the set of pixels belonging to the background of the image. The intensity distribution at a pixel depends on the location of the landmarks:

$$\forall s \in \Lambda, \quad p(x(s)|y) = \begin{cases} p_o(x(s)) = g(x(s); 1, \tau), \text{ if } s \in \Lambda^o(y), \\ p_b(x(s)) = g(x(s); 0, \tau), \text{ if } s \in \Lambda^b(y). \end{cases}$$

where g denotes the Gaussian density function. Each image is obtained by sampling from a Uniform distribution on Λ a location y. The 25 pixels belonging to $\Lambda^o(y)$ are sampled from p_o, while the rest of the pixels are sampled from p_b with τ fixed. Figure 6.1 shows two instances of the simulated images. When $\tau = 0.5$, the 5-by-5 square is easy to locate in the image but when $\sigma = 1$ the square does not seem to be distinguishable from the background.

The task is to detect the location of the square in the image given the distribution of the intensity inside and outside of the square. The deformation model in this example is a 2D translation of the square. We compare the performance of the three following prediction methods, based on the maximization of the cost function $\ell(x|y)$.

1. **Intensity Matching by Sum of Squared Differences** (SSD) (or using the DIM introduced in Chapter 3): A template is defined on a grid Λ'_T and is composed of a central 5-by-5 square of ones (mean of the object Gaussian) and padded with zeros. Therefore, $\Lambda^o_T \subset \Lambda'_T$. At each tentative location $y \in \Lambda$, the neighboring image intensity is compared to the template using the SSD. We denote by $\Lambda^o(y)$ the set of pixels belonging to the object when it lies in y and by $f_y(\Lambda'_T) \setminus \Lambda^o(y)$ the set of pixels around the object that are assumed to come from the background model. The likelihood is a sum over two terms, coming from the object and the background:

$$\ell_{\text{SSD}}(x|y) = -\left(\sum_{s \in \Lambda^o(y)} (x(s) - 1)^2 + \sum_{s \in f_y(\Lambda'_T) \setminus \Lambda^o(y)} (x(s))^2 \right). \tag{6.4}$$

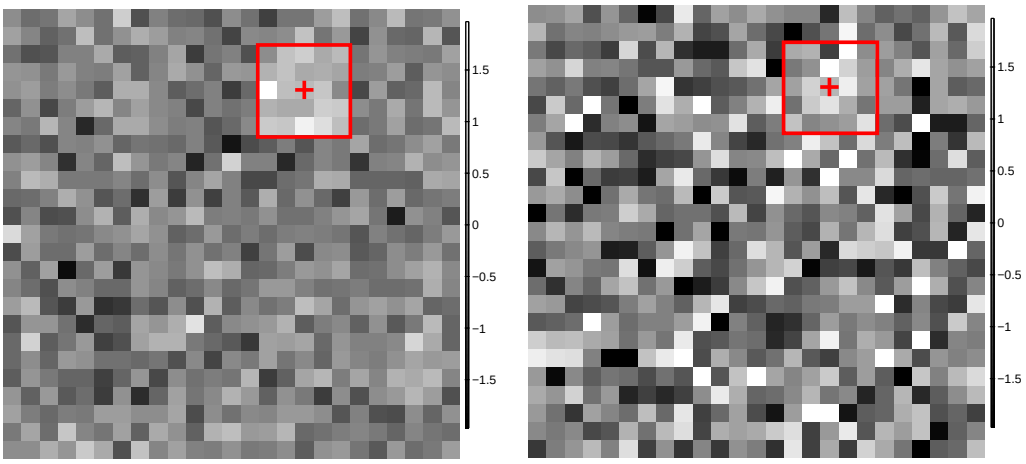


Figure 6.1: Examples of random images. **Left:** Noise level is $\tau = 0.5$. **Right:** Noise level is $\tau = 1$

2. **Deformable Object**: We use the likelihood function defined in (6.2). The background intensity distribution follows a Gaussian distribution with zero mean and τ^2 variance. The deformable object intensity distribution is also a Gaussian distribution of mean 1 and variance τ^2. Therefore, the Deformable Object (DO) likelihood function is, up to a constant,

$$\ell_{\mathrm{DO}}(x|y) = -\sum_{\Lambda^o(y)} \left(x(s) - \frac{1}{2} \right). \tag{6.5}$$

3. **Template Matching by Normalized Sum of Squared Differences** (NSSD): it consists in computing the correlation between a patch of the image and the template. We use the same template as for SSD, defined on Λ'_T, and denote by $\Lambda'(y)$ the image of Λ'_T by f_y. The correlation is written as:

$$\ell_{\mathrm{NSSD}}(x|y) = \sum_{s \in \Lambda'(y)} \frac{(x(s) - \bar{x}_{\Lambda'(y)})(tp(s) - \bar{tp})}{\gamma(x, \Lambda'(y))\gamma(tp)},$$

where $\bar{x}_{\Lambda'(y)}$ is the mean intensity in the window $\Lambda'(y)$, $tp(s)$ the template value at s (0 or 1 in our toy example), $\bar{tp}$ the mean value of the template (25/49 for the toy example), $\gamma(x, \Lambda'(y))$ and $\gamma(tp)$ the standard deviation of respectively the intensity in the image patch $\Lambda'(y)$ and of the template.

Since we are working with translation, the cost function is computed for each possible value of y and $\hat{y} = \arg\max_y \ell(x|y)$. In order to avoid border effects, we actually build larger images and allow y to live in a subset of the image.

We compare the performance of the 3 methods on an experiment with 1000 random images. y lives in a central square of size 25×25, for images of size 45×45. We perform experiments with $|\Lambda'_T| = 7 \times 7$ and $|\Lambda'_T| = 11 \times 11$, while $|\Lambda^o_T| = 25$ for all the experiments. The experiment is repeated for different levels of noise: $\tau = 0.5, 0.75, 1$. Table 6.1

Method		$\tau = 0.50$	$\tau = 0.75$	$\tau = 1.00$
	$\boldsymbol{d}$	0.01	0.08	**0.52**
DO	std	0.09	0.28	1.97
5×5	p($\boldsymbol{d}$=0)	~ 1.00	0.99	0.92
	$\boldsymbol{d}$	0.03	4.72	**10.34**
SSD	std	0.61	7.17	7.71
7×7	p($\boldsymbol{d}$=0)	$\sim$1.00	0.62	0.22
	$\boldsymbol{d}$	0.01	1.11	4.69
NSSD	std	0.10	3.92	7.24
7×7	p($\boldsymbol{d}$=0)	$\sim$1.00	0.91	0.63
	$\boldsymbol{d}$	0.01	0.11	**0.97**
NSSD	std	0.09	0.49	3.24
11×11	p($\boldsymbol{d}$=0)	$\sim$1.00	0.99	0.88

Table 6.1: Results of the detection experiments on 1000 random images with variable amount of noise. All measurements are in number of pixels

contains the mean Euclidean distance (**d**) between $\hat{y}$ and the real location of y, the standard deviation of **d**, and the proportion of images for which the prediction is exact, i.e **d**=0.

For a moderate amount of noise ($\tau = 0.50$), the 4 predictors reach the same level of accuracy and predict correctly the location of the square almost in all the images. As the amount of noise increases, the performance of SSD deteriorates the most, followed by NSSD(7). The robustness of NSSD increases if the support of computation increases too. We have not observe that behavior with SSD (result not shown in the table). NSSD(11) is very robust even for large amount of noise. However, DO outperforms all the other predictors even when the amount of noise is noticeably large. It is also the one that relies on the smallest region for computation.

While DO and SSD have very different behaviors they are in reality closely related. While DO is the full likelihood function written such that its computation depends only on what happens at the location of the object, SSD is a truncated likelihood. Instead of defining the likelihood of the whole image, it is the likelihood of a patch in the image. Let us compare the cost function associated to these two models:

$$\begin{aligned}
\ell_{\text{DO}}(x|y) &= \sum_{s\in\Lambda^o(y)} \ln p_o(x(s)|y) + \sum_{s\in\Lambda^b(y)} \ln p_b(x(s)), \\
&= \sum_{s\in\Lambda^o(y)} \ln \frac{p_o(x(s)|y)}{p_b(x(s))} + \sum_{s\in\Lambda} \ln p_b(x(s)), \qquad (6.6) \\
\ell_{\text{SSD}}(x|y) &= \sum_{s\in\Lambda^o(y)} \ln p_o(x(s)|y) + \sum_{s\in\Lambda'(y)\setminus\Lambda^o(y)} \ln p_b(x(s)), \\
&= \sum_{s\in\Lambda^o(y)} \ln \frac{p_o(x(s)|y)}{p_b(x(s))} + \sum_{s\in\Lambda'(y)} \ln p_b(x(s)). \qquad (6.7)
\end{aligned}$$

Therefore the SSD and DO likelihood functions differ only in their second term. While the second term of (6.6) does not depend on the location of y, the second term of (6.7) depends on y and needs to be computed for each tentative location of y.

SSD is an approximation of the image likelihood which is commonly used to reduce the region of computation in cases where the deformation has infinite support such as

Thin Plate Spline (TPS) deformation [8, 69] or when the transformation is affine. Usually the template is chosen so that it covers the interesting pattern, but not too large to reduce computation. The above experiment shows that in case of significant amount of noise, it may bring the template to match to a wrong position, while comparing the foreground and background likelihood is in our experiment more robust to noise.

6.2 Deformable Intensity Object

The Deformable Intensity Object is a model where the object and the background are described by their intensity distribution at each pixel. The model is based on similar ideas as the ones introduced in Chapter 3.

