Table des matières

Table des matières

Chapitre 1

1 Introduction

Ce travail de thèse porte sur la segmentation d'images traitée par les modèles statistiques de forme et d'apparence. Nous abordons sa problématique du point de vue théorique et pratique, ce qui nous a permis de proposer une démarche synthétique pour un autre type de mise en œuvre.

1.1 Le Contexte

Le traitement d'images voit ses débuts dans les années 1920 dans la transmission de données par câble mais ne connaît de vrai essor que dans les années 1960 avec le développement des ordinateurs. Au départ, les techniques de traitement d'images sont essentiellement des méthodes de restauration et de compression d'images. Avec les progrès de l'informatique, se développent des techniques de détection des primitives (contour, points d'intérêt, lignes d'intérêts, …) et de nombreux autres traitements dans les domaines aussi variés que le médical, la télévision, l'imagerie satellitaire, le multimédia. C'est dans les années 2000 que l'image numérique et par conséquent le traitement d'images devient omniprésent. Que cela soit sur internet, à la télévision, sur les téléphones, dans le domaine médical, l'image est partout (figure 1.1). Actuellement, il ne s'agit plus uniquement de traiter les images pour les améliorer, mais aussi de les comprendre et les interpréter. C'est dans ce contexte que la reconnaissance des objets dans les images devient un sujet de recherche important. Et pour reconnaître des objets afin d'interpréter les images, il est nécessaire souvent au préalable de les segmenter, c'est-à-dire séparer les objets d'intérêts du fond de l'image.

(a) Image sur téléphone portable (b) Image sur téléviseur (c) Image médicale (d) Image satellitaire

Figure 1.1. Diverses applications pour l'image [Herbulot 2007].

La segmentation d'images est un vaste sujet d'étude et fait partie des grands thèmes de l'imagerie numérique. C'est une étape essentielle en traitement d'image dans la mesure où elle conditionne l'interprétation qui va être faîte sur ces images. Elle est primordiale en analyse d'images du simple seuillage des niveaux de gris aux techniques plus complexes. Elle est considérée comme l'étape fondamentale de plusieurs processus d'analyse d'image dédiés à la détection ou l'identification des objets. Elle est un enjeu majeur et fait l'objet de nombreuses recherches. A ce titre de nombreuses publications font état de segmentation et de nombreux algorithmes ont été proposés durant les dernières décennies. En effet, la

segmentation dépend fortement de l'application et il n'existe pas de solution générale mais un ensemble d'outils mathématiques et algorithmiques qu'il est possible de combiner pour résoudre des problèmes spécifiques. Ces outils sont basés sur différentes approches : contour, région, texture, etc. Une classification des différentes techniques de segmentation a été proposé dans [Lecœur et al. 2007] et cite cinq catégories d'approches de segmentation que sont les segmentations utilisant les contours comme critère de décision, celles basées sur les régions, celles basées sur la forme, celles préférant une approche structurelle et celles faisant appel à la théorie des graphes (figure1.2). Segmenter une image signifie trouver ses régions homogènes et ses contours.

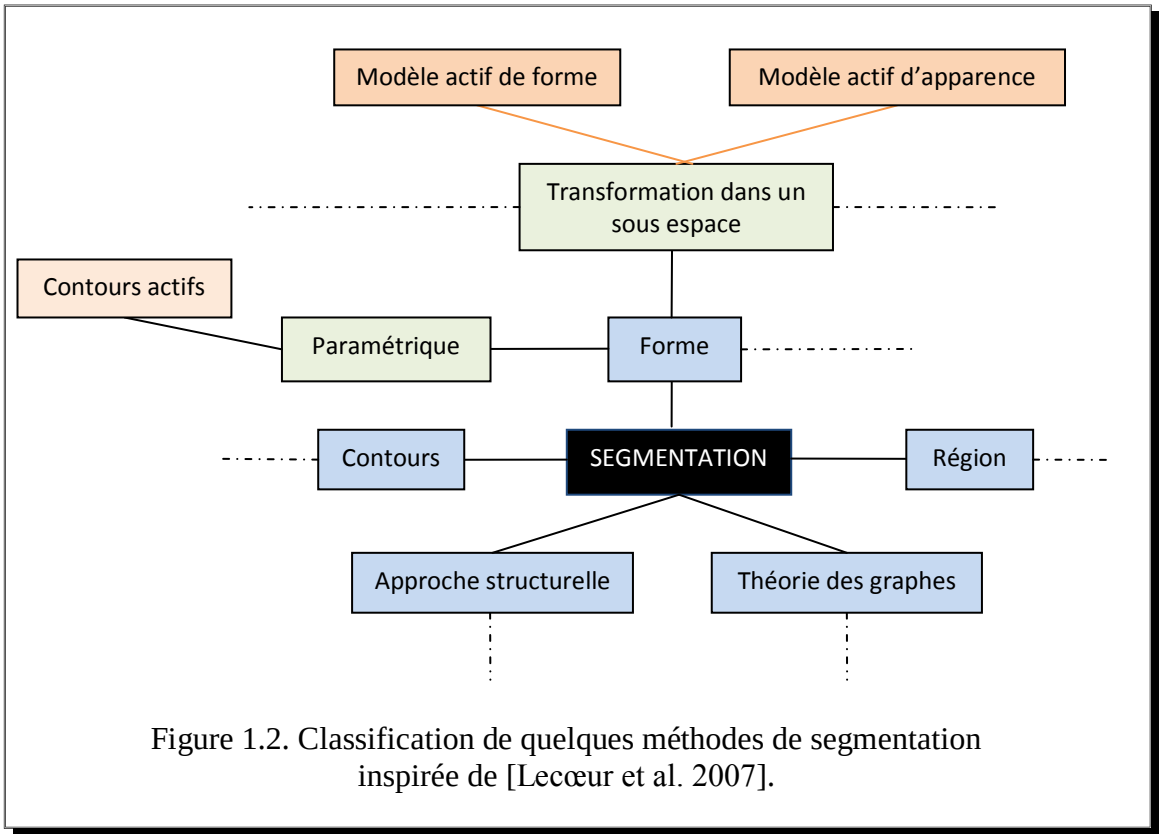


Figure 1.2. Classification de quelques méthodes de segmentation inspirée de [Lecœur et al. 2007].

L'importance de la segmentation croit avec celle de l'image de notre société. En imagerie médicale, la segmentation peut aider le médecin dans son diagnostic. En compression vidéo, elle permet de traiter différemment une zone d'intérêt qui bénéficiera d'une plus grande précision du reste de l'image qui pourra être fortement compressé. En indexation, la segmentation sert à extraire un objet que l'on souhaiterait retrouver dans d'autres images. Les applications sont nombreuses et la liste des exemples cités est loin d'être exhaustive. Il est bien entendu impossible de concevoir un algorithme qui soit robuste par rapport à toutes les perturbations possibles, et ce sur tous les types d'images envisageables. Toute avancée crée de nouveaux problèmes.

1.2 La position du problème

Les modèles statistiques de forme et d'apparence sont des processus complexes composés d'une chaîne séquentielle de traitements devant aboutir à la reconnaissance d'un objet prédéfini. Le terme modèle désigne ici la mise en forme des connaissances à priori, c'est-à-dire directement disponibles, et des connaissances à posteriori, c'est-à-dire issues des traitements intermédiaires, postérieurs aux traitements préalables. L'élaboration de tels modèles recouvre plusieurs aspects telles que la représentation des formes, la représentation des apparences, la mise en correspondance, la segmentation, et ils ont recours à des techniques d'analyse statistique. Les modèles obtenus sont exploitables dans de nombreux domaines d'application. De tels modèles permettent l'intégration des outils méthodologiques pour guider la segmentation selon les spécifications de l'application visée. La méthode se veut être adaptable et robuste au sens où elle donne des résultats aussi stables que possibles sur un échantillon de données tout en limitant les cas d'échecs. C'est pour cette raison que ces dernières années, les modèles actifs de forme et d'apparence sont devenus très populaires. Nous sommes donc confrontés à la fois à la méthodologie des modèles et à la problématique pratique des images par le biais du domaine vaste du traitement d'images. Afin de rester cohérent dans le contexte de la problématique des modèles statistiques de forme et d'apparence dédiés à la segmentation d'images, nous avons fondé une démarche qui a pu nous aider à appréhender progressivement la méthodologie du point de vue théorique et à concevoir une stratégie en vue de son implémentation.

1.3 La contribution de nos travaux

Les modèles statistiques de forme et d'apparence possèdent de nombreux attraits à la fois théoriques et pratiques et ouvrent des perspectives. Nos recherches sur ce sujet ont été initialement motivées par la recherche d'une étude détaillée de leur formalisme. Apparemment aucune étude portant sur les modèles statistiques de forme et d'apparence n'a porté de détails sur le formalisme théorique de ces modèles. Nous avons apporté plus de détails dans la description du formalisme standard des modèles dans le chapitre 3 (figure 1.3).

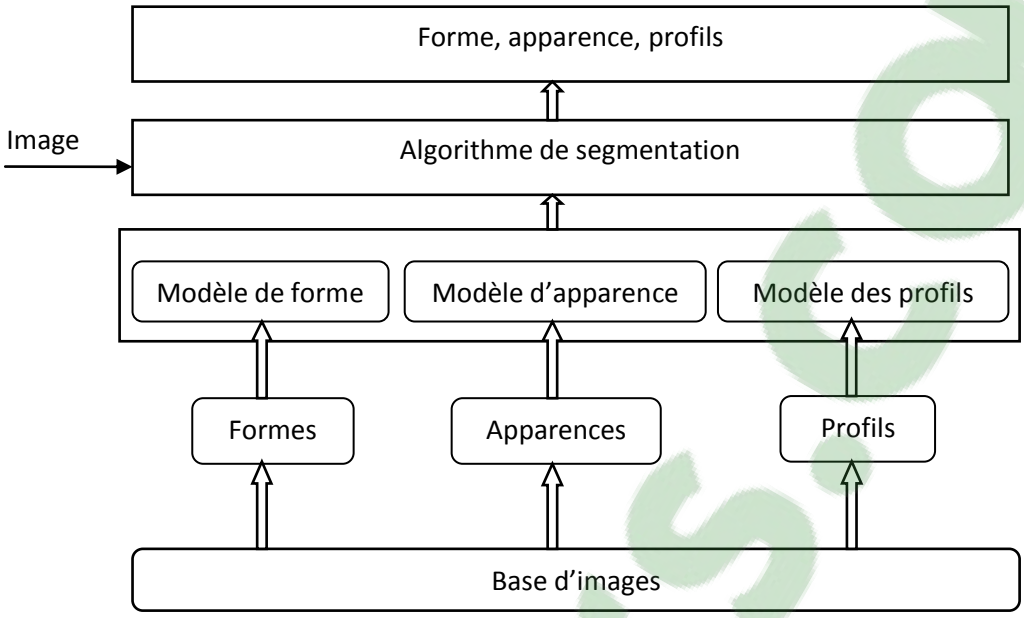


Figure 1.3. Modèle actif de forme et d'apparence [Naoui et al. 2014].

Nous avons ajouté également la possibilité de modifier la structure des modèles et nous proposons un schéma centré autour de leur base de connaissances, afin de le placer dans une perspective différente de celle présentée par la majorité des études portant sur les modèles statistiques de forme et d'apparence. La majorité des applications proposées dans la littérature sont séquentielles et nous n'avons identifié aucun paradigme clair qui explicite leur implémentation. Le schéma général que nous présentons à la figure 1.4 met en relief la base de connaissances à priori et à posteriori nécessaires pour guider la segmentation, et devant permettre l'interaction avec chaque opération du modèle.

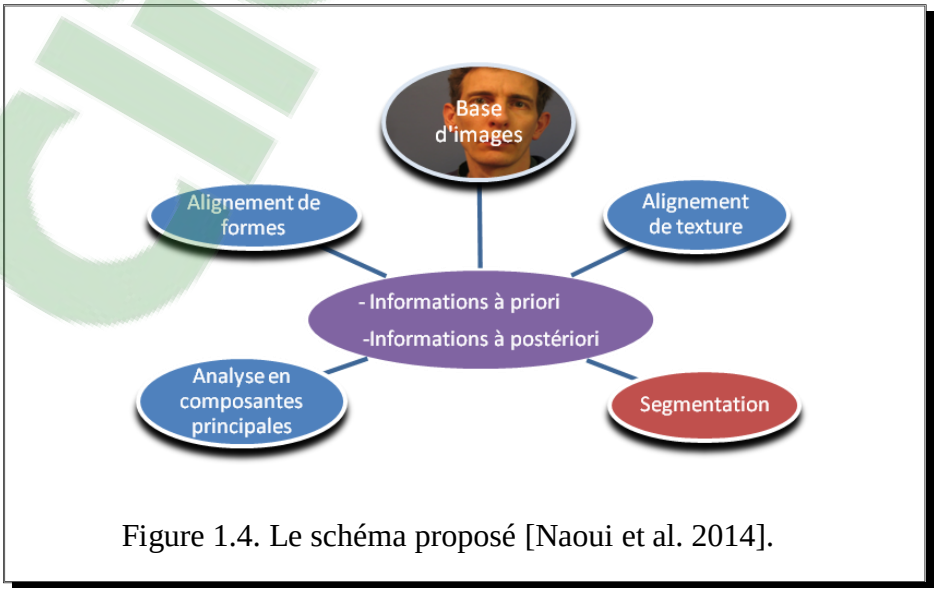


Figure 1.4. Le schéma proposé [Naoui et al. 2014].

Nos recherches sur la segmentation et les architectures avancées ont abouti à la conviction que le schéma que nous proposons se prête convenablement aux avantages des nouvelles architectures dédiées au domaine du traitement d'images, telles que les architectures distribuées et parallèles. Dans notre première contribution, nous l'avons formulé en termes de schéma distribué de type client-serveur [Naoui et al. 2013] et nous avons procédé à l'implémentation des deux premières opérations : l'alignement des formes et l'analyse en composantes principales. L'implémentation proposée se veut être générale et non spécifique à une application particulière. Notre deuxième contribution a porté sur l'idée d'intégrer le parallélisme dans l'implémentation des modèles. Nous l'avons formulé en termes de schéma distribué et parallèle [Naoui et al. 2014] et nous avons cherché à explorer dans le formalisme des modèles le type de parallélisme à appliquer. Tout d'abord nous nous sommes concentrés sur la nature de la structure des données manipulées dans les modèles. Les informations traitées par les modèles ont la particularité d'avoir une structure vectorielle et donc régulière. Le data-parallélisme est le modèle qui convient à de telle structure, et nous avons proposé de l'appliquer. Nous avons procédé à une étude de faisabilité pour les premières opérations des modèles qui traitent ce type d'informations. Pour chaque opération traitée, il était fondamental tout d'abord d'expliciter les formules de base, parfois d'en proposer des nouvelles, et ensuite de les projeter dans un parallélisme de données. Nous avons récupéré les mêmes formules pour des données différentes et nous avons déduit que l'étude de faisabilité pour le type de parallélisme proposé est confirmée. Dans le cas du modèle de forme, nous avons déduit qu'il est possible de séparer le modèle en deux sous modèles l'un relatif aux traitements des abscisses et l'autre relatif aux traitements des ordonnées. Une déduction importante qui mérite d'être revue et prise en compte dans nos futurs travaux. Comme pour la première contribution, nous avons procédé à des implémentations pour nous permettre d'une part de construire les premières briques de notre propre démarche d'implémentation et d'autre part pour tester l'avantage de nos propositions.

1.4 L'organisation du manuscrit

Ce document est organisé en six parties. Dans le chapitre 2, nous présentons un état de l'art des méthodes de segmentation d'images. Dans le chapitre 3, nous présentons les modèles statistiques de forme et d'apparence, leur problématique, et leur évolution. Les derniers chapitres sont consacrés au contexte de notre travail. Une partie est dédiée aux architectures avancées et aux architectures avancées dédiées au domaine de traitement d'images. Une autre partie est consacrée à nos contributions. Comme tout travail qui s'achève, nous concluons par un bilan général sur l'apport de nos contributions et nous présentons les perspectives émergeant de ce travail. Nous indiquerons les possibles améliorations ainsi que les perspectives pour la continuation de recherches sur la thématique abordée dans ce document.

Chapitre 2

Segmentation d'images : Etat de l'art

Dans ce chapitre, nous dressons un bref état de l'art des méthodes associées à la segmentation d'images. Etant donné le très grand nombre de publications associées à ces méthodes, nous avons sélectionné des exemples qui nous semblent représentatif des approches les plus fréquentes. Cependant des références complémentaires seront proposées tout au long de ce manuscrit, au fur et à mesure que les points méthodologiques abordés se préciseront.