6.2.1 Image Likelihood

We denote f_y the spatial transformation of $\mathbb{R}^d$ such that $f_y^{-1}(\bar{y}) = y$ and $f_{\bar{y}}$, the spatial transformation such that $f_{\bar{y}}^{-1}(\bar{y}) = \bar{y}$, i.e. the identity [1]. To simplify the notation we introduce $t = f_y^{-1}(s)$ and $t_b = f_{\bar{y}}^{-1}(s)$. The intensity distribution at each pixel is assumed independent from the neighboring pixels when the location of the landmarks is given and is modeled as follows:

$$\forall s \in \Lambda, \quad p(x(s)|y) \sim \begin{cases} g(x(s); x_o(t), \tau_o(t)), & \text{if } t \in \Lambda_T^o, \\ g(x(s); x_b(t_b), \tau_b(t_b)), & \text{if } t \in \Lambda_T^b. \end{cases} \tag{6.8}$$

The model parameters are composed of the deformable object template x_o, τ_o defined for all $t \in \Lambda_T^o$ and the background template x_b, τ_b defined for all $t \in \Lambda_T^b$. The joint log-likelihood of an image and a set of landmarks is:

$$\ell(x, y) = \ln p(y) + \sum_{s \in \Lambda^o(y)} \ln g(x(s); x_o(t), \tau_o(t)) + \sum_{s \in \Lambda^b(y)} \ln g(x(s); x_b(t_b), \tau_b(t_b)) \tag{6.9}$$

As we did before the log-likelihood can be rearranged:

$$\begin{aligned} \ell(x, y) &= \ln p(y) + \sum_{s \in \Lambda^o(y)} \ln \frac{g(x(s); x_o(t), \tau_o(t))}{g(x(s); x_b(t_b), \tau_b(t_b))} + \sum_{s \in \Lambda} \ln g(x(s); x_b(t_b), \tau_b(t_b)), \\ &\propto \ln p(y) + \sum_{s \in \Lambda^o(y)} \ln \frac{\tau_b(t_b)}{\tau_o(t)} + \sum_{s \in \Lambda^o(y)} \left[\frac{(x(s) - x_b(t_b))^2}{2\tau_b^2(t_b)} - \frac{(x(s) - x_o(t))^2}{2\tau_o^2(t)} \right]. \end{aligned} \tag{6.10}$$

If for the tentative location y,the intensity at pixel $s \in \Lambda^o(y)$ is more likely to comes from the background model than from the object model, i.e. $\frac{|x(s)-x_b(t_b)|^2}{\tau_b^2(t_b)} > \frac{|x(s)-x_o(t)|^2}{\tau_o^2(t)}$, the likelihood decreases, but if the pixel indeed belongs to the object, the likelihood function increases. The sum over the whole image of the probability for the pixel to belong to the background does not depend on the location of the landmarks and can therefore be disregarded for the maximization of the likelihood function. If at $s \in \Lambda^o(y)$, the background

[1] Even though we introduce here a quite heavy notation for the identity, it is helpful for the following computation to write it this way.

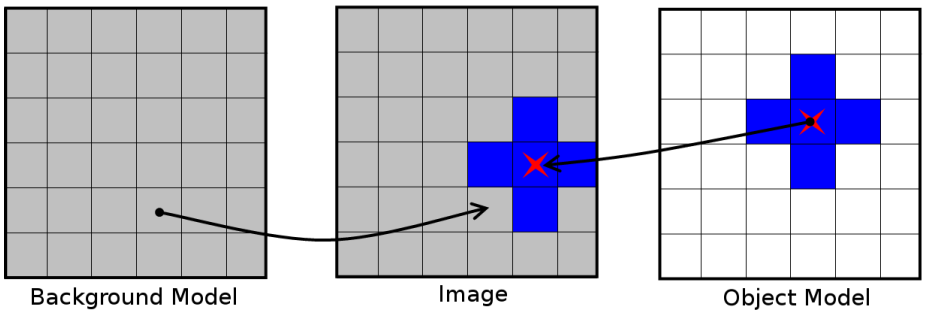


Figure 6.2: Deformable Object Model. The left square represents the background model (x_b, τ_b) and the right square represents the deformable object template (x_o, τ_o). The central square represents the image resulting from the superimposition of the deformed intensity model (here translated) on top of the background image. The arrows points represents the mapping from the background template $f_{\bar{y}}$ to the image and from the deformable object to the image f_y. The landmark location is represented by the red cross.

model $x_b(f_{\bar{y}}^{-1}(s)), \tau_b(f_{\bar{y}}^{-1}(s))$ and the object model $x_o(f_y^{-1}(s)), \tau_o(f_y^{-1}(s))$ are identical, the pixel is neutral and its contribution to the log-likelihood is 0. Notice though that the object model associated to s depends on the position of y, i.e. the deformation of the object. Therefore a pixel which contains no information for a given value of y may contain some information for another value of y.

6.2.2 Model Estimation

The model is estimated from a set of training images, in which the landmarks have been located manually. The correspondences between images are given and the intensity values observed. The parameters to be estimated are: x_o, τ_o for the object model and x_b, τ_b for the background model. The training images are independent samples of the joint distribution $p(x, y)$, therefore using (6.9), the likelihood of the training sample (x_1^N, y_1^N) is, up to a constant:

$$\ell(x_1^N, y_1^N) = \sum_{i=1}^{N} \ln p(y) + \sum_{i=1}^{N} \sum_{s \in \Lambda^o(y^{(i)})} \ln g(x^{(i)}(s); x_o(t^{(i)}), \tau_o(t^{(i)})) + \sum_{i=1}^{N} \sum_{s \in \Lambda^b(y^{(i)})} \ln g(x^{(i)}(s); x_b(t_b), \tau_b(t_b)), \quad (6.11)$$

with $t^{(i)} = f_{y^{(i)}}^{-1}(s)$ and $t_b = f_{\bar{y}}^{-1}(s)$.

Figure 6.2 illustrates the Deformable Object Model on a simple example. The arrows represents for a pixel s in the image the corresponding locations t and t_b in the object and background model. Notice the difference of position between t and t_b. In the case of the object model, t is located with respect to $\bar{y}$ such that the relative positions of y and s are preserved, but in the background model, t_b corresponds to the absolute location of the pixel in the image. Intuitively the intensity at a pixel s is compared to what should be observed if the pixel were belonging to the deformed object or to the background.

The maximization of the training set likelihood leads to closed form estimates of the model parameters. The object and background model are estimated independently, using the training images. One needs to fix Λ_T^o beforehand. We start with the estimation of the object model:

$$\ell_1(x_1^N|y_1^N) = \sum_{i=1}^{N} \sum_{s\in\Lambda_i^o(y^{(i)})} \ln g(x^{(i)}(s); x_o(t^{(i)}), \tau_o(t^{(i)})), \tag{6.12}$$

with $\forall i, t^{(i)} = f_{y^{(i)}}^{-1}(s)$ and $\Lambda_i^o(y^{(i)})$ the object domain in the image i. Because the domain $\Lambda_i^o(y^{(i)})$ depends on each image, we prefer to work on the template domain Λ_T^o which is common to all the images. Therefore we perform the approximated integral change of variable, for each image i: $t^{(i)} = f_{y^{(i)}}^{-1}(s)$. After the change of variable the log-likelihood expression on the object region is:

$$\begin{aligned}\ell_1(x_1^N|y_1^N) &\simeq \sum_{i=1}^{N} \sum_{t\in\Lambda_T^o} \ln g(x(f_{y^{(i)}}(t)); x_o(t), \tau_o(t))|J_{f_{y^{(i)}}}(t)|, \\ &= -\sum_{t\in\Lambda_T^o} \sum_{i=1}^{N} \left[\ln \tau(t)\sqrt{2\pi} + \frac{1}{2\tau_o^2(t)}(x(f_{y^{(i)}}(t)) - x_o(t))^2\right] |J_{f_{y^{(i)}}}(t)|. \end{aligned} \tag{6.13}$$

Thanks to the change of variable it is possible to swap the sums and to work for each pixel $t \in \Lambda_T^o$ independently. At each location t, the Gaussian parameters are estimated from a weighted sample of N intensity values $x^{(i)}(f_{y^{(i)}}(t)), 1 \leq i \leq N$:

$$\forall t \in \Lambda_T^o, \quad \hat{x}_o(t) = \frac{\sum_{i=1}^N x^{(i)}(f_{y^{(i)}}(t))|J_{f_{y^{(i)}}}(t)|}{\sum_{i=1}^N |J_{f_{y^{(i)}}}(t)|}, \tag{6.14}$$

$$\forall t \in \Lambda_T^o, \quad \hat{\tau}_o^2(t) = \frac{\sum_{i=1}^N (x^{(i)}(f_{y^{(i)}}(t)) - \hat{x}_o(t))^2|J_{f_{y^{(i)}}}(t)|}{\sum_{i=1}^N |J_{f_{y^{(i)}}}(t)|}. \tag{6.15}$$

The estimators of the deformable object template corresponds to the estimator of the DIM template, (3.22) and (3.23), except that here they are valid for $t \in \Lambda_T^o$ only while in Chapter 3 these expressions were valid for the whole image domain Λ.

The estimation of the background model is obtained by maximizing the second term of the likelihood function:

$$\ell_2(x_1^N|y_1^N) = \sum_{i=1}^{N} \sum_{s\in\Lambda^b(y^{(i)})} \ln g(x^{(i)}(s); x_b(t_b), \tau_b(t_b)), \tag{6.16}$$

with $t_b = f_{\bar{y}}^{-1}(s)$. The pixels included in the sum are not the same for all the images since $\Lambda^b(y^{(i)})$ depends on the image. We introduce a delta function to rewrite the likelihood on the whole domain and perform the change of variable: $t_b = f_{\bar{y}}^{-1}(s)$. Since $f_{\bar{y}}$ is the identity, its Jacobian is 1 for all $t \in \Lambda_T$, thus:

$$\ell_2(x_1^N|y_1^N) = \sum_{i=1}^{N} \sum_{t_b\in\Lambda_T} \delta(f_{\bar{y}}(t_b) \in \Lambda_i^b) \ln g(x^{(i)}(f_{\bar{y}}(t_b)); x_b(t_b), \tau_b(t_b)). \tag{6.17}$$

The resulting estimates of the Gaussian parameters are:

$$\forall t \in \Lambda_T, \quad \hat{x}_b(t_b) = \frac{\sum_{i=1}^{N} \delta(f_{\bar{y}}(t_b) \in \Lambda_i^b) x^{(i)}(f_{\bar{y}}(t_b))}{\sum_{i=1}^{N} \delta(f_{\bar{y}}(t_b) \in \Lambda_i^b)}, \tag{6.18}$$

$$\forall t \in \Lambda_T, \quad \hat{\tau}_b^2(t_b) = \frac{\sum_{i=1}^{N} \delta(f_{\bar{y}}(t_b) \in \Lambda_i^b)(x^{(i)}(f_{y^{(i)}}(t_b)) - \hat{x}_b(t_b))^2}{\sum_{i=1}^{N} \delta(f_{\bar{y}}(t_b) \in \Lambda_i^b)}. \tag{6.19}$$

(6.18) is the average intensity at pixel t_b, except that the images in which the object lies in t_b are disregarded and the average is computed on the rest of the training set. Similarly (6.19) is the classical MLE except that again the images containing the object at the considered pixel are not used for this computation. If the background is observed in all the images, the estimate is simply the average or classical ML variance estimate. However if the background is observed in none of the training images, the background distribution cannot be estimated. There are two ways to address this issue. One can either add a prior distribution such as a Gaussian distribution with the average image intensity as a mean and a very large variance, so that the pixel with few observations can be estimated, or choose the partition of the object and the background in a way that there exists always at least one (or more) images to estimate the background model.

6.2.3 Choice of the Partition

In computer vision, the choice of the moving object is often obvious. If one works on a scene in which a person is moving in front of a cluttered background, the person is usually modeled as the moving object. It is possible to estimate the background model as long as there exists few frames in the video sequence for which the person moves and the background appears. In medical imaging though, while the anatomy is variable, the global organization is preserved between images. The misalignment of the structure in the image can offer the possibility to estimate a background model, but often, as soon as the images are roughly aligned, parts of the structures are always superimposed. It is therefore not possible to estimate the background at some of the pixel. One solution to this lack of observation is to revisit the notion of moving object. For example in the case of the Corpus Callosum, instead of considering the whole structure as the moving object, one can decide that the boundary of the object is in practice the deformable object. Indeed, if the images are roughly aligned, this is mostly the boundaries of the structure that varies.

6.2.4 Landmark Detection

Similarly to what was done with the DIM (Chapter 3), the location of the landmarks y in a new image x is obtained by likelihood maximization using a gradient ascent method.

We recall the conditional log-likelihood of an image, denoting $t = f_y^{-1}(s)$ and $t_b =$

$f_{\bar{y}}^{-1}(s)$:

$$\ell(x|y) = \sum_{s\in\Lambda^o(y)} \ln g(x(s); x_o(t), \tau_o(t)) + \sum_{s\in\Lambda^b(y)} \ln g(x(s); x_b(t_b), \tau_b(t_b)) \tag{6.20}$$

or

$$\ell(x|y) \propto \sum_{s\in\Lambda^o(y)} \ln \frac{g(x(s); x_o(t), \tau_o(t))}{g(x(s); x_b(t_b), \tau_b(t_b))}. \tag{6.21}$$

We bring the expression back onto the template support using the change of variable $t = f_y^{-1}(s)$. Thus, $t_b = f_{\bar{y}}^{-1} \circ f_y(t)$,

$$\ell(x|y) \propto \sum_{t\in\Lambda_T^o} |J_{f_y}(t)| \ln \frac{g(x(f_y(t)); x_o(t), \tau_o(t))}{g(x(f_y(t)); x_b(t_b), \tau_b(t_b)}, \tag{6.22}$$

$$\propto \sum_{t\in\Lambda_T^o} |J_{f_y}(t)| \left[\ln \frac{\tau_b(t_b)}{\tau_o(t)} + \frac{(x(s) - x_b(t_b))^2}{\tau_b^2(t_b)} - \frac{(x(s) - x_o(t))^2}{\tau_o^2(t)} \right]. \tag{6.23}$$

In the latter expression the Jacobian, the intensity function x and the background template parameters $x_b(t_b), \tau_b(t_b)$ are functions of y. It is possible to write the analytical expression of the Jacobian derivative with respect to y. The image, the template mean and variance are considered as functions of $\mathbb{R}^d$ and their derivatives are computed by the chain rule. We denote by $f_y^{(l)}$ the lth coordinate of the spatial transformation f_y and $\frac{\partial}{\partial c_l}$ the derivative with respect to y of the l-th cartesian coordinate.

$$\frac{\partial x}{\partial y_{kh}}(f_y(t)) = \sum_{l=1}^{d} \frac{\partial x}{\partial c_l}(f_y(t)) \cdot \frac{\partial f_y^{(l)}}{\partial y_{kh}}(t), \tag{6.24}$$

$$\frac{\partial x_b}{\partial y_{kh}}(f_{\bar{y}}^{-1} \circ f_y(t)) = \sum_{l=1}^{d} \frac{\partial x_b}{\partial c_l}(f_{\bar{y}}^{-1} \circ f_y(t)) \cdot \frac{\partial f_y^{(l)}}{\partial y_{kh}}(t), \tag{6.25}$$

$$\frac{\partial \tau_b}{\partial y_{kh}}(f_{\bar{y}}^{-1} \circ f_y(t)) = \sum_{l=1}^{d} \frac{\partial \tau_b}{\partial c_l}(f_{\bar{y}}^{-1} \circ f_y(t)) \cdot \frac{\partial f_y^{(l)}}{\partial y_{kh}}(t). \tag{6.26}$$

It follows that the gradient of (6.23) with respect to y is:

$$\frac{\partial \ell(x|y)}{\partial y} = \sum_{t\in\Lambda_T^o} \frac{\partial |J_{f_y}(t)|}{\partial y} \left[\ln \frac{\tau_b(t_b)}{\tau_o(t)} + \frac{(x(f_y)(t) - x_b(t_b))^2}{\tau_b^2(t_b)} - \frac{(x(f_y(t)) - x_o(t))^2}{\tau_o^2(t)} \right]$$
$$+ \sum_{t\in\Lambda_T^o} |J_{f_y}(t)| \sum_{l=1}^{d} \left[D(t, y, l) \cdot \frac{\partial f_y^{(l)}}{\partial y} \right], \tag{6.27}$$

with,

$$D(t, y, l) = \left\langle \begin{pmatrix} \frac{x(f_y(t)) - x_b(t_b)}{\tau_b^2(t_b)} - \frac{x(f_y(t)) - x_o(t)}{\tau_o^2(t)} \\ -\frac{x(f_y(t)) - x_b(t_b)}{\tau_b^2(t_b)} \\ -\frac{(x(f_y(t)) - x_b(t_b))^2}{\tau_b^3(t_b)} + \frac{1}{\tau_b(t_b)} \end{pmatrix}, \begin{pmatrix} \frac{\partial x}{\partial c_l}(f_y(t)) \\ \frac{\partial x_b}{\partial c_l}(t_b) \\ \frac{\partial \tau_b}{\partial c_l}(t_b) \end{pmatrix} \right\rangle.$$

The gradient is computed on the moving object support only whichever choice of deformation was made. If the background model is constant, i.e. the same at all $t \in \Lambda_T$, the gradient expression boils down to the classical gradient of the DIM except for the support which is a partition of the template while it was the whole template support in Chapter 3.

The Deformable Intensity Object (DIO) that we proposed inherits of the same issues as the Deformable Intensity Model, i.e. it is sensitive to the variations of intensity distribution between images. The main advantage of DIO over DIM is the possibility to reduce the computation of the cost function to a finite domain of the image even if the deformation model has infinite support.

6.3 Tissue-based Deformable Intensity Object

In the deformable tissue object (T-DIO) model, we adapt the idea of the deformable object to the Tissue-based Deformable Intensity Model (T-DIM) presented in Chapter 5.

6.3.1 Image Likelihood

We recall that the T-DIM was composed of a photometric model, modeled by a mixture of Gaussian distributions and a geometric model, which is modeled by a deformable model of the tissue types of the image. In the case of the T-DIO model, we keep the idea of a photometric model which applies to the whole image, but the deformable model is now composed of two elements: the deformable object and the background. We use the same notations as in the preceding section. Recall that $(\Lambda_T^o, \Lambda_T^b)$ is a partition of the template grid Λ_T, and $(\Lambda^o(y), \Lambda^b(y))$ the image partition resulting from the deformation f_y. $x(s)$ denotes the intensity value at s and $z(s)$ denotes the corresponding tissue type. This is an unobserved discrete random variable. Finally $y \in \mathbb{R}^{dK}$ is the vector of the K landmarks. The image-specific photometric parameters $\mu(j,i), \sigma^2(j,i)$ are considered as nuisance parameters, while the distribution of the tissue types given the landmarks location is encoded by the deformable object model. The log-likelihood is:

$$\ln p_\theta(x^{(i)}|y^{(i)}) = \sum_{s\in\Lambda_i} \ln \sum_{j=1}^{J} p_\theta(x^{(i)}(s)|z(s)=j)p(z(s)=j|y^{(i)}). \tag{6.28}$$

The specificity of the DO model is that the probability $p(z(s)=j|y)$ is modeled differently depending upon the pixel s belongs to the object or to the background:

$$\forall s,j,y, \quad p(z(s)=j|y) = \begin{cases} \pi_o(j,t), & \text{if } s \in \Lambda^o(y), \quad t = f_y^{-1}(s), \\ \pi_b(j,t_b), & \text{if } s \in \Lambda^b(y), \quad t_b = f_{\bar{y}}^{-1}(s). \end{cases} \tag{6.29}$$

We choose to model the photometry of the image with a single mixture model whose parameters are the image specific nuisance parameters:

$$\forall i, \forall s, \quad p_\theta(x^{(i)}(s)|z(s)=j) = g(x^{(i)}(s); \mu(j,i), \sigma^2(j,i)). \tag{6.30}$$