2.1 Introduction

La segmentation joue un rôle prépondérant dans le domaine de traitement d'images. Elle est sans doute la tâche qui mobilise le plus d'efforts. Elle n'apparaît pas toujours de façon explicite, mais elle est toujours présente, même lorsque les images à analyser sont simples. Elle est une brique parmi un enchaînement de traitements répondant à un problème concret. Elle est réalisée avant les étapes d'analyse et de prise de décision dans plusieurs processus d'analyse d'image, telle que la détection des objets (figure 2.1). Elle aide à localiser et à délimiter les entités présentes dans l'image. L'intérêt de ces entités est de pouvoir être manipulé via des traitements de haut niveau pour extraire des caractéristiques de forme, de position, de taille, etc. La segmentation d'images constitue une partie clé de tels systèmes qui conditionne les étapes ultérieures. C'est un domaine réputé difficile en analyse d'images et qui englobe toute la problématique liée à la délimitation des zones. Il n'existe pas de théories s'appliquant à différents types d'images, mais plutôt des méthodes variées que l'on choisit et que l'on développe pour résoudre des problèmes d'analyse sur un type d'images bien défini. Le choix de la méthode est généralement conditionné par la représentation d'images que l'on choisit d'adopter [Cousty 2007].

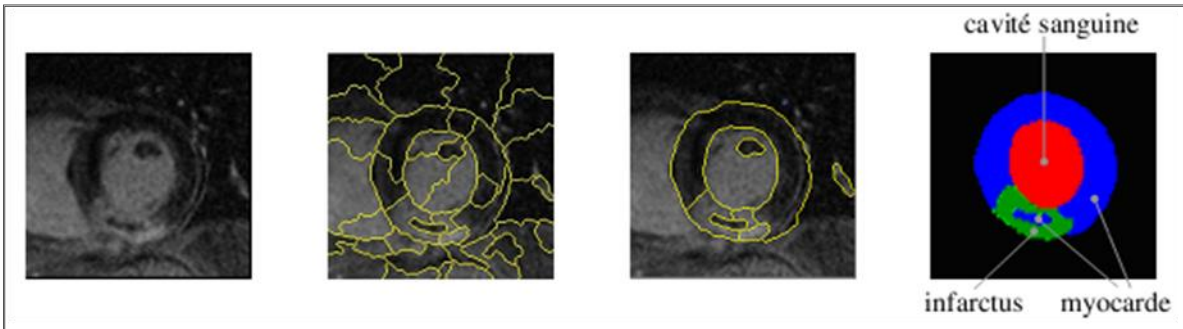


Figure 2.1. Exemple de segmentation d'image : (a) Image par résonnance magnétique cardiaque ; (b) séparation en régions ; (c) fusion de régions ; (d) résultat du processus de segmentation [Cousty 2007].

Devant la multitude des méthodes proposées, la segmentation d'images est toujours un sujet d'actualité et un problème permanent qui reste ouvert. Elle ne peut être appliquée que si elle remplit un certain nombre de critères afin de ne pas produire d'erreurs qui se répercuteraient sur l'ensemble de la chaîne de traitements. Ces critères se résument généralement en précision, et en robustesse. En pratique, ces critères ne sont pas complètement remplis, et il est important de prendre en compte le contexte d'utilisation envisagé pour concevoir une méthode de segmentation. Autrement dit, le critère principal doit être la réalisation de la tâche visée et qui devra permettra d'arriver à une bonne interprétation. En raison de la variabilité des images, et du grand nombre d'applications possibles, les obstacles rencontrés pendant la segmentation sont multiples [Meurie 2005] [Ciofolo 2005]. Les plus courants sont :

- La variabilité des formes à segmenter,
- Le bruit sur l'image,
- Le faible contraste et les frontières mal définies,
- La complexité des régions environnant la cible,
- L'hétérogénéité des intensités,
- Etc.

Le problème de l'évaluation de la qualité de la segmentation devient primordial. L'efficacité d'un algorithme de segmentation est illustrée habituellement par la présentation de quelques résultats de segmentation, ce qui n'autorise que des conclusions subjectives sur les performances de cet algorithme [Chabrier 2005]. Néanmoins un algorithme de segmentation doit au moins être stable et régulier. Valider correctement une segmentation nécessite souvent la vérité terrain. Une segmentation idéale serait en mesure de traiter une grande variété d'images issues de modalités diverses et de donner des résultats précis.

2.2 Définition

Segmenter une image signifie trouver ses régions homogènes et ses contours. Les régions doivent correspondre aux objets de l'image et les contours leurs frontières apparentes [Landré 2005, Lecoeur et al. 2007]. La définition formelle de la segmentation comme traitement de bas niveau date de l'année 1974 [Cocquerez et al. 1995] et propose de partitionner l'image *I* en sous ensembles *Ri* disjoints et connexes appelés régions tels que :

1- $\forall i \; R_i \neq \emptyset$
2- R_i est connexe
3- $P(R_i) = True \;\; \forall i$
4- $P(R_i \cup R_j) = False \;\; \forall i,j$
5- $\forall i,j \; R_i \cap R_j = \emptyset$
6- $I = \bigcup_i R_i$

Le terme de *région* fait référence à des parties de même dimension topologique que le support de l'image : 2D si l'image est en 2D, 3D si elle est en 3D, etc. [Guigues 2003].

La segmentation est donc l'affectation des pixels à des régions homogènes et disjointes formant une partition de l'image. Les pixels qui appartiennent à une même région partagent une propriété commune dite critère d'homogénéité. Cependant, il est important de souligner que le nombre de régions est indéterminé et qu'il peut donc exister plusieurs segmentations possibles pour un critère d'homogénéité donné. Pour cette subdivision en régions distinctes homogènes, il est reconnu deux grandes approches, l'approche région et l'approche frontière. Ces deux approches sont duales car une région définit une ligne sur son contour et une ligne fermée définit une région intérieure.

Les critères de segmentation forment la partie active du processus de segmentation. Ils définissent la partition à obtenir et dépendent de l'application. Ils sont nombreux et variés. Ils

sont souvent combinés afin de produire des résultats aussi pertinents que possible. Dans sa thèse, Dupas [2009] a cité quatre grandes classes de critères pour guider le processus de segmentation : ***les critères scalaires*** : basés sur les valeurs de l'image qui utilisent directement l'information de valeur des éléments de l'image afin de déterminer l'homogénéité. Parmi ces critères, nous trouvons les critères basés sur l'intensité des niveaux de gris ou l'intensité de couleur (la variance, l'histogramme local, la texture locale) ; ***Les critères fréquentiels*** : pour ce type de critères, l'image est considérée comme un signal multidimensionnel et les partitions sont déterminées à partir des opérations de traitement de signal. Les critères les plus utilisés sont la transformée de Fourier, la transformée en cosinus, et les ondelettes ; ***Les critères géométriques*** : représentent les paramètres géométriques mesurables d'un objet tels que : la dimension (longueur, largeur, etc.), l'aire, l'élongation (rapport longueur/largeur), le volume, la forme (courbure, etc.); ***Les critères topologiques*** : représentent les invariants topologiques des objets définis dans des espaces topologiques. Parmi ces critères, nous trouvons la caractéristique d'Euler, et les nombres de Betti.

Une étude formelle morpho-mathématique des deux notions « critère » et « segmentation » a été aussi abordée dans les travaux de Serra [2003]. Cette étude est fondée d'une part, sur la définition mathématique de la partition d'un ensemble en parties disjointes, connexes et maximales, et d'autre part, sur les propriétés que doit vérifier un critère de segmentation. Les propriétés établies pour un critère de segmentation devraient toujours permettre de trouver au moins une partition qui les vérifie.

2.3 Méthodes de segmentation

2.3.1 Problématique

La segmentation est un vaste sujet d'étude. Sa complexité et sa diversité justifient de nombreuses techniques. Les premières méthodes datent des années 80. Aujourd'hui, de nombreuses publications font état de segmentation, et leur nombre est impressionnant. On regroupe de façon usuelle les méthodes de segmentation en 4 groupes basés respectivement sur : - une approche globale de l'image dite pixellaire, – la recherche de frontières ; – la recherche de régions, - la coopération entre les trois premières. En plus de ces critères usuels, la classification proposée par Lecoeur [Lecoeur et al. 2007] dans le domaine de l'imagerie médicale, identifie d'autres critères : les méthodes basées sur la forme, celles préférant une approche structurelle et celles faisant appel à la théorie des graphes (figure 2.2).

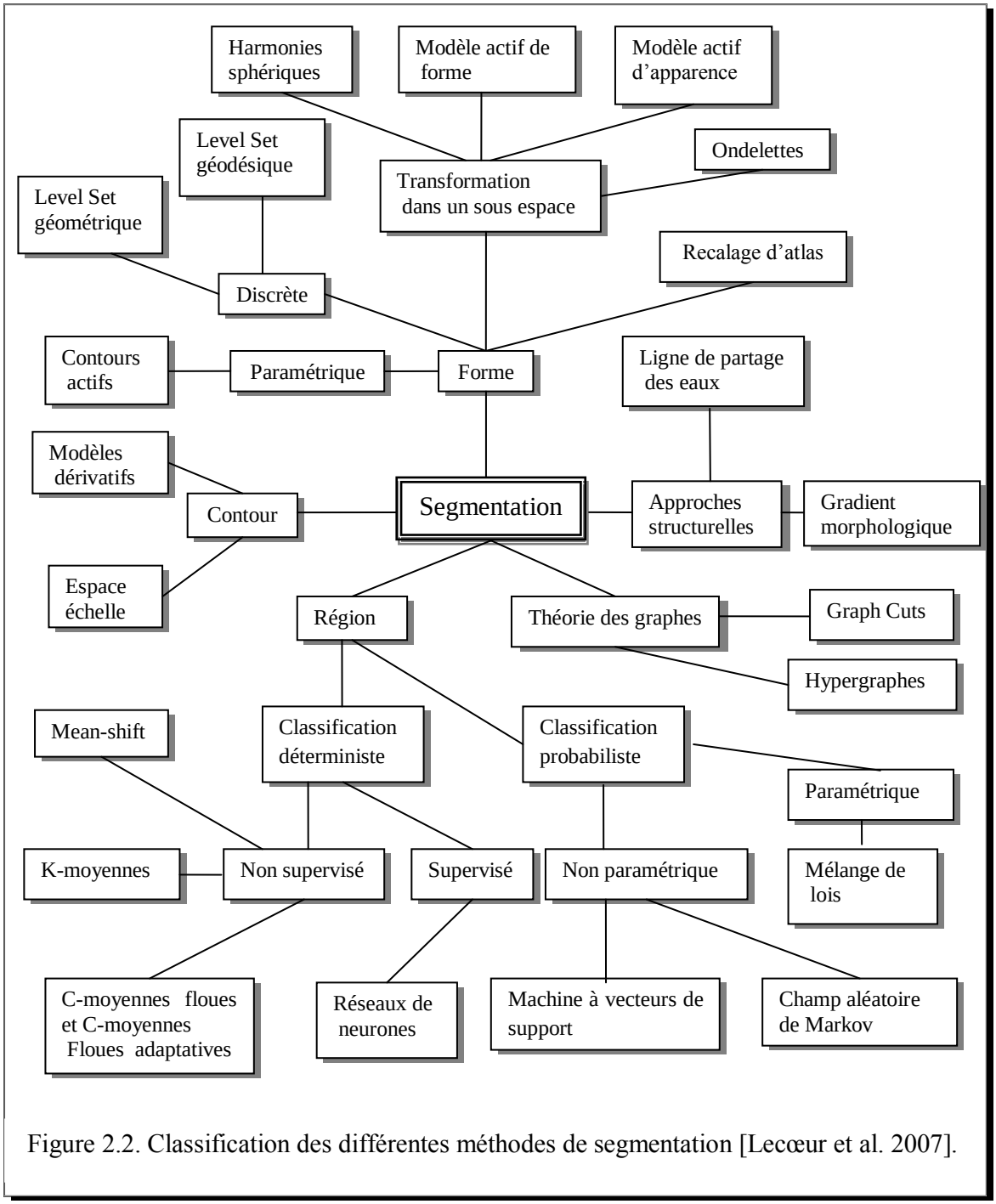


Figure 2.2. Classification des différentes méthodes de segmentation [Lecœur et al. 2007].

D'une manière générale, lorsqu'on parcourt la littérature consacrée à la segmentation, nous nous en inspirons pour dire que les méthodes de segmentation peuvent être classées en deux grandes catégories comme l'illustre la figure 2.3. L'approche directe de bas niveau extrait à partir de l'image seule une information pertinente, et l'approche indirecte de haut niveau fait intervenir une modélisation de l'image ou de la donnée recherchée [Montagnat 1999] [Dupas 2009].

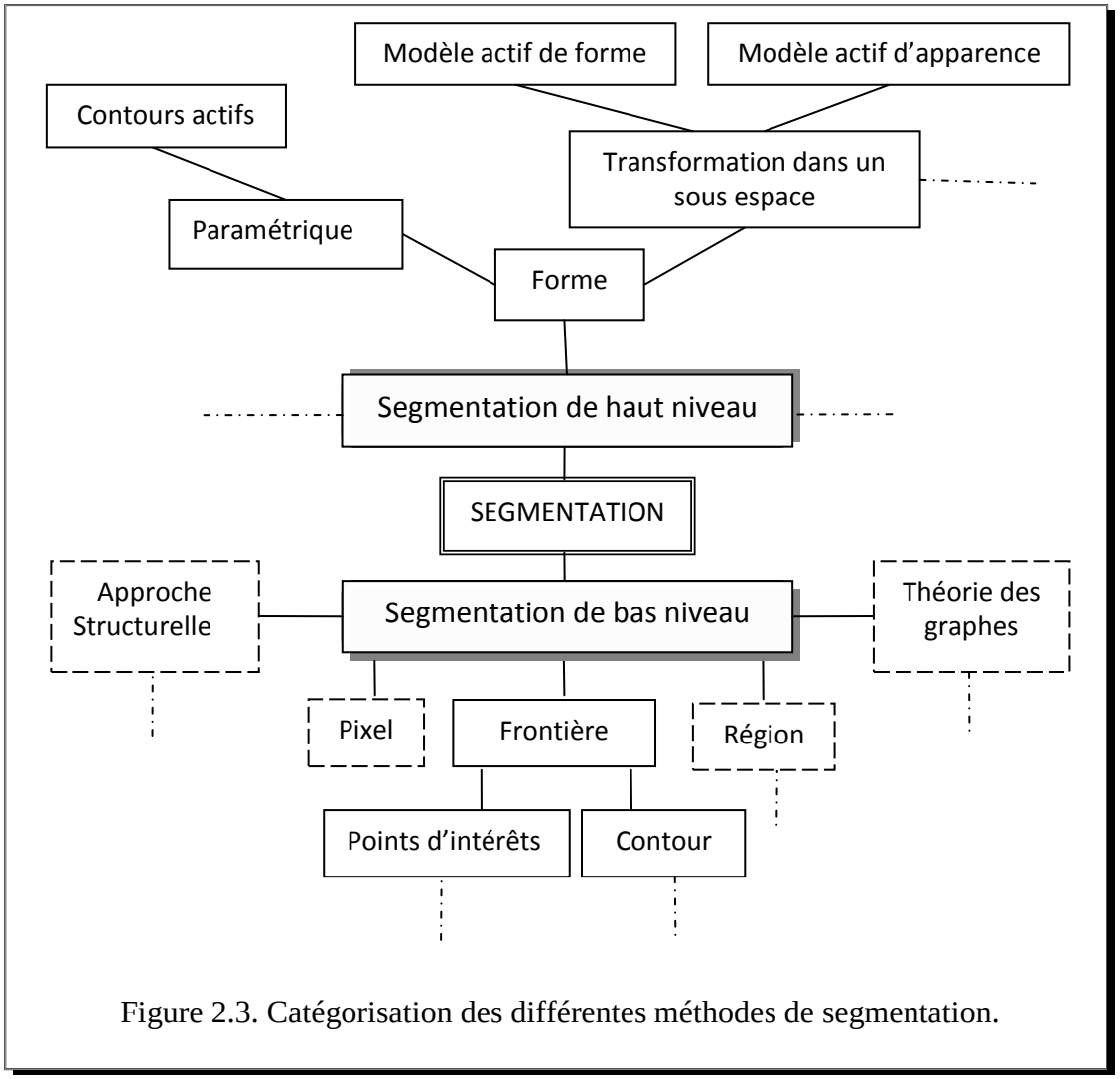
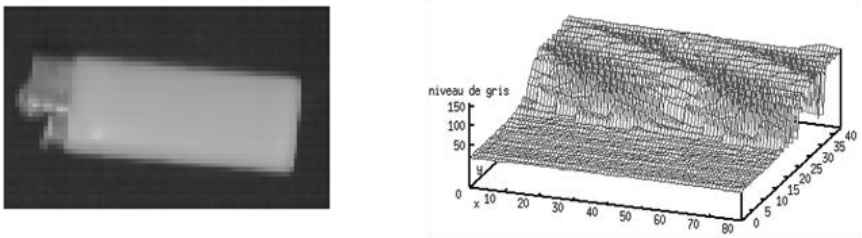


Figure 2.3. Catégorisation des différentes méthodes de segmentation.