The likelihood function can be rearranged as a sum over the moving object and a constant term:

$$\forall i, \quad \ln p_\theta(x^{(i)}|y^{(i)}) \propto \sum_{s\in\Lambda^o(y^{(i)})} \ln \frac{\sum_{j=1}^J g(x^{(i)}(s);\mu(j,i),\sigma^2(j,i))\pi_o(j,f_{y^{(i)}}^{-1}(s))}{\sum_{j=1}^J g(x^{(i)}(s);\mu(j,i),\sigma^2(j,i))\pi_b(j,f_{\bar{y}}^{-1}(s))}. \tag{6.31}$$

6.3.2 Model Estimation

Because the tissue types at each pixels are unobserved, the estimation of the model parameters is performed by the EM algorithm. Denoting θ the parameters $(\mu(j,i),\sigma(j,i),1\leq j\leq J,1\leq i\leq N;\pi_o(j,t),t\in\Lambda_T^o,\pi_b(j,t),t\in\Lambda_T)$, the auxiliary Q-function of the EM algorithm for a training set of images (x_1^N,y_1^N) is:

$$\begin{aligned} Q(\theta,\theta') &= \mathbb{E}_z\left[\ln p_\theta(x_1^N,z_1^N|y_1^N)|x_1^N,y_1^N\right], \qquad (6.32)\\ &= \sum_{z^{(1)}(s)}\cdots\sum_{z^{(N)}(S)}\left[\ln p_\theta(x_1^N,z_1^N|y_1^N)\right]p_{\theta'}(z_1^N|x_1^N,y_1^N),\\ &= \sum_i\sum_s\sum_j\left[\ln p_\theta(x^{(i)}(s),z^{(i)}(s)=j|y^{(i)})\right]p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)}). \end{aligned}$$

The Q-function is a sum of three terms:

$$Q(\theta,\theta') = \sum_i\sum_s\sum_j \ln g(x^{(i)}(s);\mu(j,i),\sigma^2(j,i))p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)}), \tag{6.33}$$

$$+\sum_i\sum_{s\in\Lambda_i^o(y^{(i)})}\sum_j \ln\pi_o(j,f_{y^{(i)}}^{-1}(s))p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)}), \tag{6.34}$$

$$+\sum_i\sum_{s\in\Lambda_i^b(y^{(i)})}\sum_j \ln\pi_b(j,f_{\bar{y}}^{-1}(s))p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)}). \tag{6.35}$$

The E-step is as usual the computation of the posterior distribution for all i, all j and all s, but the expression depends on whether s belongs to the object or to the background,

$$\forall s\in\Lambda_i^o(y^{(i)}), \quad p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)}) \propto g(x^{(i)}(s),\mu'(j,i),\sigma'^2(j,i))\pi_o(j,f_{y^{(i)}}^{-1}(s)), \tag{6.36}$$

$$\forall s\in\Lambda_i^b(y^{(i)}), \quad p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)}) \propto g(x^{(i)}(s),\mu'(j,i),\sigma'^2(j,i))\pi_b(j,f_{\bar{y}}^{-1}(s)). \tag{6.37}$$

The M-step consists in maximizing the Q-function with respect to the nuisance parameters $\mu(j,i),\sigma^2(j,i)$ for all i and j but also with respect to the template parameters $\pi_o(j,t)$ for all j and $t\in\Lambda_T^o$, $\pi_b(j,t)$ for all j and $t\in\Lambda_T$. The optimization can be performed independently with respect to each of these sets of parameters. The update of the nuisance parameters comes from the maximization of (6.33):

$$\forall i,j, \quad \hat{\mu}(j,i) = \frac{\sum_s x^{(i)}(s)p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)})}{\sum_s p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)})}, \tag{6.38}$$

$$\forall i,j, \quad \hat{\sigma}^2(j,i) = \frac{\sum_s (x^{(i)}(s)-\mu'(j,i))^2 p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)})}{\sum_s p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s),y^{(i)})}. \tag{6.39}$$

The update expression of the object template is obtained by maximizing (6.34). We perform the approximated integral change of variable $t = f_{y^{(i)}}^{-1}(s)$ for each image:

$$\begin{aligned} Q_2(\theta,\theta') &= \sum_i \sum_{s\in\Lambda_i^o(y^{(i)})} \sum_j \ln \pi_o(j, f_{y^{(i)}}^{-1}(s)) p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s), y^{(i)}) \\ &\simeq \sum_i \sum_{t\in\Lambda_T^o} \sum_j |J_{f_{y^{(i)}}}(t)| \ln \pi_o(j,t) p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t))=j|x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}). \end{aligned} \tag{6.40}$$

Hence,

$$\forall j, \forall t \in \Lambda_T^o, \quad \pi_o(j,t) \propto \sum_i |J_{f_{y^{(i)}}}(t)| p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t))=j|x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}). \tag{6.41}$$

Finally the estimate of the background model is obtained by maximizing (6.35). We perform the change of variable $t_b = f_{\bar{y}}^{-1}(s)$:

$$\begin{aligned} &Q_3(\theta,\theta') \\ &= \sum_i \sum_{s\in\Lambda_i^b(y^{(i)})} \sum_j \ln \pi_b(j, f_{\bar{y}}^{-1}(s)) p_{\theta'}(z^{(i)}(s)=j|x^{(i)}(s), y^{(i)}) \\ &\simeq \sum_i \sum_{t_b\in(f_{\bar{y}}^{-1}\circ f_{y^{(i)}})(\Lambda_T^b)} \sum_j \ln \pi_b(j,t_b) p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t_b))=j|x^{(i)}(f_{y^{(i)}}(t_b)), y^{(i)}), \\ &\simeq \sum_i \sum_{t_b\in\Lambda_T} \sum_j \delta(t_b \in (f_{\bar{y}}^{-1}\circ f_{y^{(i)}})(\Lambda_T^b)) \ln \pi_b(j,t_b) p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t_b))=j|x^{(i)}(f_{y^{(i)}}(t_b)), y^{(i)}). \end{aligned}$$

It follows that $\forall j, \forall t \in \Lambda_T$,

$$\pi_b(t,j) \propto \sum_i \delta(t \in (f_{\bar{y}}^{-1}\circ f_{y^{(i)}})(\Lambda_T^b)) p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t))=j|x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}). \tag{6.42}$$

The training algorithm derived from this model is similar to Algorithm 5.6. The estimators of the object intensity distribution correspond exactly to the estimators of the T-DIM parameters. The background template estimate is a simple sum of the posterior probabilities, but using only the images that contain some background at the pixel of interest. The important point is that even though the computation of the template estimate is locally performed, because the intensity model is common to the background and deformable object, all pixels are used to perform the estimation of the photometric model. In the experiments in 3D of the preceding chapter, we have seen that it was necessary in terms of computation to use a subset of the image only. By doing so, not only the template was estimated locally only, but also the photometric model was estimated from a small region of the image. It sometimes creates some issues in the estimation of the photometric parameters that are avoided with T-DIO.

6.3.3 Landmark Detection

Because both the nuisance parameters and the segmentation are unknown, the estimation of the landmark location requires to use the EM algorithm. The purpose is to estimate

simultaneously the photometric parameters $\theta = (\mu(j), \sigma^2(j), \forall j)$ and the landmark location y. The Q-function is in this case:

$$\begin{aligned} Q(\theta, y; \theta', y') &= \mathbb{E}_z \left[\ln p_\theta(x, z|y)\right] p_{\theta'}(z|x, y'), \\ &= \sum_s \sum_j \left[\ln p_\theta(x(s)|z(s)) p(z(s)|y)\right] p_{\theta'}(z(s) = j|x(s), y'). \end{aligned} \tag{6.43}$$

Using the same reasoning as for the training set, the Q-function is a sum of three terms:

$$\begin{aligned} Q(\theta, y; \theta', y') = &\sum_s \sum_j \ln g(x(s); \mu(j), \sigma^2(j)) p_{\theta'}(z(s) = j|x(s), y') \\ &+ \sum_{s \in \Lambda^o(y)} \sum_j \ln q_o(j, f_y^{-1}(s)) p_{\theta'}(z(s) = j|x(s), y') \\ &+ \sum_{s \in \Lambda^b(y)} \sum_j \ln q_b(j, f_{\bar{y}}^{-1}(s)) p_{\theta'}(z(s) = j|x(s), y'). \end{aligned} \tag{6.44}$$

The E-step is as usual, the computation of the posterior distribution for all j and s and stays the same as for training:

$$\forall s \in \Lambda^o(y), \quad p_{\theta'}(z(s) = j|x(s), y) \propto g(x(s), \mu'(j), \sigma'^2(j)) \pi_o(j, f_y^{-1}(s)), \tag{6.45}$$

$$\forall s \in \Lambda^b(y), \quad p_{\theta'}(z(s) = j|x(s), y) \propto g(x(s), \mu'(j), \sigma'^2(j)) \pi_b(j, f_{\bar{y}}^{-1}(s)). \tag{6.46}$$

The estimation of the photometric parameters is also the same as in the training algorithm:

$$\forall j, \quad \hat{\mu}(j) = \frac{\sum_s x(s) p_{\theta'}(z(s) = j|x(s), y')}{\sum_s p_{\theta'}(z(s) = j|x(s), y')}, \tag{6.47}$$

$$\forall j, \quad \hat{\sigma}^2(j) = \frac{\sum_s (x(s) - \mu'(j))^2 p_{\theta'}(z(s) = j|x(s), y')}{\sum_s p_{\theta'}(z(s) = j|x(s), y')}. \tag{6.48}$$

The optimization with respect to the landmark location is difficult to carry out because of the posterior distribution, therefore we replace the maximization of the Q-function by a gradient-based maximization of $\ell(x|y; \hat{\theta})$, the likelihood at the current estimates of the nuisance parameters. We have seen in Chapter 4, that by doing so, the likelihood function is still guaranteed to increase at each iteration of the EM algorithm.

Therefore we compute the gradient of the likelihood function as written in (6.31). However, the domain $\Lambda^o(y)$ depends on the location of the landmarks, and involves the computation of the inverse transformation f_y^{-1}. We use the integral change of variable $t = f_y^{-1}(s)$ and denote $t_b = f_{\bar{y}}^{-1} \circ f_y(t)$. Up to a constant, the expression of the likelihood function is:

$$\ell(x|y; \hat{\theta}) = \sum_{t \in \Lambda_T^o} |J_{f_y}(t)| \ln \frac{\sum_{j=1}^J g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j)) \pi_o(j, t)}{\sum_{j=1}^J g(x(f_y(t)); \hat{\mu}(j), \hat{\sigma}^2(j)) \pi_b(j, t_b)}. \tag{6.49}$$

The gradient is therefore a sum of three terms (to simplify the notation we denote $g(x(f_y(t)), \hat{\mu}(j), \hat{\sigma}(j))$ by $g(x(f_y(t)), j)$,

$$\begin{aligned}\frac{\ell(x|y;\hat{\theta})}{\partial y} &= \sum_{t\in\Lambda_T^o} \frac{\partial |J_{f_y}(t)|}{\partial y} \ln \frac{\sum_{j=1}^J g(x(f_y(t));j)\pi_o(j,t)}{\sum_{j=1}^J g(x(f_y(t));j)\pi_b(j,t_b)} \\ &+ \sum_{t\in\Lambda_T^o} \frac{\partial x(f_y(t))}{\partial y} \cdot \frac{\sum_j \frac{\hat{\mu}(j)-x(f_y(t))}{\hat{\sigma}^2(j)} \pi_o(j,t) g(x(f_y(t));j)}{\sum_j \pi_o(j,t) g(x(f_y(t));j)} \\ &- \sum_{t\in\Lambda_T^o} \frac{\partial x(f_y(t))}{\partial y} \cdot \frac{\sum_j \left[\frac{\hat{\mu}(j)-x(f_y(t))}{\hat{\sigma}^2(j)} \pi_b(j,t_b) + \frac{\partial \pi_b(j,t_b)}{\partial y}\right] g(x(f_y(t));j)}{\sum_j \pi_b(j,t_b) g(x(f_y(t));j)}\end{aligned} \tag{6.50}$$

This is the third term of the above gradient expression that makes the difference between T-DIM and T-DIO. Algorithms 6.9 and 6.10 respectively summarize the training and testing algorithms for the T-DIO model. It is very similar to the algorithm 5.6 derived in Chapter 5. The differences comes from the computation of the posterior distribution, which in the case of the T-DIO model has a different form depending whether the pixel falls into the object or the background. In practice T-DIM can be seen as a specific case of the T-DIO, in which the background model is a Uniform distribution at each pixel.

6.4 Experiments

We test the DIO algorithm for the simultaneous detection of SCC1 and SCC2 in the 2D-SCC data set. Before estimating the model parameters, we need to define the partition $(\Lambda_T^o, \Lambda_T^b)$ of the template support. We choose Λ_T^o as the union of the discs centered in SCC1 and SCC2 of fixed radii. We keep using the Gaussian spline model for the deformation. We look at the performance of the algorithm at detecting both SCC1 and SCC2 for different kernel parameters (σ between 3 and 10 pixels) and for different deformable object sizes (radii of Λ_T^o between 2 and 5 pixels). Figure 6.3 illustrates the learnt background and deformable object templates when $\sigma = 10$ and when the radii of the deformable object region is 4 pixels. The leftmost image represents the mean intensity at each pixel in the background model, while the rightmost image represents the mean intensity for the pixels belonging to the deformable object. Notice that the background mean is not very smooth around the tip. At a pixel t of the background model, we disregarded from the intensity average estimation the images that contain the object at the corresponding location. Therefore the set of images used to estimate the intensity distribution varies depending on the pixel location. It follows that the intensity template appears less smooth than the template learnt in the preceding chapters.

The lowest average prediction error for SCC1 is 1.52 mm with a standard deviation of 0.87 mm. As for SCC2 the best prediction performance reached is 0.91 mm of average error with a standard deviation of 0.43 mm. This is comparable to the results we have obtained with DIM.

Figure 6.4 represents the predicted landmark locations on three images of the testing set, as well as the region of the image that is identified as the deformable object. The

Algorithm 6.9 Deformable Tissue Model: Training

LEARNING

Let (x_1^N, y_1^N) be a training set of images and $(\Lambda_T^o, \Lambda_T^b)$ a partition of Λ_T the template support. $\theta = \{\forall j, \forall i, \mu(j,i), \sigma^2(j,i); \forall j, \forall t \in \Lambda_T^o, \pi_o(j,t), \forall t, \pi_b(j,t)\}$ denotes the set of parameters.

Initialize $\forall j, \forall i,\ \mu(j,i), \sigma^2(j,i)$, and $\forall j, \forall t \in {\Lambda_T}^o, \pi_o(j,t)$, $\forall j, \forall t \in \Lambda_T, \pi_b(j,t)$.
Iterate until convergence:

- **E-step:** compute:

$$\forall i, \text{ if } s \in \Lambda_i^o(y^{(i)}), \quad p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}) \propto g(x^{(i)}(s), \mu'(j,i), \sigma'^2(j,i)) \pi_o(j, f_{y^{(i)}}^{-1}(s)),$$

$$\forall i, \text{ if } s \in \Lambda_i^b(y^{(i)}), \quad p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}) \propto g(x^{(i)}(s), \mu'(j,i), \sigma'^2(j,i)) \pi_b(j, f_{\bar{y}}^{-1}(s)).$$

- **M-step:**
  - Update the photometric parameters,

$$\forall j, i, \qquad \mu(j,i) \leftarrow \frac{\sum_s x^{(i)}(s) p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)})}{\sum_s p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)})},$$

$$\forall j, i, \qquad \sigma^2(j,i) \leftarrow \frac{\sum_s \left(x^{(i)}(s) - \mu(j,i)\right)^2 p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)})}{\sum_s p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)})},$$

  - Update the template estimate,

$$\forall j, t \in \Lambda_T^o, \quad \pi(j,t) \propto \sum_i |J_{f_{y^{(i)}}}(t)| p_\theta(z^{(i)}(s) = j | x^{(i)}(s), y^{(i)}),$$

$$\forall j, t \in \Lambda_T, \quad \pi_b(t,j) \propto \sum_i \delta(t \in (f_{\bar{y}}^{-1} \circ f_{y^{(i)}})(\Lambda_T^b)) p_{\theta'}(z^{(i)}(f_{y^{(i)}}(t)) = j | x^{(i)}(f_{y^{(i)}}(t)), y^{(i)}).$$

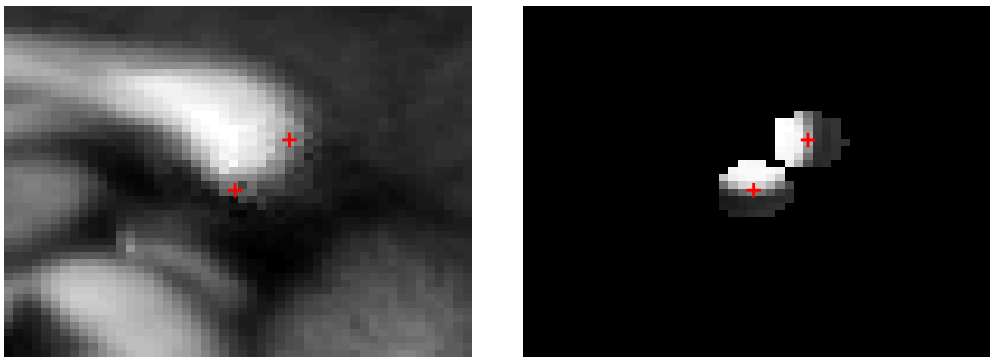


Figure 6.3: Template of the Deformable Intensity Object model. In both figures the red crosses represent the location of the landmarks. The Left subfigure represents the mean μ_b of the background intensity model. The rightmost image represents the mean intensity in the deformable object μ_o. Recall that only the deformation model acts only on the deformable object template.

Algorithm 6.10 Deformable Tissue Model: Testing

TESTING

Let x be a testing image of unknown photometric parameters $\theta = (\mu(j), \sigma^2(j), 1 \leq j \leq J)$ and π_o, π_b the parameters learnt during training,

Initialize $\forall j, \mu(j), \sigma^2(j)$ and $y \leftarrow \bar{y}$
Iterate until convergence

- **E-step:** compute:

$$\text{if } s \in \Lambda^o(y), \quad p_\theta(z(s) = j | x(s), y) \propto g(x(s), \mu'(j), \sigma'^2(j)) \pi_o(j, f_y^{-1}(s)),$$
$$\text{if } s \in \Lambda^b(y), \quad p_\theta(z(s) = j | x(s), y) \propto g(x(s), \mu'(j), \sigma'^2(j)) \pi_b(j, f_{\bar{y}}^{-1}(s)).$$

- **M-step:**

  - **Update** the photometric parameters

$$\forall j, \qquad \mu(j) \leftarrow \frac{\sum_s x(s) p_\theta(z(s) = j | x(s), y)}{\sum_s p_\theta(z(s) = j | x(s), y)}.$$
$$\forall j, \qquad \sigma^2(j) \leftarrow \frac{\sum_s \left(x^{(i)}(s) - \mu(j)\right)^2 p_\theta(z(s) = j | x(s), y)}{\sum_s p_\theta(z(s) = j | x(s), y)},$$

  - **Compute** the gradient direction $\frac{\partial \ell}{\partial y}(x|y;\theta)$ from (6.50).