L'approche directe consiste à appliquer des opérateurs qui modifient les intensités de l'image. On trouve dans cette catégorie, les méthodes de seuillage et ses différents raffinements, les opérations morpho-mathématiques, et les approches par croissance de région. Ces méthodes conduisent à des transformations de l'image mais ne permettent pas d'interpréter ou de modéliser les informations contenues dans l'image. L'utilisation de connaissances à priori sur le contenu de l'image est exclue, ce qui élimine toutes les méthodes fondées sur une approche descendante ou centrée sur des modèles. L'approche indirecte ou approche par modèles, introduit dans le processus de segmentation une information à priori sur les structures recherchées. Il peut s'agir d'une information sur la forme des objets, leur régularité, leur texture, etc. Les modèles géométriques tels que les modèles déformables font partie de cette catégorie.

La difficulté du problème vient du fait que les données initiales de l'image véhiculent peu et pas d'informations et c'est la vue d'ensemble de tous ces pixels qui permet de dégager une information pertinente. La figure 2.4 montre la représentation d'une fonction image. On n'y discerne pas aussi nettement l'objet que dans l'image originale. Le problème principal en

vision est que l'unité d'observation n'est pas l'unité d'analyse. L'unité de travail est le pixel, dont les propriétés sont la position et l'intensité. Cette connaissance n'amène rien quant à la vision de l'image. Il n'est pas question par exemple de décrire la forme, la position et l'orientation de l'objet. La représentation d'un objet dans l'image correspond au point de vue de cet objet sous une certaine orientation. Son apparition dans l'image est une configuration spatiale des valeurs de plusieurs pixels, et une valeur particulière apporte peu d'information. La segmentation nécessite des algorithmes performants. Il est important de garder à l'esprit que toute méthode ou algorithme de segmentation doit être intégré dans un processus complet d'analyse et fournir les outils de passage à la reconnaissance de l'objet.



(a).Image originale (b). Fonction intensité *I(x, y)*

Figure 2.4. Visualisation d'une image par sa fonction d'intensité [Fiorio 1995].

A l'instar des méthodes générales de segmentation qui définissent la segmentation comme traitement de bas niveau, les autres outils méthodologiques de segmentation ont situé la reconnaissance de formes dans le processus d'interprétation d'images et l'ont focalisé sur des données spécifiques. Le processus de reconnaissance dépend essentiellement des attributs repérés dans l'image. La segmentation définie comme outil central du processus d'interprétation d'images, est replacé dans un autre axe de traitement de haut niveau. Elle est définie comme étant une composante essentielle de toute réponse à une question de reconnaissance de formes. La reconnaissance de forme a besoin d'un modèle et la notion d'apprentissage est au cœur de sa problématique. L'opération de reconnaissance est descendante des modèles vers les données, dans le sens où elle cherche à retrouver dans l'image un modèle d'objet connu. Quelques exemples cités dans [Guigues 2003] définissent les contextes pratiques d'appel à la reconnaissance de formes : interpréter des images médicales et les classer en fonction des pathologies dont elles témoignent, interpréter automatiquement des images satellitaires pour reconnaître les réseaux routiers, etc. Le point commun à ces applications est la localisation de la structure à reconnaître. La segmentation est donc le sous problème qui consiste à délimiter la zone correspondant à la structure recherchée. La tâche de délimitation des régions est alors associée à une capacité de haut niveau, à savoir interpréter les zones extraites, les ranger dans certaines classes sémantiques particulières. C'est sous cette forme que sont posés la plupart des problèmes pratiques d'analyse d'images.

Les quelques outils auxquels nous nous intéressons particulièrement, sont connus sous le nom de modèles déformables. Ils réalisent deux tâches à la fois : la reconnaissance des objets et la délinéation précise de leurs contours [Cousty 2007]. Ils sont fréquemment rencontrés dans la segmentation d'images difficiles en raison de la présence d'un bruit ou d'un manque d'information. L'introduction des connaissances à priori devra améliorer la segmentation pour compléter, corriger et interpréter l'information, comme c'est le cas de la reconnaissance faciale. Le traitement d'images médicales peut se révéler difficile selon les modalités d'acquisition envisagées. Il est donc important d'introduire une connaissance forte sur les objets recherchés. Ces cas difficiles sont fréquents dans l'imagerie où les applications peuvent concerner le traitement d'une grande quantité de données. L'approche de segmentation (figure 2.5) dont il sera question suppose qu'on ne pourra segmenter correctement une image que si elle est a été comprise, c'est-à-dire si l'on est capable de désigner les objets que l'on juge intéressants dans cette image. Avec cette restriction importante, la segmentation d'images consistera à cerner les limites d'objets présents dans le champ d'analyse, objets préalablement désignés et marqués par des procédures adéquates [Ciofolo 2005]. Ces marquages sont utilisés par d'autres procédures chargées elles de segmenter et de fournir avec le maximum de précision, les limites des objets ou des régions en question. Nous citons particulièrement les méthodes de segmentation faisant appel à cette approche.

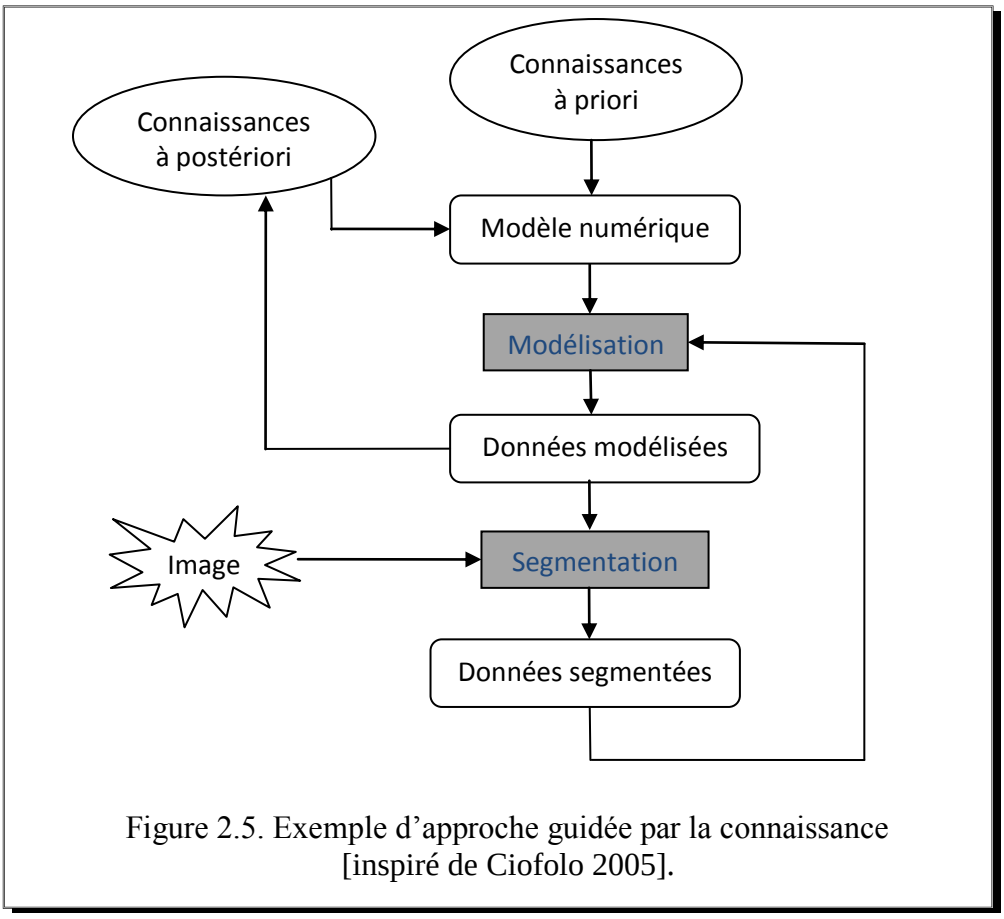


Figure 2.5. Exemple d'approche guidée par la connaissance
[inspiré de Ciofolo 2005].

Les connaissances à priori peuvent correspondre au savoir d'un expert, être simplement dictées par le bon sens, ou provenir d'un système d'imagerie complémentaire ou d'un processus de traitement d'images complémentaire. Les connaissances à posteriori correspondent aux connaissances issues de traitements préalables, générées par exemple par une segmentation préliminaire, pour être ensuite réinjectées dans le processus de segmentation.

Les différentes méthodes se recouvrent largement et afin d'affiner les résultats de segmentation, de nombreux travaux utilisent des approches modulaires qui combinent plusieurs méthodes. Les approches coopératives que ce soit en coopération région-contour [Cocquerez et al. 1995] ou en coopération multi-niveaux associant les traitements bas-niveaux aux données symboliques semblent aussi prometteuses.

2.3.2 Segmentation de bas niveau

De manière non exhaustive, nous proposons un panorama des différentes méthodes de segmentation (classées de bas au haut niveau,) utilisées pour partitionner les images en niveau de gris.

2.3.2.1 Introduction

La segmentation de bas niveau correspond à une description de l'image en primitives simples de l'image, telles que les régions et les contours (figure 2.6). Les régions correspondent aux zones de même homogénéité et les contours marquent les transitions entre les zones homogènes. C'est un domaine très vaste où l'on retrouve de très nombreuses approches.

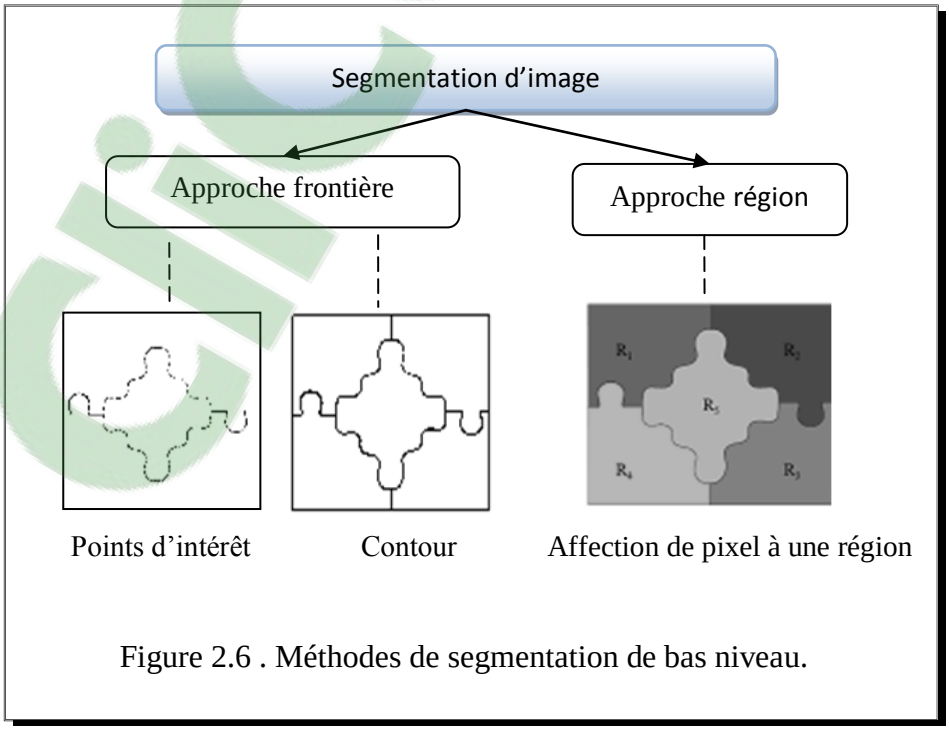


Figure 2.6 . Méthodes de segmentation de bas niveau.

2.3.2.2 Approche contour

L'approche contour (figure 2.7) est une approche non contextuelle qui ignore les relations pouvant exister entre les régions de l'image. Elle comprend toutes les méthodes de détection des contours suivies de techniques de fermeture de contours [Cocquerez et al. 1995]. Elle est essentiellement basée sur des mesures de gradient au sein de l'image. Elle consiste en une étude locale de recherche de discontinuités. Elle exploite le fait, qu'il existe une transition décelable entre deux régions adjacentes. Les contours extraits ne sont généralement pas fermés et/ou continus. Il est souvent nécessaire d'associer une méthode de suivi et/ou de fermeture des contours selon le résultat escompté [Capri 2007].

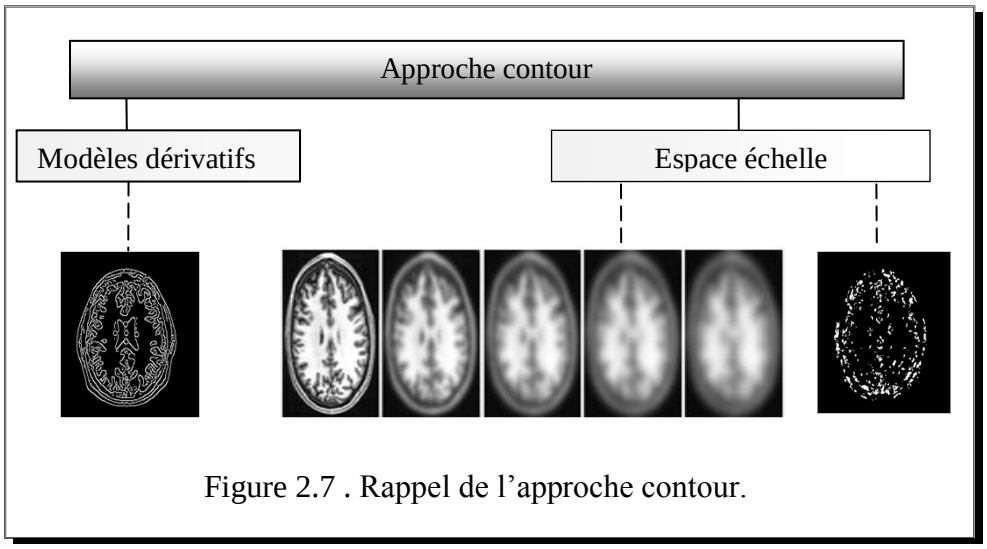


Figure 2.7 . Rappel de l'approche contour.

- **<u>Les Méthodes dérivatives</u>**

Les méthodes dérivatives sont les plus immédiates pour détecter et localiser les variations du signal. Elles supposent que l'image est une fonction scalaire dérivable et bornée en tout point. L'approche consiste à trouver les points frontières entre deux zones homogènes de caractéristiques différentes. Les points frontières ou points de contours sont les points qui présentent une discontinuité importante de luminance par rapport à leurs voisins. Ces discontinuités peuvent être extraites par des filtres de type gradient (Robinson, Kirsh, Prewitt, Sobel, Roberts, etc.) ou Laplacien. Un point de contour est déterminé par le maximum de la norme du gradient (figure 2.8) ou en étudiant le passage par zéro du laplacien. L'avantage des opérateurs de détection de contours est la simplicité d'utilisation. Par contre ils sont très sensibles au bruit et donnent parfois des contours ouverts. Seulement ils ont le mérite d'avoir fondé les bases de la détection de contours. Les méthodes optimales de détection de contours (Canny, Deriche, Marr-Hildrech) calculent le gradient en chaque pixel de l'image et procèdent à l'extraction des points de contour d'un seul pixel d'épaisseur par sélection des maxima locaux des normes de gradient. L'utilisation d'une méthode de seuillage permet de supprimer les pixels isolés ou au contraire de prolonger certaines portions de contours.

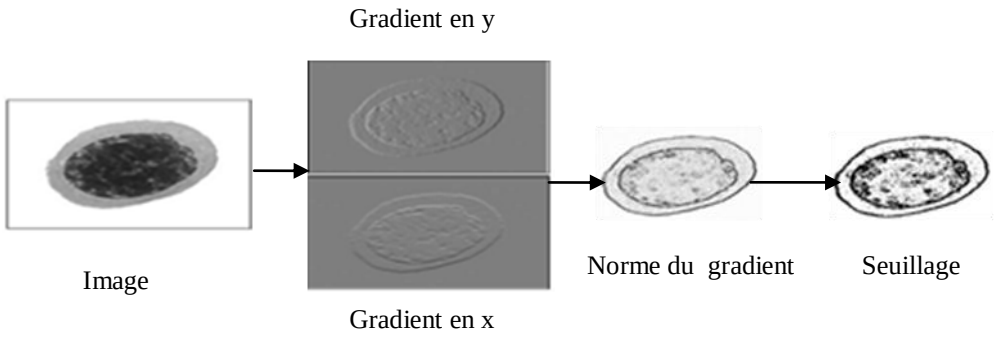


Figure 2.8. Calcul de la norme de gradient pour tout point de l'image et seuil fixé à priori pour la norme du gradient.

La figure 2.9 représente le schéma de fonctionnement d'un détecteur de contour. L'image subit donc toute une série de traitements. L'inconvénient du lissage est l'élimination et le déplacement de certain contours, et la création de faux contours. Une étape supplémentaire peut être ajoutée au schéma de la figure 2.9 : la correction. C'est une étape qui n'est pas toujours présente, et en pratique elle s'exprimera par un dernier filtrage pour éliminer les faux contours [Fiorio 1995].