  - **Update** the location of the landmarks,

$$y \leftarrow y + a \cdot \frac{\partial \ell(x|y;\theta)}{\partial y}, \quad \text{with} \quad a \leftarrow \arg\max_{a \in \mathbb{R}^+} \ell\left(x | y + a \frac{\partial \ell(x|y;\theta)}{\partial y}; \theta\right),$$

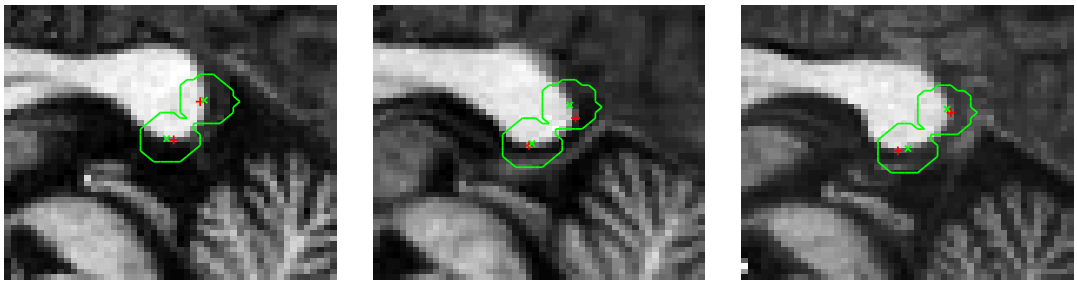


Figure 6.4: Landmark detection. This figure illustrates the landmark detection results on 3 images of the testing set. In each image, the red crosses represent the location of the landmarks located by the expert and in green the location of the automatic landmarks. The green contour marks the limit between the pixels belonging to the deformable object and those belonging to the background, when the landmarks are located at the green crosses.

prediction in the rightmost and the leftmost images is good, but in the middle image, there is a vertical error in the detection of SCC1.

6.5 Chapter Conclusion

Let us look to finish at a slightly different problem that can be solved with a Deformable Object Model and applied to many situations in medical imaging. Let x be an image which results of the action of a random deformation f on the probabilistic template. An object is superimposed to that image, but this time the geometry of the object is unknown and its intensity distribution is independent from its location in the image. Let $z(s)$ be a discrete hidden random variable which encodes the segmentation at pixel s. If pixel s belongs to the object then $z(s) = 1$, but if the another pixel s belongs to the background image, $z(s) = 0$. We write the log-likelihood of the image assuming conditional independence of the image pixels:

$$\begin{aligned}\ell(x) &= \sum_{s\in\Lambda} \ln \sum_{j=0}^{1} p(x(s)|z(s)=j)p(z(s)=j),\\ &= \sum_{s\in\Lambda} \ln \left[z(s)p(x(s)|z(s)=1)p(z(s)=1) + (1-z(s))p(x(s)|z(s)=0)p(z(s)=0)\right].\end{aligned} \tag{6.51}$$

We model the conditional intensity distribution as follows:

$$\begin{aligned}&\forall s, \quad (x(s)=u|z(s)=1) \sim p_o(u),\\ &\forall s, \quad (x(s)=u|z(s)=0) \sim g\left(u; \boldsymbol{x}_b(f^{-1}(s)), \tau_b(f^{-1}(s))\right).\end{aligned}$$

p_o is the intensity distribution of the object. g is the probability of observing a grayscale value at pixel s. It is given by a deformable probabilistic intensity model (x_b, τ_b). Let us assume that we have learnt the model (x_b, τ_b) from some training data. Three cases may occur:

- The registering transformation f^{-1} is known, but both the segmentation z and the distribution of the object p_o are unknown. One can use the EM algorithm to simultaneously learn the segmentation distribution and the distribution of the superimposed object. The resulting algorithm is similar to an outlier detection procedure. The pixels that are not well explained by the deformable intensity model are assigned to the object. In medical imaging, this type of algorithm can be used for the delineation of an abnormal tissue.

- The distribution of the object p_o is known, but the segmentation z and the registering deformation f^{-1} are unknown. Again this problem can be solved by the EM algorithm, taking care of modeling the deformation as a hidden variable or as a nuisance parameter. This situation occurs when one tries to register two images with occlusion.

- Finally if neither p_o, nor the segmentation z, nor the deformation f^{-1} is known, this is the most challenging case. If there exists an identifiable solution, the EM algorithm is again one way to deal with this optimization problem.

The latter situation occurs relatively frequently in medical imaging when two images were taken at different time or when a contrast agent is added. Examples of problems in which this situation occurs are: the delineation of the infarct from delayed enhancement MR images with the help of another heart image which was captured before the contrast agent was given to the patient; the study of the evolution of an infectious disease in the lung using CT images.

In summary the Deformable Object Model combined with the algorithms presented in the preceding chapters provides a unified formulation for many problems of medical imaging.

CHAPTER 7

CONCLUSION

7.1 Summary

The models proposed in the preceding chapters have all in common the definition and estimation of a probabilistic deformable model. Along the chapters the complexity of the model increases. The algorithms for the estimation of the model and the prediction of the landmark locations are modified. Since in all cases the training and testing algorithms are derived from the modeling assumptions using likelihood principles, the progression of the algorithms follows the model changes.

In the Deformable Intensity Model, which is the simplest of the deformable models. We model the intensity observed in the image by a Gaussian distribution at each pixel, hence the estimation of the template is pretty straightforward and boils down to the weighted average of the registered images. The testing algorithm consists simply in maximizing the likelihood with respect to the landmark locations by gradient ascent. The model is extremely simple to estimate and provides a simple intensity matching algorithm. However the precision of the landmark detection depends heavily on the adequacy of the intensity distribution in the template and in the image. Since it is not always possible (or desirable) to match the intensities by processing the images, we propose two models which are not affected by intensity variations.

The first approach we proposed is to model the distribution of edges in the image, since edges are invariant to the intensity changes. Therefore the Deformable Edge Model (DEM) is not affected by intensity variation, neither are the resulting training and testing algorithms. Modeling the edge distribution rather than the image intensities has two major consequences though. First it is not possible to generate from the learnt model an intensity image. In addition since noise affects the edge detector output, we need to introduce a hidden variable encoding the presence of an edge in the template. Training needs to be performed via the EM algorithm. The testing algorithm though is still a gradient ascent. While the resulting optimization function are independent from the intensity variation, they also deal with sparse information in the image. Unfortunately the sparseness of the information in the image slightly reduces the performance of the algorithm compared to DIM.

We therefore proposed another intensity invariant model, the Tissue-based Deformable Intensity Model (T-DIM), which models the tissue distribution at each pixel by a deformable model and the intensity distribution of each image by a mixture of Gaussian distributions. This model overcomes the difficulties of both DIM and DEM. The joint estimation of the deformable template and of the intensity parameter is performed using an EM algorithm. The testing algorithm corresponding to the complete generative model is again a gradient ascent. For practical reasons in the experiments we prefer to consider the intensity as a

nuisance parameter. In that case the testing algorithm derived from the model is also based on the EM algorithm since both the intensity model and the segmentation of the image are unknown. This model has a wide range of applications beyond detecting landmarks. It can be used to derive joint segmentation/registration algorithms.

For larger images and for 3D images, the T-DIM algorithms becomes quite computationally intense. One solution consists in working with local deformations only so that the support of the likelihood variation is finite and therefore the computation can be reduced to a subregion of the image. However limiting the support of the deformations means that one excludes affine deformations from the possible deformation model. To overcome this limitation we proposed to modify the image model and to introduce an object-based approach. The image is modeled as the superimposition of a deformable model on top of a still background. This model can be adapted to both the DIM and the T-DIM. We obtain the Deformable Intensity Object (DIO) and the Tissue-based Deformable Intensity Object (T-DIO). Both models compare the probability that an image fragment comes either from the deformable object or from the background. The computation is reduced to a subregion of the image.

We have illustrated the different models on the detection of landmarks in brain MR images. The performance achieved on the detection of SCC are of the order of 1mm which is the pixel resolution in our case. The common advantage of the set of models proposed is that it is not necessary to have a prior knowledge on the type of landmarks and the geometry of the underlying structure in the image. Thanks to the training set of images, it is possible to learn automatically a template rather than tailoring manually the detector for each type of landmark. It generalizes to any type of landmarks and also to the simultaneous detection of a variable number of landmarks. The proposed models are also tested on the much more challenging detection of the head of the hippocampus HoH. Because the region is poor in characterizing features, we have encountered difficulties with some of the proposed algorithms to detect landmarks in the hippocampus. We can still show a significant improvement thanks to our method as shown in Chapters 3 and 5.

7.2 Assessing the Performance of the Algorithms

Assessing the performance of an automatic landmarking method is not easy and rises many questions. In all the presented algorithms, we use a likelihood maximization method to predict the location of the landmarks. We would like to assess the quality of the prediction by building some confidence ellipses. Let x be a set of observations and θ the model parameters, estimated by the MLE estimator $\hat{\theta}_{MLE}$. The maximum likelihood estimator is asymptotically normal:

$$\hat{\theta}_{MLE} \rightarrow \mathcal{N}\left(\theta; \frac{1}{N} I^{-1}(\theta)\right), \text{ with } N \text{ the number of samples,}$$

$$\text{and the Fisher Information matrix, } I(\theta) = \mathbb{E}_x\left[\left(\frac{\partial}{\partial\theta}\ell(x|\theta)\right)^\top \left(\frac{\partial}{\partial\theta}\ell(x|\theta)\right)\right].$$

Since the expectation with respect to all possible images cannot be computed in general, the Fisher information matrix is estimated by:

$$\hat{I}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial\theta}\ell(x^{(i)}|\hat{\theta})\right]^{\top}\left[\frac{\partial}{\partial\theta}\ell(x^{(i)}|\hat{\theta})\right].$$

In the proposed models, it is generally possible to compute analytically both the first and second partial derivatives of the likelihood function with respect to the model parameters. The resulting estimate of the Fisher Information Matrix is used to build confidence ellipses around the landmarks. It is expected that the ellipse will align with the average direction of the surrounding edge.

In our work we measure the quality of a predicted landmark by computing the Euclidean distance between the predicted location and the position marked by the expert. This measurement does not take into account the error of the expert himself. One solution would be to ask several expert to landmark the images and compute the distance between the predicted location and the set of manual landmarks. Alternatively, another way to measure properly the precision is to measure the distance between the predicted landmark and the correction of its location by an expert. Finally since the landmark locations are often used for registration, one could compare the output of the registration performed using the automatic landmarks or the manual landmarks. the expert landmarks and the automatic landmarks.