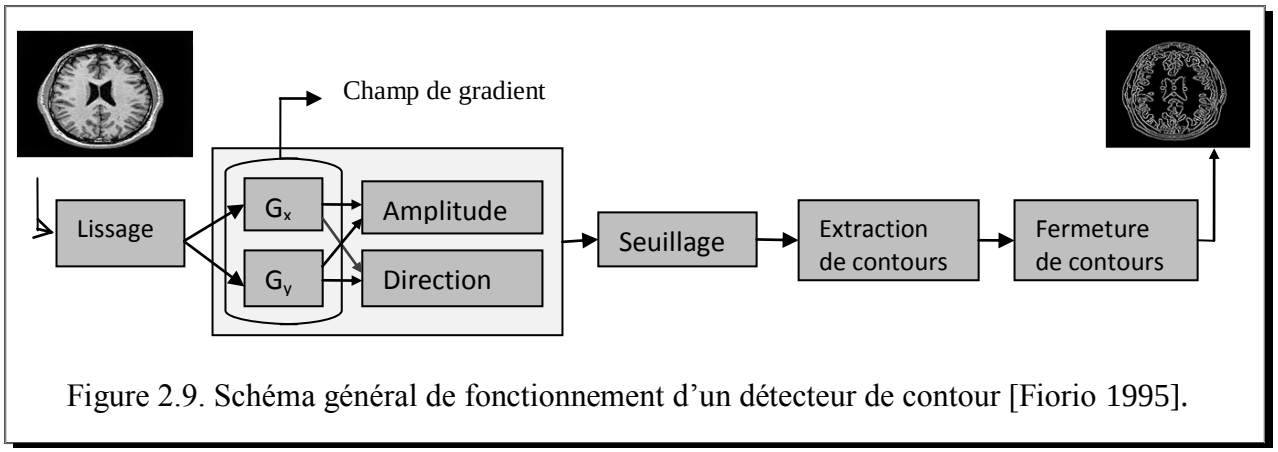


Figure 2.9. Schéma général de fonctionnement d'un détecteur de contour [Fiorio 1995].

Les principales limites des méthodes dérivatives sont :

- Les contours extraits ne correspondent pas souvent aux contours des objets;
- Les méthodes citées utilisent l'information locale définie autour d'un voisinage et il n'y a pas d'information globale;
- Le processus de fermeture des contours produit parfois des discontinuités et génère donc des lacunes dans l'image;
- Il est souvent difficile d'identifier et de classer les contours parasites.

En général Les contours extraits sont la plupart morcelés et peu précis et il faut utiliser des techniques de reconstruction de contours par interpolation ou connaitre à priori la forme de l'objet. Ces techniques sont en général limitées pour traiter des images bruitées et complexes.

- **Détecteurs multi-échelle**

A.Rosenfeld cité dans [Fiorio 1995] a proposé d'utiliser différentes tailles de filtres et de combiner les résultats obtenus. Sur ce principe s'est développé un autre type de filtres dits détecteurs multi-échelle. Les variations de l'image à différents filtrages sont représentées dans un espace échelle. La théorie formelle associée à la notion d'*espace-échelle* définit les structures d'image à différents niveaux de représentation paramétrés par un facteur d'échelle. L'espace échelle associée à une image f(x, y) est une famille de signaux dérivés L(x, y, t) définis à partir d'une convolution de l'image f (x, y) avec un noyau gaussien *g* à l'échelle *t*, tel que :

$$L(x, y, t) = g(x, y, t) * f(x, y)$$

$$g(x, y, t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/2t}$$

Avec la condition initiale que $g(x, y, 0) = f(x, y)$. Le facteur d'échelle est défini de telle sorte qu'il représente la variance de la gaussienne au carrée.

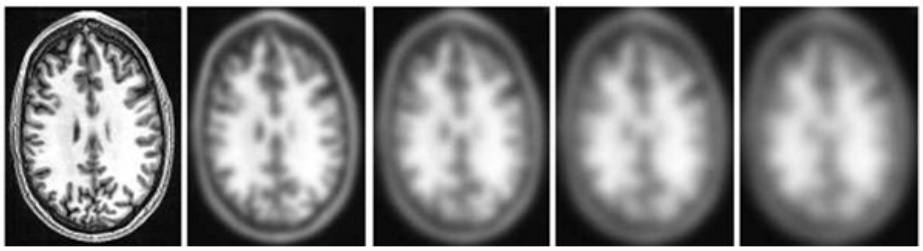


Figure 2.10. Un exemple d'un espace échelle gaussien appliqué à une image cérébrale [Lecœur et al. 2007].

La segmentation établie dans un espace-échelle permet de relier les différentes structures au travers des différentes échelles.

- **Les détecteurs de points d'intérêt**

La détection des points d'intérêt, tout comme la détection des contours, est une étape préliminaire à de nombreux processus dans le domaine de la vision par ordinateur. Les détecteurs de points d'intérêt ont presque le même schéma de détection que celui du détecteur de contour.

Les points d'intérêt sont des points de l'image qui se distinguent des autres. Ils ont la propriété d'être plus identifiable que les autres pixels de l'image et donc ils sont plus faciles à reconnaître, à suivre, ou à mettre en correspondance. Ils correspondent généralement à une discontinuité des niveaux de gris comme le montre la figure 2.11. Ils peuvent également apparaître lors de la modification de la structure, de la texture, ou de la géométrie de l'image.

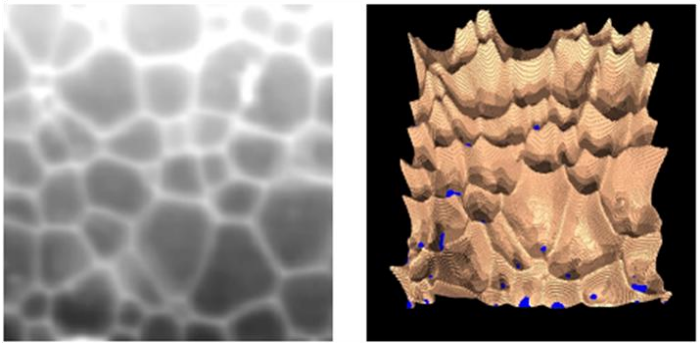
Figure 2.11. Différents types de points d'intérêt : coin simple, jonction en 'V', jonction en 'T', jonction en 'L', jonction en 'damier' [Grand-Brochier 2011].

Le détecteur analyse l'image, d'un point de vue globale, afin d'en extraire les points caractéristiques. L'étude locale du voisinage permet de construire un descripteur propre à chacun des points d'intérêt. Afin d'être le plus invariant possible aux changements d'échelle, les points d'intérêt sont extraits par le biais d'une analyse multi-échelle globale de l'image. Nous citerons le détecteur Harris-Laplace, le Fast-Hessian, et la différence de Gaussienne. La partie description s'appuie sur une exploration locale du point d'intérêt afin de représenter les caractéristiques du voisinage. Le détecteur SIFT (Scale Invariant Feature Transformation) utilise les histogrammes de gradient orientés pour la construction des descripteurs. Le détecteur SURF (Speeded Up Robust Features) fortement influencé par l'approche SIFT, utilise les ondelettes de Haar sur l'image intégrale. Les détecteurs à échelle fixe calculent la réponse avec une taille de fenêtre et une force de lissage gaussien prédéfinis. Nous citons le détecteur Harris, le Moravec, le Kitchen-Rosenfeld, le Susan, le Fast. La thèse de Grand-Brochier [2011] fournit une base détaillée pour la compréhension des approches de détection et de description des points d'intérêt.

- **Les approches structurelles**

Les approches structurelles issues de la morphologie mathématique proposent un autre cadre pour l'approche contour de la segmentation d'images [Beucher 1990, Lecœur et al. 2007]. La morphologique mathématique est une méthodologie ensembliste. Définie dans un espace discret, elle consiste à extraire de la connaissance de l'image à partir des réponses fournies à différents tests de transformation, à travers son interaction avec des formes géométriques simples appelées éléments structurants dont les caractéristiques géométriques sont connues. La dilation et l'érosion sont les opérations morphologiques de base analogues aux opérations mathématiques d'inclusion et d'intersection. Les opérateurs correspondants constituent un ensemble d'outils de reconnaissance des éléments structurants dans l'image. Les réponses positives des deux opérations forment de nouveaux éléments structurants. La

différence symétrique entre l'image dilatée et l'image érodée par le même élément structurant de taille unitaire donne le gradient morphologique de l'image. Une méthodologie classique de segmentation issue de la morphologie mathématique consiste à utiliser ces opérateurs pour la reconnaissance suivis d'algorithmes de délinéation. La ligne de partage des eaux [Beucher 1990, Cousty 2007] est la méthode qui réalise l'étape de délinéation dans une chaîne de segmentation d'image. La LPE fut élaborée à la même époque où S. Beucher et F. Meyer [Beucher 1990] élaborent le gradient morphologique pour déterminer les contours des objets dans des images en niveaux de gris similaires aux opérateurs différentiels. Elle utilise la description de l'image en termes géographiques. Le niveau de gris de chaque pixel représente son altitude. Un pixel noir représente le niveau le plus bas (niveau 0) et un pixel blanc le niveau le plus haut (niveau 255). C'est sur le relief de l'image (figure 2.12), que l'on peut procéder à la recherche de la LPE de façon similaire au domaine de la topographie.



(a) (b)

Figure 2.12. (a) Une image en niveau de gris ; (b) Son relief représenté en 3D.

Une procédure typique de segmentation en morphologie mathématique comprend une étape de filtrage de l'image, la détection des contours, le calcul de la LPE et un post-traitement. La thèse de Beucher [1990] et celle de Cousty [2007] fournissent une base détaillée pour la compréhension des approches structurelles pour la détection des contours.

2.3.2.3 Approche région

L'approche région est l'approche duale de la détection des contours. Elle consiste à rechercher des zones possédant des attributs communs. Chaque pixel de l'image reçoit une étiquette qui indique son appartenance à une région donnée. On déduit une carte des régions de l'image traitée. Chaque objet peut donc être constitué d'un ensemble de régions homogènes. La bibliographie est abondante dans cette catégorie et on recense beaucoup de méthodes. Les principales méthodes citées pour l'approche région sont : les méthodes de croissance de région qui agrègent les pixels voisins selon un critère d'homogénéité et les

méthodes de fusion ou de division de régions qui fusionnent ou divisent les régions en fonction d'un critère donné (figure 2.13).

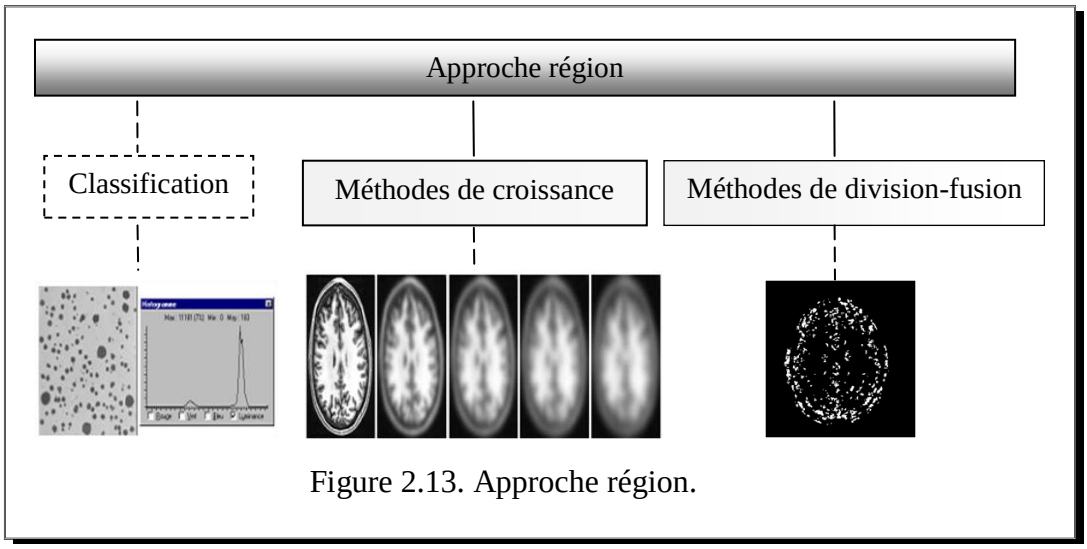


Figure 2.13. Approche région.

- **La croissance de régions**

La croissance de régions est un processus qui s'effectue à partir de pixels initiaux appelés germes sélectionnés de façon aléatoire ou automatique. Lors d'une itération, les pixels adjacents à la région sont traités. Les pixels sont agrégés dans la région s'ils vérifient la condition d'homogénéité. Les pixels non intégrés aux régions peuvent eux même générer de nouvelles régions, ou être assimilés à la région la plus proche à l'aide du calcul d'une mesure de similarité. Le processus s'arrête lorsque tous les pixels adjacents aux régions ont été affectés. Les méthodes de segmentation par croissance de régions peuvent être subdivisées en deux classes : « agrégation de points » et « regroupement itératif d'ensembles de points ». Le premier type d'algorithme associe à chaque pixel un vecteur de propriétés. Deux pixels sont regroupés si les vecteurs sont suffisamment similaires. L'idée du deuxième type d'algorithme est de définir une succession de partitions de l'image par regroupement itératif de régions connexes [Capri 2007].

- **Les méthodes de décomposition et de fusion**

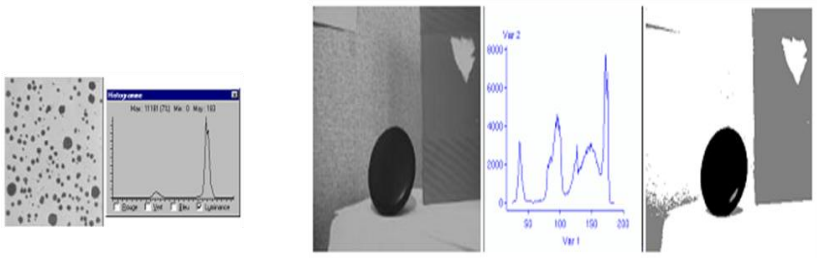
Le principe de ces méthodes consiste en une alternance de phases de division et de fusion de régions jusqu'à optimiser un critère d'homogénéité choisi préalablement. Pour l'étape de fusion, on recherche dans les couples de régions adjacentes quasi similaires, ceux candidats à un possible regroupement. Chaque couple retenu est noté en fonction de l'impact qu'aurait sa fusion sur le critère d'homogénéité global. Les couples les mieux notés sont alors fusionnés. L'étape de décomposition agit de façon opposée. Les régions les moins homogènes sont divisées en régions plus petites. Le résultat final est obtenu lorsque la condition d'arrêt prédéfinie est atteinte ou lorsque les notes attribuées aux couples candidats à la fusion n'évoluent plus significativement. Les méthodes de décomposition les plus utilisées reposent sur le diagramme de Voronoï et l'utilisation d'un arbre quaternaire. Le diagramme de Voronoï

permet de partitionner l'image en polygones. Initialement, il est construit sur un ensemble de germes sélectionnés de façon aléatoire sur l'image via un processus de Poisson. L'application des étapes de décomposition et de fusion permettent la suppression de régions superflues ainsi qu'une décomposition des régions hétérogènes au sens du critère initialement adopté. L'arbre quaternaire consiste à diviser l'image en régions rectangulaires répondant toutes au critère d'homogénéité avant d'appliquer l'étape de fusion. L'arbre est construit itérativement en divisant la région en quatre si elle ne répond pas au critère d'homogénéité. Une fois l'arbre quaternaire établi, les feuilles de l'arbre ayant des caractéristiques similaires, sont regroupées ensemble durant la phase de fusion.

2.3.2.4 Approche pixellaire

- **Le seuillage**

La segmentation par seuillage est une autre catégorie de segmentation de bas niveau utilisée souvent comme une pré-segmentation. Le seuillage permet de segmenter l'image en plusieurs classes en n'utilisant que l'histogramme de l'image (figure 2.14). Une classe est caractérisée par sa distribution en niveaux de gris. A chaque pic de l'histogramme est associée une classe. Il existe de très nombreuses méthodes de seuillage d'un histogramme. La plupart de ces méthodes s'appliquent correctement que si l'histogramme contient des pics séparés. Ces méthodes traitent généralement le cas de la segmentation en deux classes et la segmentation multi-classe n'est que très rarement garantie. Parmi les méthodes de seuillage connues, nous citons : la détection de vallées, la minimisation de variance, le seuillage entropique, la méthode de pourcentage, la maximisation de contraste, etc.



(a) (b)

Figure 2.14. Exemples de seuillage : (a) seuillage en 2 classes (b) seuillage en 4 classes.

Les méthodes de seuillage sont utilisées dans des applications très particulières. Les avantages de ces méthodes sont la simplicité et la rapidité. Cependant, ces méthodes ne sont pas suffisantes pour la plupart des applications où la complexité de l'information contenue

dans l'image ne peut être résumée par l'histogramme des niveaux de gris sans trop de pertes d'information.

- **La classification**

Les méthodes de classification partitionnent l'espace de luminance en utilisant les niveaux de gris présents dans l'image. Elles se basent le plus souvent sur le calcul de l'histogramme de répartition de l'image. Les différents modes de l'histogramme et les vallées correspondantes sont ensuite déterminés. Les classes sont déduites à partir des intervalles entre les vallées. Les méthodes de classification les plus classiques sont la segmentation par seuillage et la méthode des nuées dynamiques. On trouve également d'autres méthodes de classification comme celles citées et schématisées dans [Lecœur et al. 2007].