7.3 Other Applications for Medical Imaging

Throughout the chapters we have illustrated the proposed models on the detection of landmarks in brain MRI. It is naturally applicable to any type of imaging modalities that produces binary or scalar images, as long as a training set of landmarked images is available. The method generalizes to 3D and even to video sequences (3D+Time). In order to generalize the proposed model to non-scalar modalities, it is necessary to propose a statistical model of the image data at each pixel or voxel and to understand how deformations act on this type of data. Considering the instances Diffusion Tensor Images, the measurement at each voxel is a tensor. In order to estimate a probabilistic deformable template, one needs to define a statistical model of the tensor variations and to understand how the deformations act on the tensors. Indeed in the case of scalar images the grayscale value is simply transported on the image grid. When deforming a DTI image though, the local directional tensor needs to be realigned [59]. Several models of action of deformations have been proposed. In [24], the authors proposed a statistical model that can be used for the description of a set of DTI tensors.

Even though the models were illustrated on the detection of landmarks in this thesis, there exist other important problems of medical imaging to which this type of approach could be applied. We already mentioned in Chapter 6 some applications for the Deformable Intensity Object. The models we proposed are composed of two parts: the statistical model of the image measurements and the deformation model. Changes in the

deformation model can broad up the possible applications of these models. If the number of control points increases so that it is possible to deform the whole image support using the spline interpolated deformation model, the local registration algorithm for landmark detection becomes an image registration algorithm. Working with a larger number of control points rises two main issues.

A first solution consists in imposing some constraints on the displacement of the control points such that crossings and foldings are avoided. There exists already an extended literature in the domain of diffeomorphic landmark matching [44, 11, 84]. Another point of view is to prevent crossing locally, imposing constraints on the relative position or displacement of the control points, [12, 47].

The second issue consists in learning the statistical model, when the number of control points increases. Indeed the algorithms we have presented relies on the exact matching of the landmarks in the training set. If the number of control points increases it becomes difficult specially in 3D volumes to mark manually matching points setting correspondences between images. Fortunately, the family of models we proposed relies on sparse correspondences to model the image variations and it is possible to locate landmarks only on the regions to be registered. In [68], the author proposes a method for spline based interpolation on data with directional error. By incorporating the localization error in the spline estimation algorithm, it is possible to relax the strict correspondences between the landmarks during training.

Therefore if the proposed statistical models are coupled to some of the existing methods for image deformation, we can derive some registration algorithms from the Deformable Intensity Model and Deformable Edge Model. They would respectively be intensity-based or edge-based matching methods. In the case of the Tissue-based Deformable Intensity Model, not only the images are matched but in addition a segmentation of the images is obtained. Therefore coupled to a deformation model covering the entire image, the T-DIM can be used for joint registration /segmentation of the images. Finally since the T-DIM models the intensity distribution separately from the geometry, it is possible to work on multi-modality image registration techniques.

BIBLIOGRAPHY

[1] S. Allassonnière. *Representation and Estimation of Deformable Template Models for Shape Recognition and Computational Anatomy*. PhD thesis, Université Paris XIII, July 2007.

[2] S. Allassonnière, Y. Amit, and A. Trouvé. Toward a coherent statistical framework for dense deformable template estimation. *Journal of the Royal Statistical Society*, B(69):3–29, 2007.

[3] J. Ashburner and K. J. Friston. Nonlinear spatial normalization using basis functions. In *Human Brain Mapping*, volume 7, pages 254–266, 1999.

[4] J. Ashburner and K. J. Friston. Unified segmentation. *NeuroImage*, (26):839–851, 2005.

[5] R. Bajcsy and S. Kovačič. Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, (46):1–21, 1989.

[6] D. I. Barnea and H. F. Silverman. A class of algorithms for fast digital image registration. *IEEE Transactions on Computers*, 21(2):179–186, 1972.

[7] M. F. Beg, P. A. Helm, E. McVeigh, M. I. Miller, and R. L. Winslow. Computational cardiac anatomy using mri. *Magnetic Resonance Medicine*, 52(5):1167–1174, November 2004.

[8] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, 1989.

[9] F. L. Bookstein. *Morphometric Tools for Landmark Data: Geometry and Biology*. Cambridge University Press, February 1992.

[10] M. Bro-Nielsen and C. Gramkow. Fast fluid registration of medical images. In *Proceeding of 4th International Conference on Visualization in Biomedical computing (VBC'96)*, volume 1131 of *Lecture Notes in Computer Science*, pages 267–276. Springer Berlin Heidelberg, 1996.

[11] V. Camion and L. Younes. Geodesic interpolating splines. In *Proceedings of Energy Minimisation in Computer Vision and Pattern Recognition (EMMCVPR)*, volume 2134, pages 513–527. Springer-Verlag, 2001.

[12] Y. Choi and S. Lee. Injectivity conditions of 2d and 3d uniform cubic b-spline functions. *Graphical Models*, 6(62):411–427, 2000.

[13] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marshal. Automated multi-modality image registration based on information theory. In C. B.

Y Bizais and R. D. Paola, editors, *Information Processing in Medical Imaging*, page 263–274. Kluwer Academic, 1995.

[14] T. F. Cootes and C. J. Taylor. Statistical models of appearance for medical image analysis and computer vision. In *SPIE 2001 Medical Imaging*, volume 4322, pages 236–248, 2001.

[15] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. Technical report, University of Manchester, March 2004.

[16] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[17] R. Cox. Afni: Software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research*, 29:162–173, 1996.

[18] C. Davatzikos. Spatial transformation and registration of brain imaging using elastically deformable models. *Computer Vision and Image Understanding*, 2(66):207–222, 1997.

[19] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39:1–38, 1977.

[20] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley, 1998.

[21] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion. *R.A.I.R.O. Analyse Numerique*, (10:12):63–76, 1976.

[22] H. M. Duvernoy. *The Human Hippocampus: Functional Anatomy, Vascularization and Serial Sections with MRI*. Springer, 2nd edition, 1998.

[23] A. Evans, D. Collins, S. Mills, E. Brown, R. Kelly, and T. Peters. 3d statistical neuroanatomical models from 305 mri volumes. In *Nuclear Science Symposium and Medical Imaging Conference*, volume 3, pages 1813–1817, 1993.

[24] P. Fillard, V. Arsigny, X. Pennec, and N. Ayache. Clinical DT-MRI estimation, smoothing and fiber tracking with log-Euclidean metrics. *IEEE Transactions on Medical Imaging*, 26(11):1472–1482, Nov. 2007. PMID: 18041263.

[25] B. Fischer and J. Modersitzki. Combination of automatic non-rigid and landmark based registration: the best of both worlds. In M. Sonka and J. M. Fitzpatrick, editors, *Medical Imaging 2003: Image Processing. Edited by Sonka, Milan; Fitzpatrick, J. Michael. Proceedings of the SPIE, Volume 5032, pp. 1037-1048 (2003).*, pages 1037–1048, May 2003.

[26] B. Fischl, D. H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Makris, B. Rosen, and A. M. Dale. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33:341–355, january 2002.

[27] B. Fischl, D. H. Salat, A. J. van der Kouwe, N. Makris, F. Ségonne, B. T. Quinn, and A. M. Dale. Sequence-independent segmentation of magnetic resonance images. *NeuroImage*, (23):S69–S84, 2004.

[28] P. T. Fletcher, L. Conglin, S. M. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005, August 2004.

[29] S. Frantz, K. Rohr, and H. Stiehl. Localization of 3D anatomical point landmarks in 3D tomographic images using deformable models. In *Proc. MICCAI*, volume 1935 of *Lecture Notes in Computer Science*, pages 492–501, 2000.

[30] J. Friedman. Regularized discriminant analysis. *Journal of the Royal Society of Statistics*, 84:17–42, 1989.

[31] K. J. Friston, J. Ashburner, J. B. Poline, C. D. Frith, J. D. Heather, and R. Frackowiak. Spatial registration and normalisation of images. In *Human Brain Mapping*, volume 2, pages 165–189, 1995.

[32] D. Geman and B. Jedynak. An active testing model for tracking roads from satellite images. *IEEE Trans. Pattern Anal. Mach. Intell*, 18:1–14, 1996.

[33] C. Glasbey and K. Mardia. A penalized likelihood approach to image warping (with discussion). *Journal of the Royal Statistical Society*, B(63):465–514, 2001.

[34] J. Glaunès and S. Joshi. Template estimation from unlabeled point set data and surfaces for computational anatomy. In X. Pennec and S. Joshi, editors, *Proc. of the International Workshop on the Mathematical Foundations of Computational Anatomy (MFCA-2006)*, pages 29–39, October 2006.

[35] A. Goshtasby, L. Staib, C. Studholme, and D. Terzopoulos. Non-rigid image registration: Guest editors' introduction. *Computer Vision and Image Understanding*, 89(2/3):109–113, 2003.

[36] P. Green and B. Silverman. *Nonparametric Regression and Generalized Linear Models, A roughness penalty approach*, volume 58 of *Monographs on Statistics and Applied Probability*. CRC Press, 1994.

[37] T. Greene and W. Rayens. Partially pooled covariance estimation in discriminant analysis. *Communications in Statistics*, 18(10):3679–3702, 1989.

[38] U. Grenander. *General pattern theory : a mathematical study of regular structures*. Oxford University Press, 1993.

[39] U. Grenander and M. Miller. Computational anatomy: An emerging discipline. In *Quaterly of Applied Mathematics*, number 4, pages 617–694. LVI, December 1998.

[40] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[41] C. Harris and M. Stephens. A combined corner and edge detection. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[42] P. Hellier and C. Barillot. Coupling dense and landmark-based approaches for non rigid registration. *IEEE Transactions on Medical Imaging*, 22(2):217–227, 2003.

[43] H. Johnson and G. Christensen. Consistent landmark and intensity-based image registration. *IEEE Transactions on Medical Imaging*, 21:450–461, 2002.

[44] S. Joshi and M. Miller. Landmark matching via large deformation diffeomorphisms. In *IEEE Trans. in Image Processing*, volume 9, pages 1357–1370, 2000.

[45] J.Talairach and P. Tournoux. *Co-planar stereotaxic Atlas of the Human Brain*. Thieme Medical Publishers, 1988.

[46] K. V. Leemput. A statistical framework for partial volume segmentation. In W. Niessen and M. Viergever, editors, *MICCAI*, volume 2208 of *Lecture Notes in Computer Science*, pages 204–212, 2001.

[47] K. V. Leemput. Probabilistic brain atlas encoding using bayesian inference. In R. Larsen, M. Nielsen, and J. Sporring, editors, *MICCAI*, volume 4190 of *Lecture Notes in Computer Science*, pages 704–711, 2006.

[48] K. V. Leemput, F. Maes, D. Vandermeulen, and P. Suetens. Automated model-based bias field correction of MR images of the brain. *IEEE Transactions on Medical Imaging*, 18(10):885–896, October 1999.

[49] K. V. Leemput, F. Maes, D. Vandermeulen, and P. Suetens. Automated model-based tissue classification of MR images of the brain. *IEEE Transactions on Medical Imaging*, 18(10):897–908, October 1999.

[50] H. Lester, S. Arridge, K. Jansons, L. Lemieux, J. Hajnal, and A. Oatridge. Non-linear registration with the variable viscosity fluid algorithm. In *Information Processing in Medical Imaging (IPMI'99)*, pages 238–251, 1999.

[51] H. Li, B. S. Manjunath, and S. K. Mitra. A contour–based approach to multisensor image registration. *IEEE Transactions on Image Processing*, 4(3):320–334, March 1995.

[52] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[53] J. Ma, M. I. Miller, A. Trouvé, and L. Younes. Bayesian template estimation in computational anatomy. (in process).

[54] F. Maes, A. Collignon, D. Vandermeulen, G. Marshal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16:187–198, 1997.

[55] G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. John Wiley, 1997.

[56] M. Miller. Computational anatomy: shape, growth and atrophy. *NeuroImage*, 23:S19–S33, 2004.

[57] M. Miller, C. Priebe, and Y. Park. Collaborative computational anatomy: the perfect storm for mri morphometry study of the human brain via diffeomophic metric mapping, multidimensional scaling and linear discriminant analysis. *Proceedings of the National Academy of Science*, to appear.

[58] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm, 2001.

[59] J.-M. Peyrat, M. Sermesant, X. Pennec, H. Delingette, C. Xu, E. R. McVeigh, and N. Ayache. A computational framework for the statistical analysis of cardiac diffusion tensors: Application to a small database of canine hearts. *IEEE Transactions on Medical Imaging*, 26(11):1500–1514, November 2007. PMID: 18041265.

[60] K. M. Pohl, J. Fisher, W. E. L. Grimson, R. Kikinis, and W. M. Wells. A bayesian model for joint segmentation and registration. *NeuroImage*, 31(1):228–239, May 2006.

[61] K. M. Pohl, W. M. Wells, A. Guimond, K. Kasai, M. E. Shenton, R. Kikinis, W. E. L. Grimson, and S. K. Warfield. Incorporating non-rigid registration into expectation-maximization algorithm to segment mr images. In T. Dohi and R. Kikinis, editors, *MICCAI*, volume 2488 of *Lecture Notes in COmputer Science*, pages 564–571, 2002.

[62] M. Prastawa, J. Gilmore, W. Lin, and G. Gerig. Automatic segmentation of MR images of the developing newborn brain. *Medical Image Analysis*, 9:457–466, 2005.

[63] W. K. Pratt. Correlation techniques for image registration. *IEEE Trans. Aerospace and Electronic Systems*, 10(3):353–358, 1974.

[64] C. E. Priebe, M. I. Miller, and J. T. Ratnanather. Segmenting magnetic resonance images via hierarchical mixture modelling. *Computational Statistics and Data Analysis*, 2004.

[65] A. Roche, G. Malandain, and N. Ayache. Unifying maximum likelihood approaches in medical image registration. *International Journal of Imaging Systems and Technology*, 11(1):71–80, 2000.

[66] K. Rohr. On 3d differential operators for detecting point landmarks. *Image and Vision Computing*, 15:3:219–233, 1997.

[67] K. Rohr. *Landmark-based Image Analysis using Geometric and Intensity Models*. Kluwer Academic, Dordrecht, 2001.

[68] K. Rohr, M. Fornefett, and H. Stiehl. Approximating thin-plate splines for elastic registration: Integration of landmarks errors and orientation attributes. In *International Conference on Information Processing in Medical Imaging (IPMI)*, volume 1613 of *Lecture Notes in Computer Science*, pages 252–265. Springer-Verlag Berlin Heidelberg, 1999.

[69] K. Rohr, H. Stiehl, R. Sprengel, T. Buzug, J. Weese, and M. Kuhn. Landmark-based elastic registration using approximating thin-plate splines. *IEEE Transactions on Medical Imging*, 20(6):526–534, June 2001.

[70] G. Stone. *Bivariate Splines*. PhD thesis, University of Bath, 1988.

[71] C. Studholme, D. L. G. Hill, and D. J. Hawkes. Multiresolution voxel similarity measures for MR–PET registration. In C. B. Y Bizais and R. D. Paola, editors, *Information Processing in Medical Imaging*, page 287–298. Kluwer Academic, 1995.

[72] S. Tadjudin and D. A. Landgrebe. Covariance estimation with limited training samples. *IEEE Transactions on Geoscience and Remote Sensing*, 37(4):2113–2118, July 1999.

[73] J.-P. Thirion. New feature points based on geometric invariants for 3D image registration. *Int. J. of Computer Vision*, 18:2:121–137, 1996.

[74] M. Turk and A. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Learning*, pages 586–591, 1991.

[75] C. Twining, S. Marsland, and C. Taylor. Measuring geodesic distances on the space of bounded diffeomorphisms.

[76] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[77] P. Viola. *Alignment by maximization of mutual information*. PhD thesis, Massachusetts Institute of Technology, 1995.

[78] G. Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1990.

[79] F. Wang, B. C. Vemuri, and S. J. Eisenschenk. Joint registration and segmentation of neuroanatomic structures from brain mri. *Academic Radiology*, 13(9):1104–1111, September 2006.

[80] W. Wells, R. Kikinis, W. Grimson, and F. Jolesz. Adaptive segmentation of MRI data. In *IEEE Trans. Med. Imag.*, volume 15, pages 429–442, 1996.

[81] C. F. J. Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[82] S. Wörz and K. Rohr. 3d parametric intensity models for the localization of different types of 3d anatomical point landmarks in tomographic images. In *Proc. DAGM*, volume 2781 of *Lecture Notes in Computer Science*, pages 220–227, 2003.

[83] L. Younes. *Invariance, déformations et reconnaissance de formes*. Number 44 in Mathématiques et Applications. Springer, 2004.

[84] L. Younes. Combining geodesic interpolating splines and affine transformations. *IEEE Transactions on Image Processing*, 15(5), May 2006.

[85] L. Younes. Deformation analysis for shape and image processing. http://www.cis.jhu.edu/ younes/LectureNotes/diffShape.pdf, 2007.

[86] A. Zijdenbos, R. Forghani, and A. Evans. Automatic "pipeline" analysis of 3D MRI data for clinical trials: Application to multiple sclerosis. *IEEE Transactions on Medical Imaging*, 21(10):1280–1291, October 2002.

[87] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003.

MODÉLISATION ET ESTIMATION STATISTIQUE POUR L'IMAGERIE MÉDICALE : APPLICATION À LA DÉTECTION D'AMERS.

Nous proposons une famille de modèles statistiques à atlas déformable pour l'analyse d'images médicales et plus particulièrement pour la détection d'amer. Les modèles déformables sont couramment utilisés pour la mise en correspondance d'images en vue de leur segmentation, alignement ou classification. Nous montrons que si la position des amers caractérise la déformation d'une image, le problème de détection d'amer peut être formulé comme un problème de mise en correspondance locale. Dans un premier temps, nous présentons deux modèles statistiques qui utilisent les intensités ou les contours pour détecter les amers. Ensuite nous introduisons un modèle qui simultanément segmente une nouvelle image et la met en correspondance avec un atlas pour détecter les amers. À partir de chaque modèle proposé, nous obtenons, par maximum de vraisemblance, un algorithme d'apprentissage et un algorithme de détection d'amer. Enfin en introduisant le concept d'objet déformable et de fond d'image, il est possible de limiter les calculs aux sous parties de l'image qui caractérisent la position des amers. Les algorithmes présentés sont à la fois simples et génériques. Ils permettent de détecter automatiquement un ou plusieurs amers dans des images médicales. L'approche proposée est testée pour la localisation d'amers dans des Images à Résonance Magnétique de cerveau.

mots-clés : modèle déformable - modèles statistiques et estimation - alignement - segmentation - détection d'amer - imagerie médicale

STATISTICAL MODELING AND ESTIMATION IN MEDICAL IMAGING: AUTOMATIC DETECTION OF ANATOMICAL LANDMARKS

We present a family of statistical models based on deformable template for medical image analysis, and more specifically for the detection of anatomical landmarks. Deformable template models are commonly used for image matching to perform segmentation, registration or classification. We show that if the position of the landmarks characterizes uniquely the deformation of an image, the landmark detection problem can be formalized as a local matching problem. Based on the proposed statistical models and using maximum likelihood principles, we derive both an algorithm to learn the model from training data and a testing algorithm for the detection of landmarks in new images. The first two statistical models we propose rely on intensity or edge matching to identify the location of the landmarks; while the third one uses simultaneous image segmentation and template registration to locate the landmarks. We introduce a foreground/background statistical model for medical imaging, which allows us to limit the computational effort to matching discriminative patterns surrounding the landmarks. The proposed algorithms provide simple generic methods to perform automatic detection of landmarks in medical imaging. We tested our approach on the detection of landmarks in brain Magnetic Resonance Images.

keywords : deformable template - statistical modeling and estimation - registration - segmentation - landmark detection - medical imaging