2.3.3 Segmentation de haut niveau

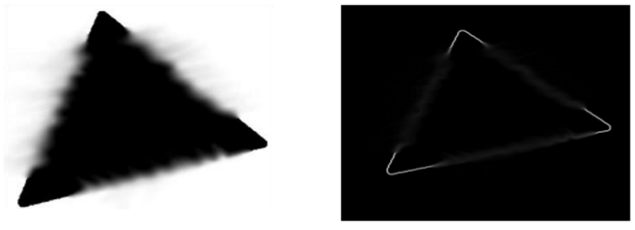
La segmentation de haut niveau est une décomposition de l'image en régions qui ont un sens, les objets de l'image. Elle est un problème fondamental posé dans le domaine de la reconnaissance de formes. Une bonne segmentation est déjà une étape importante pour la reconnaissance. La segmentation de haut niveau s'applique à des entités de nature symbolique associées à une représentation de la réalité extraite de l'image. Elles sont relatives à l'interprétation et à la compréhension de l'image et sont exprimées avec des mots du vocabulaire de l'application. Les relations entre ces zones sont exploitées pour comprendre la scène étudiée.

- **Modèles déformables**

La théorie des modèles déformables a démontré son utilité dans de nombreux contextes de traitement d'images. Sa caractéristique essentielle est que les images d'une classe donnée sont représentées non plus de manière absolue, mais de façon relative, par rapport à une image de référence, le modèle (ou template). L'effort de modélisation porte alors sur la détermination du meilleur état de référence, et sur la représentation des différents processus de déformation qui permettront d'en créer des variations. Une approche statistique permettra alors de discerner les déformations probables, de reconstruire des déformations imparfaitement observées.

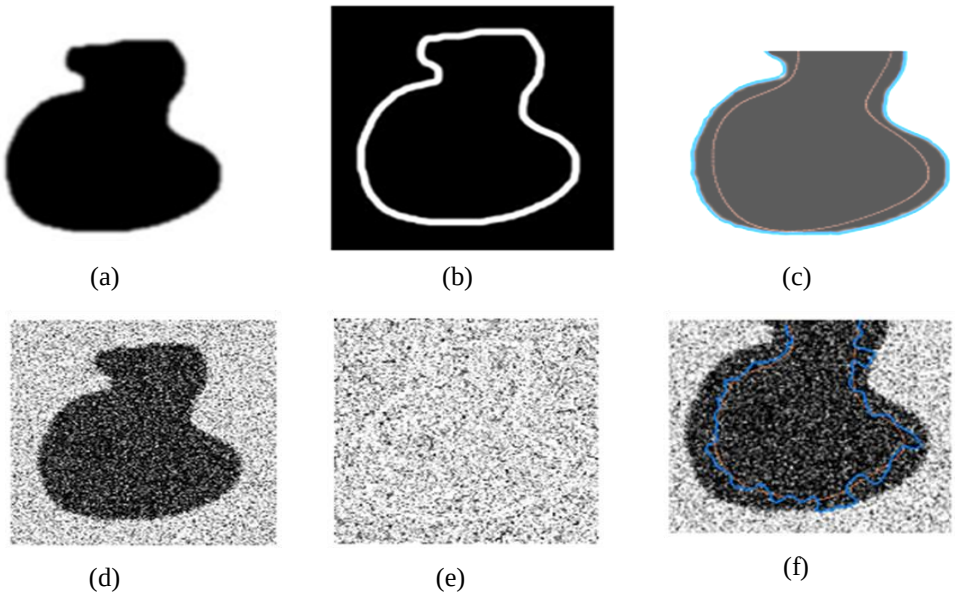
Contrairement aux approches de détection de contours, les modèles déformables incorporent de la connaissance à priori sur la forme de l'objet recherché. Cette connaissance peut être issue d'une base d'apprentissage à partir de laquelle une forme référence ainsi que des modes de déformation peuvent être extraits. Elle peut cependant être plus générale et se limiter à des propriétés de continuité ou de régularité qui caractérisent le contour d'un objet. Dans ce cas on parlera plutôt de contours actifs. Les contours actifs marquent le premier exemple notable des modèles déformables. Le modèle de contours actifs se présente sous la forme d'une courbe fermée ou ouverte dont l'initialisation est située à proximité du contour qu'on veut obtenir et dont l'évolution s'effectue selon un processus itératif de déformation contrôlé par un test de convergence. La segmentation est réalisée à travers un processus de

minimisation d'une énergie calculée en fonction de plusieurs autres énergies. L'énergie interne propre à la forme du contour exprime la régularité de la courbe ou de la surface recherchée, l'énergie externe exprime les contraintes imposées par l'utilisateur, et l'énergie potentielle est imposée par l'image. Les contours actifs sont largement utilisés pour la segmentation d'images médicales en raison de leur capacité à représenter la grande variabilité anatomique et morphologique des structures traitées.



(a) (b)

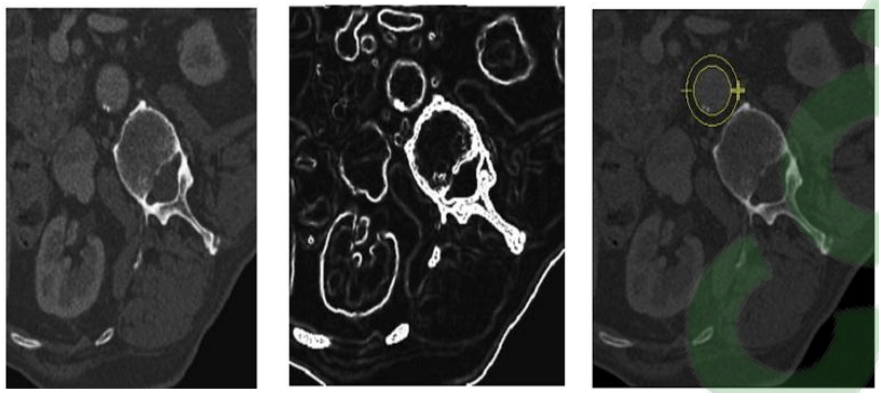
Figure 2.15. Energie externe d'une image calculée par le module du gradient. (a) triangle aux côtés altérés, (b) module du gradient du triangle : les bords ne sont plus visibles.



(a) (b) (c)

(d) (e) (f)

Figure 2.16. (a) image binaire, (b) contours par le filtre de Sobel, (c) contour actif (initial en pointillés orange et final en bleu), (d) image bruitée, (e) contours par le filtre de Sobel, (f) contour actif (initial en pointillés orange et final en bleu.) [Velut 2007].



(a) originale. (b)Sobel. (c) Contour actif initialisé dans l'aorte.

Figure 2.17. Détection de contours d'une image abdominale [Velut 2007].

- **Les modèles déformables discrets**

Les modèles déformables discrets constituent une extension intelligente des snakes. Les formes possibles sont fonction d'un ensemble d'apprentissage, alors que la forme peut être arbitraire pour un snake. La modélisation statistique de formes en traitement d'images a été popularisée depuis l'introduction du modèle « Point Distribution Model » (PDM) de Cootes et al. [1995]. Une base d'apprentissage, peuplée de formes représentées par un ensemble de points, est construite par étiquetage constant de chaque forme et par alignement de toutes les formes. Elle est ensuite analysée statistiquement par une analyse en composantes principales (ACP). Le PDM obtenu modélise alors la forme moyenne et la variabilité de la base d'apprentissage autour de cette forme moyenne. Dans le cas où la base d'apprentissage possède une distribution gaussienne, cette variabilité est censée être représentative de la classe d'objets étudiée. La segmentation est l'application privilégiée du modèle PDM, au sens de la localisation d'objets connus, i.e. appris statistiquement dans des images bidimensionnelles. Cette procédure est connue sous le nom d'« Active Shape Models » (ASM) [Cootes et al. 1995]. Le modèle PDM a été étendu par ses auteurs à la modélisation des variations de forme et de texture. La procédure de segmentation associée a été baptisée « Active Appearance Models » (AAM). Un tel modèle permet de mettre en évidence et de caractériser les variations inhérentes à un type de structures, ce qui est en soi, une information déjà précieuse, particulièrement, dans le contexte des images où la complexité et la variabilité des formes les rendent difficilement appréhendables. Ce type de modèle a suscité un certain nombre de travaux. Les modèles produits ne sont pas toujours des PDMs mais l'esprit demeure. La construction employée permet une infinité de déclinaisons si tant est qu'une représentation de formes soit disponible, une mise en correspondance établie et une analyse statistique choisie. Cependant, l'ACP demeure l'analyse statistique de choix pour dériver de tels modèles, même si on note aujourd'hui l'emploi d'analyses plus variées. De façon générale, les applications envisageables sont l'extraction de caractéristiques, la segmentation, le suivi, la morphométrie, la discrimination, etc.

2.4 Conclusion

Dans ce chapitre, nous avons rappelé la définition de la segmentation d'une image. Nous avons présenté rapidement les grandes familles de méthodes pour réaliser la segmentation. Le nombre de références bibliographiques présentant les méthodes de segmentation de (1999 à 2004) a dépassé les 10000 références [Zouagui 2004] et ce nombre a plus que doublé de (2004 à 2015). Chaque méthode citée par une référence dispose de son propre développement théorique et utilise différentes techniques algorithmiques. Ces techniques sont en plein essor du fait de la puissance des ordinateurs. Reste à dire, que la plupart des algorithmes de segmentation ont été développés pour un usage spécifique. Jusqu'à présent, il n'existe pas un algorithme unique qui puisse être utilisé dans toutes les applications ou pour toutes les catégories d'images. D'un autre côté, le choix d'un algorithme, pour une application particulière, est un problème difficile à aborder. La procédure envisagée généralement est de prendre une méthode existante, de l'améliorer, et de la comparer à des méthodes proches. La difficulté de la segmentation provient de la qualité des images à segmenter qui varie suivant les techniques d'acquisition, des problèmes de recouvrement lorsqu'un objet est partiellement caché par un autre, et aussi de l'absence de critères permettant de juger de la qualité d'une segmentation. La complexité des algorithmes est secondaire, et le problème d'efficacité des algorithmes est moins important. Dans la partie suivante, nous introduisons l'approche de segmentation qui constitue le centre de notre étude. L'approche dont il sera question suppose qu'on ne pourra segmenter correctement une image que si elle est a été comprise, c'est-à-dire si l'on est capable de désigner les objets que l'on juge intéressants dans cette image.

Chapitre 3

Modèles statistiques de forme et d'apparence

Les travaux fondateurs des modèles statistiques de forme et d'apparence pour le traitement d'images sont principalement ceux de Cootes [Cootes et al. 1995], et en particulier le *Point Distribution Model* (PDM), qui utilise une analyse en composantes principales (ACP) sur une population de formes alignées et décrites par des marqueurs. Ils permettent de contrôler à la fois la forme et la texture des images à l'aide d'un nombre réduit de paramètres. Dans le modèle actif de forme, la texture de l'objet n'est prise en compte que sous la forme de profils de niveaux de gris sur quelques segments de droite orthogonaux aux contours de l'objet. Dans le modèle actif d'apparence, la texture de l'objet est représentée par les valeurs de l'ensemble des pixels inclus dans l'enveloppe convexe de la forme de l'objet. Dans ce chapitre, nous présentons les concepts et les techniques fondamentales pour comprendre les modèles statistiques de forme 2D et de texture utilisés dans les modèles actifs d'apparence (AAM).

3.1 Introduction

Comme repris dans de nombreux travaux et mentionné au chapitre précédent, la diversité des techniques de segmentation conduit à la certitude de la-non unicité d'une méthode pour analyser toutes les catégories d'images. Les différentes méthodes citées au chapitre 2 ont bien souvent une efficacité limitée sur un type d'image donné. La classe des modèles déformables offre une alternative intéressante aux méthodes citées précédemment. Les fondements mathématiques des modèles déformables représentent la confluence de la géométrie, de la physique, et de la théorie de l'approximation [Montagnat 1999] [Gastaud 2005] [Velut 2007]. La géométrie sert à représenter les formes d'un objet, la physique impose des contraintes sur les variations possibles des formes en temps et en espace, et la théorie de l'approximation fournit un formalisme des mécanismes d'association des modèles aux données. Les modèles statistiques proposés sont bâtis par apprentissage de la variabilité de la classe d'objets étudiée et dérivés au moyen d'une analyse statistique. Ils ont été mis à contribution à plusieurs reprises. La figure 3.1 décrit le schéma global des modèles statistiques de forme et d'apparence. La première étape constitue la phase d'apprentissage et la construction des modèles et la seconde étant la phase de segmentation.

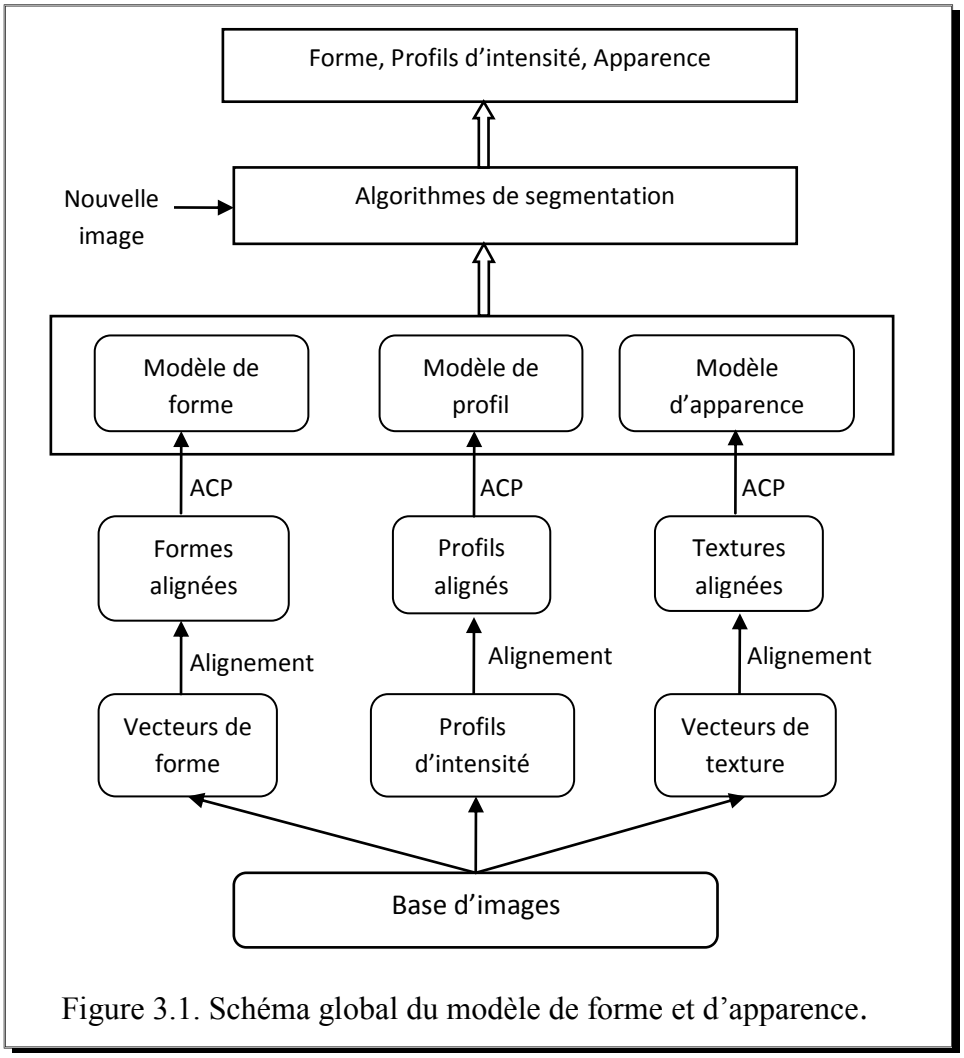


Figure 3.1. Schéma global du modèle de forme et d'apparence.

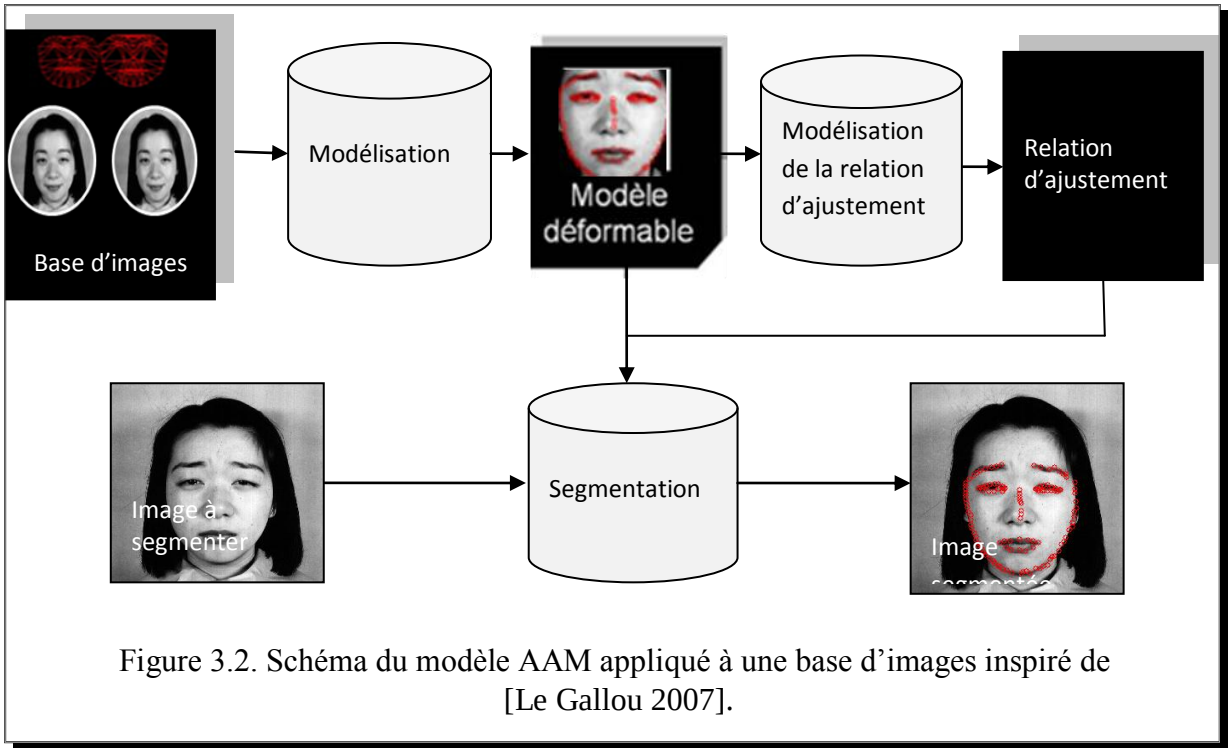


Figure 3.2. Schéma du modèle AAM appliqué à une base d'images inspiré de [Le Gallou 2007].

3.2 Modèle de forme

3.2.1 Notion de forme

Une forme extraite d'une image numérique désigne en général un ensemble de points de contours ou la configuration caractéristique d'une surface. La façon la plus simple de décrire une forme est de relever l'ensemble de pixels, ou de voxels dans le cas tridimensionnel, appartenant à son contour ou à sa surface.

Figure 3.3. Quatre copies d'une même forme « main » [Cootes et al. 1995] [Stegmann 2000].

Une définition plus adaptée est celle proposée par Kendall [Stegmann 2000] : Une forme est une information géométrique qui reste constante lorsque la position, l'échelle, et les effets de rotation sont ignorés. Autrement dit, une forme est une information géométrique invariante aux transformations euclidiennes et affines [Cootes et al.1995].

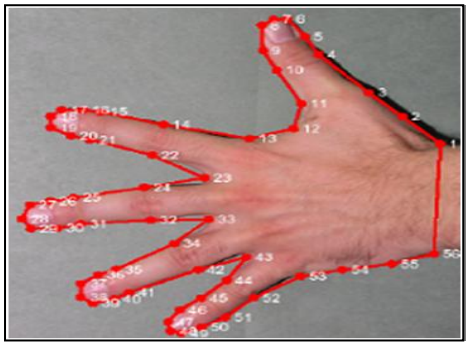


Figure 3.4. Exemple d'une main annotée (56 points) [Cootes et al.1995].

Une description plus structurelle est de localiser un nombre fini de points caractéristiques sur son contour, appelés « landmarks » en anglais. L'ensemble des points doit être adéquat pour représenter la forme générale et détaillé quand c'est nécessaire. Les points caractéristiques peuvent appartenir à trois classes [Cootes et al. 1995], [Stegmann 2000] :

- Les points anatomiques : points définis et affectés par un expert.
- Les points mathématiques : points déterminés en fonction d'une propriété mathématique ou géométrique telles que la courbature ou un point extremum.
- Les pseudo-points : points supplémentaires construits ou obtenus par interpolation des points des deux types précédents.

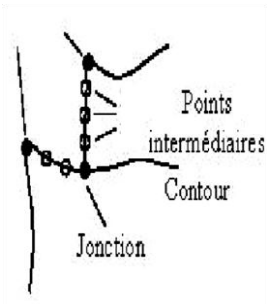


Figure 3.5. Catégories des points caractéristiques.

Les points anatomiques peuvent par exemple correspondre au centre de l'objet et ses extrémités. Les points mathématiques sont typiquement les coins et les jonctions. Les points de la troisième catégorie sont généralement les points équidistants reliant d'autres points. Cette série de points est la plus prédominante et décrit généralement les contours de l'objet.

Figure3.6. Exemple d'une main annotée de 11 points anatomiques et de 17 points supplémentaires.

Figure 3.7. Exemple d'une forme d'un visage en sélectionnant un ensemble de points clés.

Formellement, une forme décrite par *n* points caractéristiques dans un espace de *k* dimensions est décrite par un vecteur de *nk* composantes contenant les coordonnées des points caractéristiques. Pour des images 2D, le processus d'annotation est dans la plupart des cas manuel ou semi-automatique. Dans le cas 3D (x, y, z) et 4D (x, y, z, time) le procédé n'est pas aussi évident.

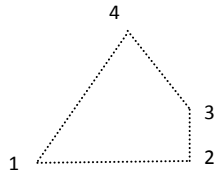
Une forme *2D* (k=2), *X*, composée de *n* points caractéristiques $\{(x_i, y_i), i = \overline{1,n}\}$ est représentée par :

$$X = (x_1, y_1 \dots, x_n, y_n)^T \qquad (1)$$

En pratique, une forme *2D* est représentée, non pas comme un vecteur de dimension $n \times 2$ mais plutôt comme un vecteur de dimension $2n \times 1$, composé de toutes les abscisses suivies de toutes les ordonnées (formule 2) :

$$X = (x_1, \dots, x_n, y_1, \dots y_n)^T \qquad (2)$$

Le résultat de l'annotation d'une base d'image (base d'apprentissage) est un ensemble de vecteurs $\{X_i\}$ où chaque vecteur X_i représente une forme associée à une image i.



Point 1 : (0.0, 0.0)
Point 2 : (3.0, 0.0)
Point 3 : (3.0, 1.0)
Point 4 : (2.0, 2.0)

$X= ((0.0, 3.0, 3.0, 2.0), (0.0, 0.0, 1.0, 2.0))^T$

Figure 3.8. Une forme simple et sa représentation par un vecteur.

3.2.2 Alignement des formes

Chaque forme de la base d'apprentissage étant décrite par un vecteur, l'ensemble de tous ces vecteurs doivent être mis en correspondance point à point, pour étudier les variations des positions de chaque point caractéristique. Plus précisément, il s'agit d'aligner les vecteurs de forme, c'est-à-dire d'ôter les différences en translation, rotation, voire mise à l'échelle, entre les instances à apprendre, pour les exprimer dans un même repère de référence. La représentation de la forme dans un repère de référence est qualifiée de 'représentation canonique'.

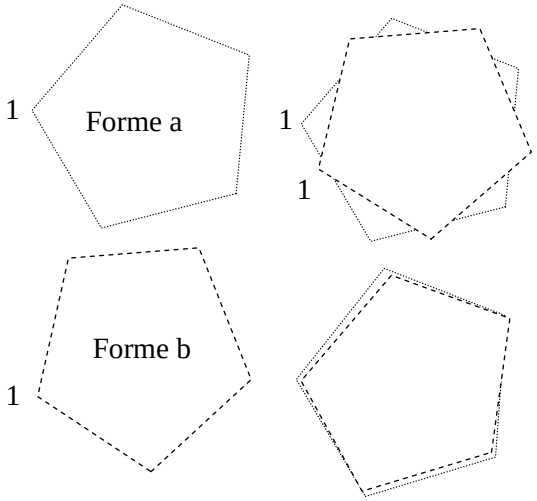


Figure 3.9. Alignement de deux formes a et b.

Le repère de référence découle soit d'un calcul basé sur les axes d'inertie de la structure, soit de la représentation de forme adoptée [Corouge 2003]. Cela facilite généralement l'établissement des correspondances et offre l'avantage de travailler dans un cadre local, propre à la forme, et d'être indépendant de la structure globale de laquelle la forme a été extraite. Le repère de référence désigne l'espace de formes composé de toutes les formes

possibles d'un objet donné [Stegmann 2000]. Formellement, l'espace de formes est la forme orbite des configurations de l'ensemble des n points [Corouge 2003].

La mise en correspondance de formes peut être réalisée de façon explicite en calculant la transformation rigide, voire affine, superposant au mieux les formes à aligner. Elle peut aussi être implicite pour certaines représentations de forme invariantes aux transformations rigides et notamment pour des descriptions relatives à un repère intrinsèque à la structure d'intérêt.

La phase de mise en correspondance est communément dénommée 'calcul de pose'. De nombreuses stratégies sont envisageables pour résoudre le problème de mise en correspondance, parmi elles, l'analyse de Procrustes [Goodall 1991] [Cootes et al. 1995] [Corouge 2003].

- **Analyse de Procrustes et calcul de pose**

L'analyse de Procrustes [Goodall 91] permet d'aligner une population d'apprentissage en supprimant les différences de position, d'orientation, et de mise à l'échelle entre exemplaires. Elle superpose de façon optimale, au sens d'une distance euclidienne les instances de la population d'apprentissage dans l'espace IR^p (p=2 ou 3) [Corouge 2003].

L'analyse de Procrustes ordinaire (APO) consiste à aligner deux formes l'une sur l'autre. Elle détermine la similitude à appliquer à une forme S_1 afin de minimiser sa distance euclidienne à une forme S_2, i.e. à déterminer les paramètres de rotation R, d'homothétie s, et de translation t minimisant :

$$d^2_{APO}(S_1, S_2) = \left\| S_2 - \left(\boldsymbol{s} S_1 R + 1_q t^t \right) \right\|^2 \quad (3)$$

L'analyse de Procrustes généralisée (APG) est appliquée à l'ensemble des instances deux à deux :

$$d^2_{APG}(S_1, \dots, S_m) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \left\| s_i S_i R_{\boldsymbol{i}} + 1_q t^t - \left(s_j S_j R_{\boldsymbol{j}} + 1_k t^t \right) \right\|^2 \quad (4)$$

La similarité utilisée dans la méthode de Procrustes est basée sur la distance euclidienne. Cependant, l'usage de cette distance n'est pas exclusif. Ainsi la distance de Hausdorff peut être employée pour définir les correspondances entre deux ensembles de points. La distance de Hausdorff est une métrique définie sur les ensembles fermés et bornés. Elle indique le degré de ressemblance entre deux ensembles de points S_1 et S_2. Elle s'exprime ainsi :

$$H(S_1, S_2) = \max \left(h(S_1, S_2), h(S_2, S_2) \right) \quad (5)$$

où h (S_1, S_2) est la distance de Hausdorff orientée de S_1 à S_2.

$$h(S_1, S_2) = \max_{s_1 \in S_1} \min_{s_2 \in S_2} \| s_1 - s_2 \| \quad (6)$$

$\|.\|$ est par exemple la norme euclidienne. La distance de Hausdorff mesure la distance du point de S_1 le plus éloigné de l'ensemble des points de S_2 et vice versa.

L'alignement de la base d'apprentissage dans un même système d'axes est réalisé généralement par une analyse de Procrustes légèrement modifiée. Autrement dit, aligner une forme X_i sur une forme X_j consiste à trouver les paramètres de similitude T composés de la translation (t_x, t_y), la rotation θ et le facteur d'échelle s, de telle sorte que la forme X'_j obtenue par rotation, multiplication par le facteur d'échelle et translation de la forme X_j soit la plus proche du vecteur X_i au sens des moindres carrés. La convergence est établie lorsque les formes deviennent stables. Aux points stables, on associe des poids pour les différencier des points instables. Plus le point varie par rapport aux autres, moins il est stable.

La transformation de similarité T appliquée à un point de forme de coordonnées x et y, est :

$$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} s * \cos\theta & s * \sin\theta \\ -s * \sin\theta & s * \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \qquad (7)$$

Aligner deux formes revient donc à :

- Calculer le centroïde de chaque forme ;
- Egaliser les tailles de chaque forme ;
- Faire coïncider les centroïdes des formes ;
- Aligner l'orientation des formes par rotation.

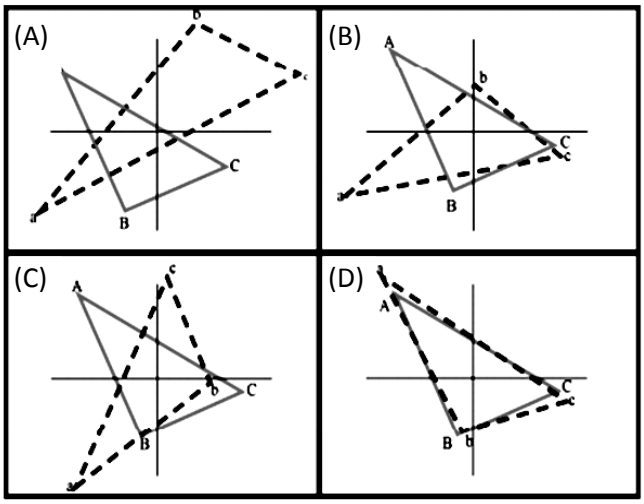


Figure 3.10. Alignement de deux formes triangulaires : (A) Deux triangles. (B) Egalisation des tailles. (C) Translation. (D). Rotation

Le centroïde d'une forme est donné par :

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i\right) \quad (8)$$

La taille d'une forme est calculée selon la norme de Frobenius :

$$S(x) = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \quad (9)$$

ou d'une manière plus simple :

$$S(x) = \sum_{i=1}^{n}\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \quad (10)$$

Etant donnés deux vecteurs X_1 et X_2 composé chacun de n points $(x_{,ij}\ y_{ij})$, la distance de Procrustes après alignement est :

$$P_d^2 = \sum_{i=1}^{n}\left((x_{i1} - x_{i2})^2 + (y_{i1} - {y_{i2}}^2)\right) \quad (11)$$

En pratique, et de manière itérative, chaque forme de la base est alignée sur la forme moyenne courante. Etant données N formes X_i, la forme moyenne $\bar{X}$ est la moyenne de Frechét [Stegmann 2000] donnée par :

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \quad (12)$$

Le schéma suivant décrit l'algorithme permettant l'alignement de toutes les formes de la base d'apprentissage [Cootes et al. 1995] :

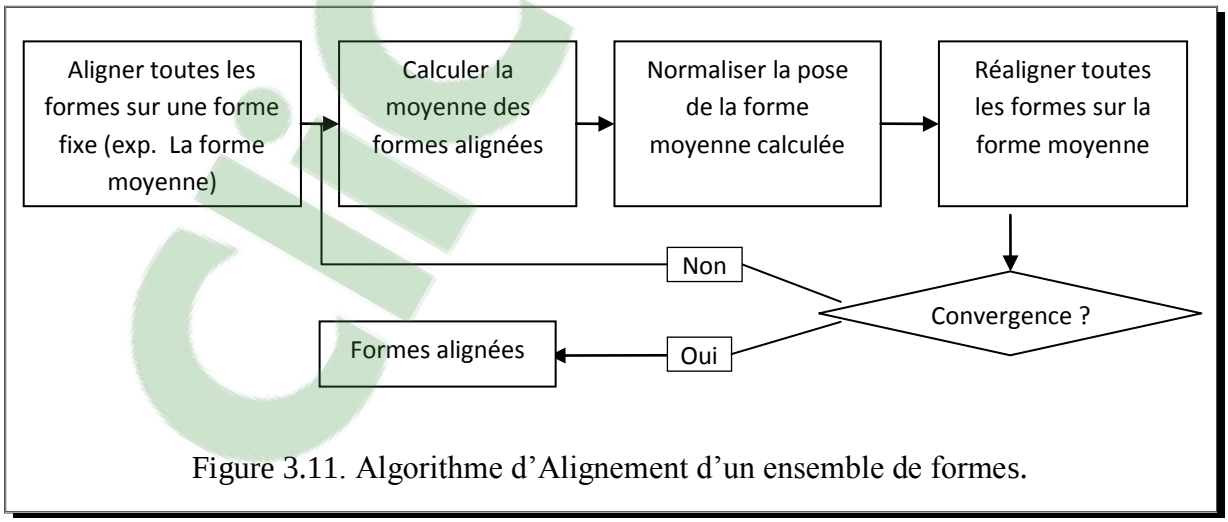


Figure 3.11. Algorithme d'Alignement d'un ensemble de formes.

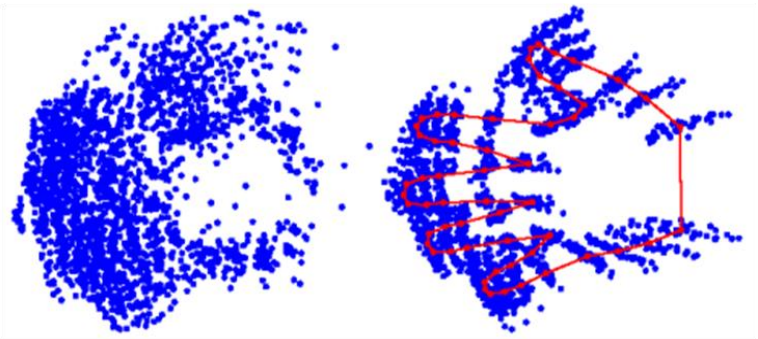


Figure 3.12. Un exemple : à gauche, un ensemble de 40 formes non alignées, à droite l'ensemble des 40 formes alignées avec la moyenne en rouge [Stegmann 2000].

Les formes envisagées comme instances de l'ensemble d'apprentissage sont destinées à être analysées statistiquement. L'ensemble d'apprentissage est considéré comme étant l'ensemble des réalisations d'un vecteur aléatoire X représentant la forme de l'objet d'étude.

3.2.3 Analyse statistique

L'analyse statistique linéaire consiste à transformer les données originales par une transformation linéaire optimisant un critère géométrique ou statistique. Les méthodes les plus répandues sont celles utilisant des statistiques du second ordre comme l'analyse en composantes principales (ACP) introduite par Karl Pearson au début du XX^{e} siècle [Cootes et al. 1995].

- **Principe de l'ACP**

L'ACP est une méthode factorielle d'analyse de données multidimensionnelles qui détermine une décomposition d'un vecteur aléatoire X en composantes décorrélées, orthogonales et ajustant au mieux la distribution de X. Les composantes sont dites « principales » et sont ordonnées par ordre décroissant selon leur degré d'ajustement.

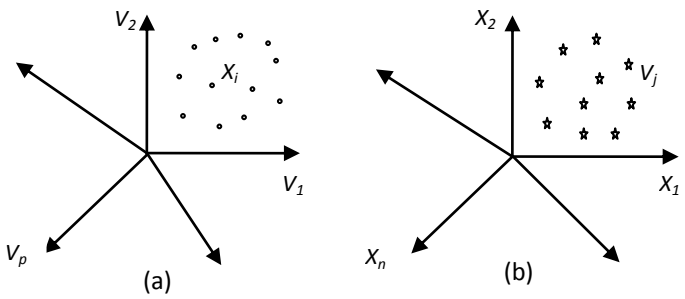


Figure 3.13. (a) Espace d'individus, (b) Espace de variables.

Etant donné un vecteur X de dimension p qui se réalise n fois, les n réalisations X_i forment un nuage de n points dit « espace d'individus » dans IR^p. En faisant de même dans l'espace IR^n, chaque composante V_j de X pourra être représentée par un point dans l'espace affine correspondant, appelé « espace de variables ». L'objectif de l'ACP est de déterminer la base orthogonale ajustant au mieux ces nuages de points selon un critère géométrique. Elle détermine les directions successives de la variance maximale qui correspond à l'optimum du critère géométrique.

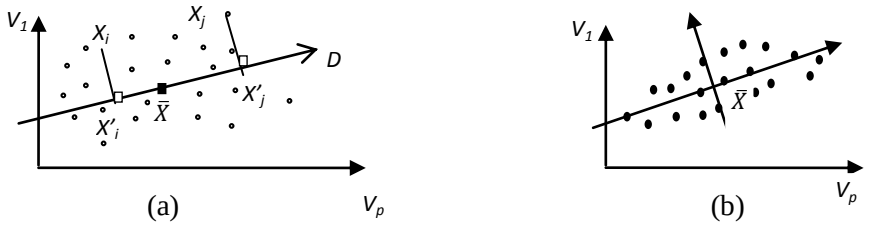


Figure 3.14. (a) Droite d'allongement maximum D. (b) Les deux premières directions principales du nuage de points.

La 1ère composante de la base correspond à la droite d'allongement maximum, D, qui maximise la somme des distances euclidiennes au carré entre les projections sur D de tous les couples d'individus (pris 2 à 2):

$$I_P = \max_D \sum_{i=1}^{n} \sum_{j=1}^{n} d^2\left(X_i', X_j'\right) \quad (13)$$

X_i', et X_j' correspondent respectivement aux projections des points X_i et X_j sur la droite D.

Il est aussi équivalent de maximiser la moyenne des distances entre les projections des points et le centre de gravité $\bar{X}$ du nuage [Corouge 2003] :

$$I_g = \max_D \frac{1}{n} \sum_{i=1}^{n} d^2(X_i', \bar{X}) \quad (14)$$

où

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad (15)$$

La formule (14) peut aussi s'écrire sous la forme suivante :

$$I_g = \max_D \frac{1}{n} \sum_{i=1}^{n} d^2(X_i', \bar{X}) = \max_D \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{p} \left(X_{ij} - \overline{X_{.j}}\right)^2 (16)$$

$$I_g = \max_D \sum_{j=1}^{p} \left[\frac{1}{n} \sum_{i=1}^{n} \left(X_{ij} - \overline{X_{.j}}\right)^2\right] = \sum_{j=1}^{p} Var(V_j) \quad (17)$$

où $Var(V_j)$ est la variance de la variable V_j.

Si le vecteur $d = \left(d_1, \dots, d_p\right)^T$ est le vecteur directeur unitaire de D, alors

$$d^2\left(X_i', \bar{X}\right) = \|X_i' - \bar{X}\|^2 = \|(X_i - \bar{X}).\, d\|^2 \quad (18)$$

$$I_g = \max_D \frac{1}{n} d^t \sum_{i=1}^{n} (X_i - \bar{X})\,(X_i - \bar{X})^t d \quad (19)$$

Le vecteur unitaire *d* solution est le vecteur propre correspondant à la plus grande valeur propre de la matrice de variance-covariance Σ des *p* variables V_j. Sous cette forme, l'inertie totale est égale à la trace de Σ :

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{X})\,(X_i - \bar{X})^t = \frac{1}{n} X X^t \quad (20)$$

Le vecteur *d* définit la première composante principale. Les composantes suivantes sont calculées de façon similaire sous contrainte d'orthogonalité avec les composantes précédemment déterminées. Le but est de trouver un nouveau repère $q_1, \dots, q_p$ tel que la quantité d'information expliquée par q_1 soit maximale, puis celle expliquée par q_2, etc.

De manière générale, la base orthogonale ajustant au mieux le nuage des n points dans IR^p, au sens des moindres carrés, est définie par le centre de gravité de ce nuage et par les vecteurs propres de la matrice de covariance ordonnés par valeur décroissante des valeurs propres associées.

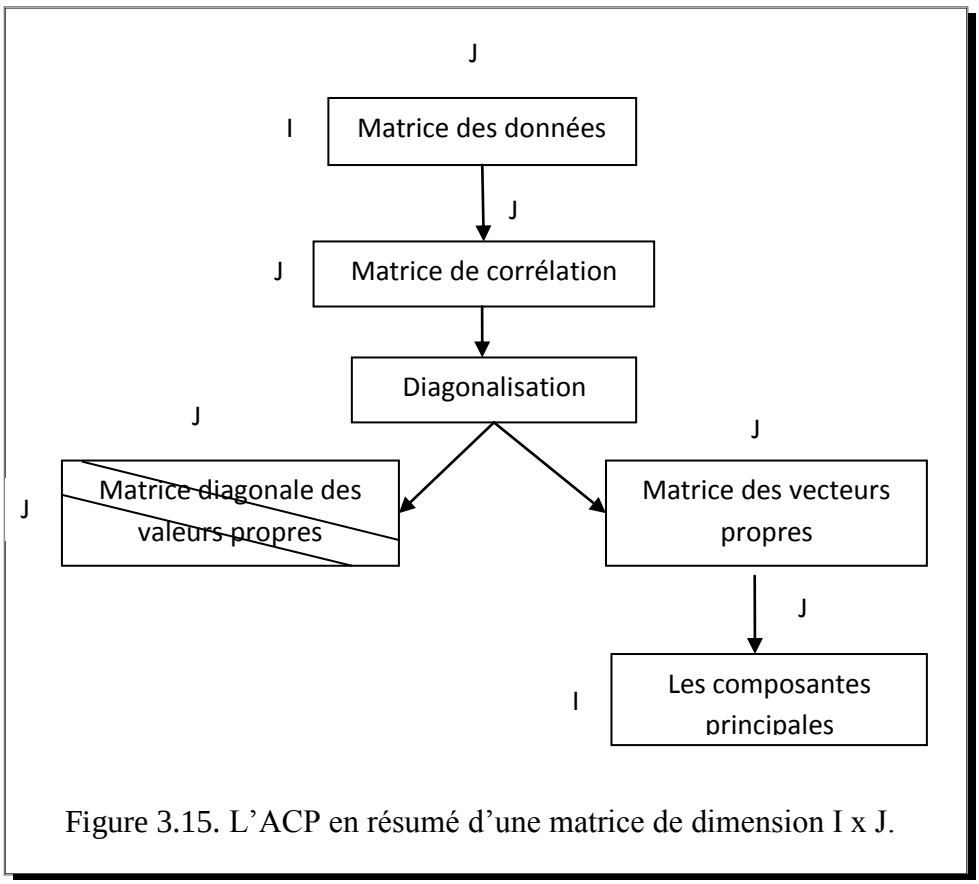


Figure 3.15. L'ACP en résumé d'une matrice de dimension I x J.

La base est obtenue par diagonalisation de la matrice de variance-covariance Σ afin d'obtenir la matrice Φ de ses vecteurs propres [Cootes et al. 1995] [Stegmann 2000] [Corouge 2003] [Ciofolo 2005] :

$$\Sigma\Phi = \Phi\Lambda \qquad (21)$$

où

$$\Phi = (\Phi_1, ..., \Phi_p) \qquad \text{et} \qquad \Lambda = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_p \end{pmatrix}$$

Chaque vecteur propre Φ_i est associé à la valeur propre λ_i. Les valeurs λ_i, $i=1, \ldots p$, sont positives et ordonnées par valeur décroissante telles que :

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$$

La matrice de covariance étant symétrique définie positive, ses vecteurs propres forment une base orthonormale ($\Phi^t \Phi = I_p$). Dans cette nouvelle base, une observation X s'écrit :

$$X = \bar{X} + \Phi b \qquad (22)$$

Le vecteur aléatoire b est de moyenne nulle et de covariance Λ :

$$\bar{b} = \frac{1}{n}\sum_{i=1}^{n} b_i = \frac{1}{n}\sum_{i=1}^{n} \Phi^t (X_i - \bar{X}) = \frac{1}{n}\Phi^t \sum_{i=1}^{n} (X_i - \bar{X}) = 0 \qquad (23)$$

$$Cov(b) = \frac{1}{n}\sum_{i=1}^{n} b_i b_i^t = \frac{1}{n}\sum_{i=1}^{n} \Phi^t (X_i - \bar{X})(X_i - \bar{X})^t \Phi = \Phi^t \boldsymbol{C} \Phi = \Lambda \qquad (24)$$

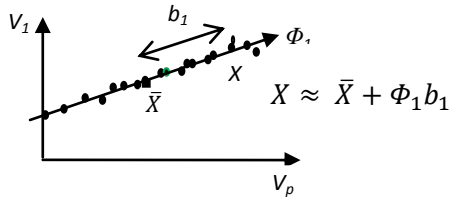


Figure 3.16. Exemple de réduction de dimensionnalité.

La décomposition obtenue permet de réaliser une approximation modale des observations et de quantifier l'erreur réalisée. Il suffit de sélectionner un certain nombre m, $m \leq p$, de modes dans la base modale pour reconstruire une observation X :

$$X \approx \bar{X} + \Phi_m b_m \qquad (25)$$

Φ_m est une sous matrice $p \times$ *de* m de Φ contenant les m vecteurs propres Φ_i sélectionnés et b_m représente une observation quelconque dans l'espace m-dimensionnel défini par les m composantes principales retenues. Cela correspond à une réduction de la dimensionnalité et produit une représentation plus compacte des observations. La non sélection du mode *i*

engendre une erreur usuellement quantifiée par le taux de variance perdue sur la variance totale λ_T [Corouge 2003] :

$$\frac{\lambda_i}{\lambda_T} \text{ où } \lambda_T = \sum_{i=1}^{p} \lambda_i = Trace(\Lambda)$$

La qualité de la reconstruction est mesurée par le taux de variance expliquée par les *m* composantes retenues :

$$\tau = \frac{\sum_{i=1}^{m} \lambda_i}{\lambda_T} \quad (26)$$

τ est appelé « taux d'inertie ». Il est une mesure globale de la qualité de représentation sur la base modale. Si *m* est égal à *p*, la reconstruction est parfaite.

Discussion

D'autres méthodes d'analyse sont envisageables. On peut en particulier citer l'analyse en composantes indépendantes (ACI) [Ciofolo 2005] qui s'est largement répandue dans la communauté du traitement du signal et des images. Contrairement à l'ACP qui utilise des statistiques d'ordre 2, l'analyse en composantes indépendantes met en jeu des statistiques d'ordre supérieur, dans le but de fournir une décomposition d'un vecteur aléatoire en composantes statistiquement indépendantes. Cependant, en pratique l'ACP fournit des résultats tout à fait intéressants et souvent satisfaisants.

3.2.4 Dérivation du modèle statistique de forme

L'application de l'analyse en composantes principales aux vecteurs de forme X_i alignés, permet de représenter les variations des vecteurs d'une manière concise, en fonction du vecteur de forme moyen $\bar{X}$ et les principales directions de variation Φ. Chaque forme représentée par *n* marqueurs peut être vue comme un point dans un espace de dimension *2n*. On suppose alors que ce nuage de points est localisé dans une région de l'espace qui correspond au domaine des formes autorisées. La forme moyenne $\bar{X}$ est donc le centre de gravité de ce nuage et ses axes principaux Φ sont estimés à l'aide de l'ACP, ce qui permet d'obtenir une représentation de la variabilité de la population d'apprentissage autour de cette forme moyenne, selon les modes principaux de déformation *b*. Dans le cas où la population d'apprentissage possède une distribution gaussienne [Cootes et al. 1995], cette variabilité est censée être représentative de la classe d'objets étudiée :

$$X = \bar{X} + \Phi b$$

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$$

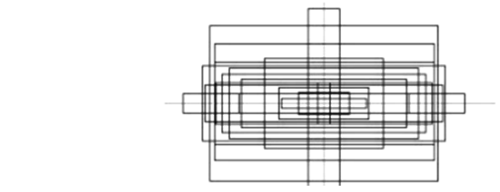
$$\Phi = [\Phi_1, \dots, \Phi_{2n}]$$

$$b = \Phi^T (X - \bar{X})$$

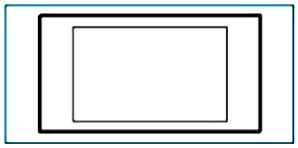
La distribution de *b* caractérise le domaine d'admissibilité de l'ensemble d'apprentissage [Corouge 2003, Ciofolo 2005]. De nouvelles instances, conformes aux observations déjà apprises, peuvent être synthétisées en choisissant le paramètre *b* dans des limites admissibles [Cootes et al. 1995]. L'intervalle de variation est :

$$-3\sqrt{\lambda_i} \leq b_i \leq +3\sqrt{\lambda_i}$$

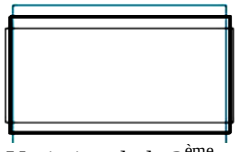
Les variations du paramètre *b* permettent de faire évoluer une forme de la moyenne vers des formes plus extrêmes.



Quelques formes simples : des rectangles symétriques



Variation de la 1ère composante principale



Variation de la 2ème composante principale

$$\bar{X} = \begin{pmatrix} 23 & 12 \\ -23 & 12 \\ -23 & -12 \\ 23 & -12 \end{pmatrix} + (b_0) \begin{pmatrix} 12 & 4 \\ -12 & 4 \\ -12 & -4 \\ -12 & -4 \end{pmatrix} + b_1 \begin{pmatrix} -4 & 12 \\ 4 & 12 \\ 4 & -12 \\ -4 & -12 \end{pmatrix} + \cdots$$

Figure 3.17. Exemple d'un modèle de forme [Millborrow 2008].

Une approximation modale peut être réalisée en conservant uniquement les *t* premiers modes. Une forme approchée s'écrit alors :

$$X = \bar{X} + \Phi_t b_t$$

$$b_t = \Phi_t (X - \bar{X})$$

Figure 3.18. (a) Exemple de formes de visage ;
(b) Effets de variation des trois premiers modes [Cootes 1995].

La qualité de la représentation obtenue sur la base tronquée peut être évaluée par le pourcentage de variance p relatif aux *t* premiers modes. Il convient selon [Cootes et al. 1995] et [Stegmann 2000] de choisir *t* et *p* tels que :

$$\sum_{i=1}^{t} \lambda_i \geq \frac{p}{100} \sum_{i=1}^{2n} \lambda_i$$

Discussion

Cootes et al. [1995] [1998a] [1998b] confirment que si les données de la base d'apprentissage sont gaussiennes, elles sont complètement caractérisées par leurs statistiques d'ordre 2, et dans ce cas, l'analyse en composantes principales semble appropriée. En revanche, si les données sont non gaussiennes, la base de décomposition obtenue n'est pas optimale. Ciofolo [2005] propose d'utiliser l'analyse en composantes indépendantes. Cependant, en pratique, bien que le caractère gaussien des données soit rarement testé et donc garanti, les résultats fournis par l'analyse statistique ACP satisfont raisonnablement les critères de spécification des modèles de forme [Cootes et al. 1995] [Cootes et al. 1998a] [Cootes et al. 1998b]. Dans le cas des structures de forme complexes, la variabilité des formes n'est pas toujours bien connue et il est difficile de vérifier si le modèle est réaliste ou non. L'intérêt de l'ACP réside alors dans la connaissance des conséquences de la violation de l'hypothèse gaussienne.

3.2.5 Caractéristiques du modèle

Le modèle de forme obtenu par apprentissage doit satisfaire certaines propriétés [Ciofolo 2005]. Il est généralement examiné en termes de compacité, de spécificité et de capacité de généralisation :

- La compacité du modèle est sa capacité à représenter une forme avec un nombre restreint de modes, sans perte importante d'information. Le pourcentage d'inertie totale expliquée,τ, fournit une mesure globale de la qualité numérique de la reconstitution selon le

nombre de modes retenus, i.e. de la qualité de la représentation dans l'espace défini par les composantes principales retenues. Ainsi le pourcentage d'inertie totale expliquée est considéré comme une mesure de compacité en ce sens.

$$\tau = \frac{\sum_{i=1}^{t} \lambda_i}{\lambda_T} \times 100 \text{ où } \lambda_T = \sum_{i=1}^{2n} \lambda_i$$

L'étude de la stabilité du taux d'inertie en fonction du cardinal des populations d'apprentissage analysées, permet de déduire la corrélation entre la compacité de la représentation et leur cardinal. Un test de « convergence » consiste alors à déterminer si la dimension de la base modale réalisant un taux d'inertie donné se stabilise lorsque le nombre d'exemplaires augmente [Corouge 2003]. Il est évident de dire que dans une base d'approximation modale compacte, les approximations obtenues sont fortement dépendantes des caractéristiques des populations d'apprentissage analysées [Cootes et al. 1995] [Corouge 2003] [Ciofolo 2005] [Stegmann 2000].

- La spécificité du modèle est son aptitude à représenter exclusivement des formes appartenant à la classe d'intérêt étudiée. Elle est évaluée en observant si les formes synthétisées à partir de la variabilité apprise sont réalistes et conformes à la classe étudiée. Ceci est difficilement mesurable et quantifiable [Corouge 2003]. Théoriquement, lorsque la distribution des exemples étudiés est gaussienne et suffisamment pleine, les instances générées sont en conformité avec les instances apprises. Dans le cas contraire, la spécificité n'est plus théoriquement garantie.

- La capacité de généralisation du modèle est son aptitude à représenter des formes non apprises mais appartenant à la classe étudiée, en calculant la distance moyenne entre la forme reconstruite et la forme exclue de la base d'apprentissage en fonction du nombre de modes choisis. Un test de « leave-one-out » [Cootes et al. 1995] [Corouge 2003] consiste alors à exclure tour à tour chaque instance de la population et à la reconstituer sur la base modale issue de la population initiale privée de cette instance. La capacité de généralisation est mesurée au moyen de l'erreur de reconstitution. L'erreur de reconstitution totale est obtenue en moyennant les erreurs calculées en excluant tour à tour chacune des instances de la base.

Discussion

L'analyse statistique en composantes principales fournit des résultats tout à fait intéressants et souvent satisfaisants en termes de compacité, capacité de généralisation et de spécificité même si ce dernier critère s'avère difficile à évaluer dans le cas de formes complexes. Lorsque la distribution n'est pas linéaire, l'analyse en composantes principales ne peut pas rendre compte de façon optimale de la structure des données. Dans ce cas, on peut avoir recours à des statistiques non linéaires permettant de fournir des axes principaux non linéaires ou d'estimer la densité de probabilité de X, comme l'analyse en composantes principales à fonctions de base à noyaux, dite « Kernel-PCA ». Elle est une forme d'analyse ACP non linéaire, dont l'idée est de transformer les données originales vers un espace dans

lequel leur distribution sera de nouveau considérée et de leur appliquer alors une ACP classique.

3.2.6 Pose du modèle

Les points du modèle (équation 22) sont originalement définis dans le repère du modèle. Afin de pouvoir contrôler la pose du modèle dans l'image, un paramètre *T* dit « de pose » est rajouté au modèle. T est composé des paramètres d'échelle *s*, de rotation θ et de translation t.

$$T= [s, \theta, t]$$

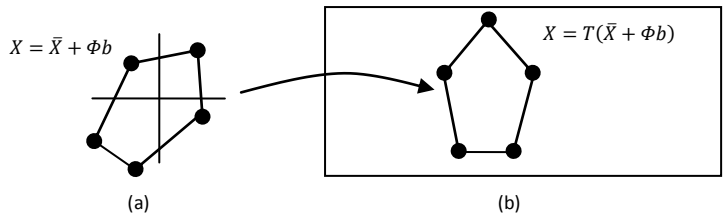


Figure 3.19. (a) Repère du modèle ; (b) Le modèle dans l'image.

La position des points du modèle, X, dans l'image, correspond à :

$$X = T_{t,s,\theta}(\bar{X} + \Phi b)$$

$T_{X,s,\theta}$ est la fonction de transformation euclidienne qui exécute une rotation, une mise à l'échelle, et une translation de tous les points du modèle par les paramètres (θ, s, t).

ou simplement :

$$X = T_{s,\theta}(\bar{X} + \Phi b) + t$$

$T_{s,\theta}$ appliqué au point (x, y) correspond à :

$$T_{s,\theta}\begin{pmatrix} x \\ y \end{pmatrix} = s * \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

3.2.7 Mise en correspondance

La représentation d'un exemple *X* par le modèle de forme consiste à trouver les paramètres de forme *b* et de pose *T* qui minimisent la distance *d* définie par :

$$d = distance\big(X, T(\bar{X} + \Phi b)\big)$$

T est la transformation de similarité qui fait correspondre l'espace de représentation du modèle à celui de l'image. La méthode de minimisation proposée par Cootes [Cootes et

al.1995] est une procédure de mise en correspondance appelée «*Matching model points to target points* ».

La méthode se résume comme suit :

1- Initialiser les paramètres de pose, b, à zéro (la forme moyenne);
2- Générer une instance du modèle en utilisant $X = \bar{X} + \Phi b$;
3- Trouver les paramètres de pose (θ, s, t) qui alignent les points du modèle au vecteur Y;
4- Projeter Y dans le repère du modèle : y= $T_{s,\theta,t}^{-1}(Y)$;
5- Mettre à l'échelle Y' en calculant $y' = y/(y.\bar{X})$;
6- Mettre à jour les paramètres du modèle $b = \Phi^t(y' - \bar{X})$;
7- Reprendre en 2 jusqu'à convergence.

Discussion

Lors de la construction d'un modèle statistique de forme, l'étape la plus sujette à caution est certainement celle de la mise en correspondance. Elle est cruciale car elle décide des relations entre les instances de la base et les instances du modèle, et introduit un biais dans le modèle lorsqu'elle est fausse. De nombreux travaux sont destinés à sa résolution, ce qui justifie la difficulté de la tâche. La mise en correspondance pose un problème à part : ce ne sont pas les techniques qui manquent, mais il arrive souvent que l'on ne sache pas comment deux objets devraient être mis en correspondance, et allant plus loin, s'il est même pertinent de les faire correspondre [Corouge 2003].

Elle est également cruciale dans la phase de recherche qui permet au modèle déformable de s'ajuster aux objets dans de nouvelles images et de trouver les paramètres d'ajustement qui déterminent la position et la forme de l'objet cible dans l'image. Cootes et al. [1995] [1998a] considèrent le problème de mise en correspondance comme un problème d'optimisation d'une fonction coût mesurant la correspondance établie entre le modèle et l'objet de l'image. Les paramètres qui ajustent correctement le modèle à l'image est l'ensemble des paramètres qui optimisent la fonction de correspondance. La mesure de correspondance prise en compte est la distance point à point entre le modèle et l'objet.

3.2.8 Profils des points caractéristiques

Un profil d'un point caractéristique représente l'ensemble des intensités des pixels situés au voisinage du point. Pratiquement, pour chaque point caractéristique *j* d'un objet *i*, on extrait le vecteur des niveaux de gris g_{ij}, de p pixels échantillonnés selon la normale au contour passant par le point (figure 3.20) :

$$g_{ij} = \left(g_{ij}^1, g_{ij}^2, \dots, g_{ij}^p\right)$$

Le profil du point caractéristique *j* est constitué de la dérivée normalisée des niveaux de gris :

$$y_{ij} = \frac{dg_{ij}}{\sum_k dg_{ij}^k}$$

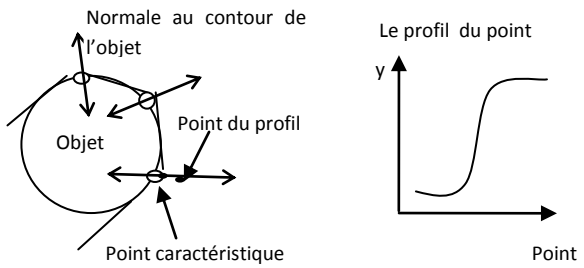


Figure 3.20. Profils des points caractéristiques.

Le choix de la dérivée normalisée permet de donner une invariance par rapport aux perturbations linéaires des conditions d'éclairage. Chaque point caractéristique est représenté donc par l'ensemble de tous ses profils $\{y_{ij}\}$ dans l'ensemble d'apprentissage. Pour résumer ces informations, on calcule la moyenne $\overline{y_{ij}}$ et la matrice de covariance $\Sigma_{y_{ij}}$ de ces vecteurs. Par exemple, si la longueur du profil est 7, $\overline{y_{ij}}$ est un vecteur de taille 7 et $\Sigma_{y_{ij}}$ est une matrice 7×7. Pour un ensemble de 68 points de marquage, nous avons ainsi 68 $\overline{y_{ij}}$ et 68 matrices $\Sigma_{y_{ij}}$.

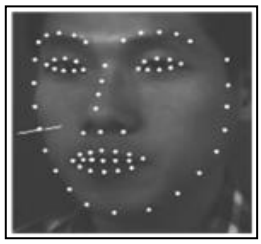


Figure 3.21. Points de repère et la normale en un point spécifique.

3.3 Modèle actif de forme

Le modèle actif de forme ASM (Active Shape Model) est la désignation de l'algorithme de mise en correspondance (Matching) dont le principe est de rechercher dans les images traitées de nouvelles formes conformes au modèle donné par le modèle de distribution des points, le PDM, autrement dit, de trouver les paramètres de pose et de forme du modèle qui décrivent le mieux les formes présentes dans les images. Toute instance X du modèle peut être créée en utilisant l'équation $X = T_{X_t,s,\theta}(\bar{X} + \Phi b)$.

L'ASM est donc un processus de recherche itératif qui consiste à déplacer les marqueurs du modèle jusqu'aux points qui leur correspondent le mieux sur l'image, en cherchant les forts gradients le long des profils d'intensité sur la normale à la frontière du modèle.

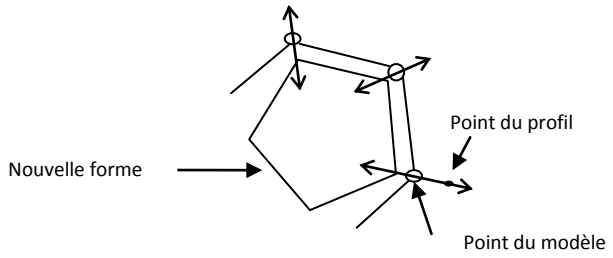


Figure 3.22. Modèle d'apparence local utilisé dans la procédure de recherche.

L'algorithme du modèle actif de forme se présente comme suit :

1. Examiner la région de l'image autour de chaque point caractéristique pour déterminer une meilleure position pour ce point, en utilisant son profil,
2. Mettre à jour les paramètres de forme et de pose,
3. Appliquer les contraintes d'admissibilité aux paramètres de forme ($|b_i| \leq 3\sqrt{\lambda_i}$)
4. Répéter jusqu'à convergence.

Durant le processus de recherche, pour chaque point caractéristique, on crée plusieurs profils appelés profils de recherche. L'image traitée est échantillonnée au voisinage du point. Pour chaque point caractéristique, on cherche la meilleure position le long de la normale passant par le point considéré, ayant un profil plus proche de son profil moyen. On échantillonne l'image traitée le long de cette normale, en prenant un nombre de pixels supérieur à celui du point caractéristique et on balaye le profil de recherche.

La distance entre un profil moyen et un profil de recherche est la distance de Mahalanobis donnée par :

$$f(y_s) = (y_s - \bar{y})^t \Sigma_y^{-1} (y_s - \bar{y})$$

où y_s est le profil de recherche, $\bar{y}$ est le profil moyen et Σ est la matrice de covariance des profils.

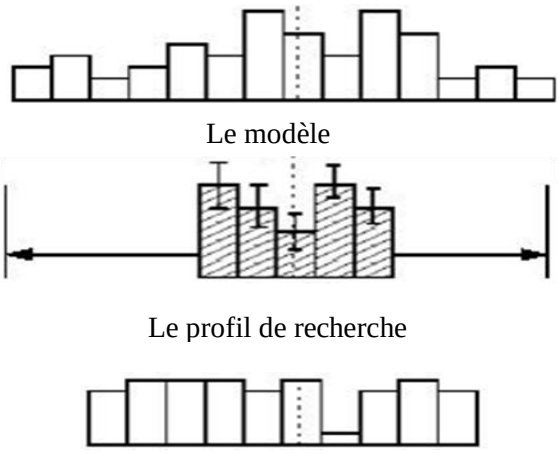


Figure 3.23. Recherche le long du profil échantillonné [Cootes et al. 1995].

Ayant une nouvelle disposition des points $X+dX$, les paramètres de pose et de forme doivent être ajustés rendant X aussi proche que possible de $X+dX$. On détermine les variations d'échelle ds, d'angle de rotation $d\theta$, et de translation dX_t, générant une forme la plus proche possible de $X+dX$ au sens des moindres carrées.

$$X + dX \approx T_{dX_t,ds,d\theta}(X)$$

Ayant mis à jour les paramètres de pose, il faut mettre à jour les paramètres de forme en résolvant l'équation matricielle :

$$X + dX = T_{X_t,s,\theta}(X)$$

La solution trouvée permettra de déterminer le vecteur b des poids associés aux directions principales du modèle.

Figure 3.24. Procédure de recherche sur un modèle de visage [Cootes et al. 1995].

Figure 3.25. Echec de la procédure de recherche [Cootes et al. 1995].

Recherche multi-résolution

Pour améliorer la robustesse et la rapidité de l'algorithme ASM, une approche multi-résolution est adoptée. Une pyramide gaussienne est construite pour chaque image (image de la base et image de test). Une image à échelle supérieure est obtenue à partir de la précédente par lissage gaussien et sous échantillonnage. Chaque image possède le quart du nombre de pixels de l'image du niveau inférieur.

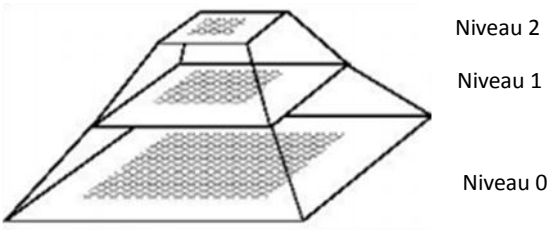


Figure 3.26. Pyramide gaussienne.

L'algorithme est appliqué à chaque niveau en commençant par le niveau le plus haut. A chaque niveau, la forme obtenue après convergence sert à initialiser le modèle au niveau inférieur. Le passage d'une échelle à une échelle supérieure se fait par simple multiplication du vecteur des coordonnées par deux.

3.4 Modèle de texture

La texture associée à une forme représente l'ensemble des échantillons de pixels en niveaux de gris pris à l'intérieur de la forme. La représentation mathématique de la texture d'un objet est un vecteur composé de m niveaux de gris :

$$g_{im} = (g_1, g_2, \dots, g_m)^T$$

Figure 3.27. Exemple d'un visage annoté de 68 points d'intérêt qui forment la forme (à gauche) et sa texture (à droite) [Bordei 2013].

L'enveloppe convexe de chaque forme est triangulée par une opération de triangulation, en appliquant par exemple la triangulation de Delaunay dont le principe stipule qu'aucun triangle ne contient de point caractéristique dans son cercle circonscrit.

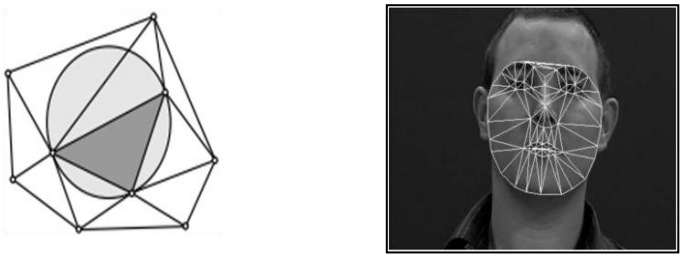


Figure 3.28. Propriété de Delaunay (à gauche) et triangulation de Delaunay d'une forme de visage (à droite) [Le Gallou 2007].

Pour construire un modèle statistique de texture, chaque image de la base d'apprentissage est déformée pour s'afficher selon la forme moyenne calculée précédemment.

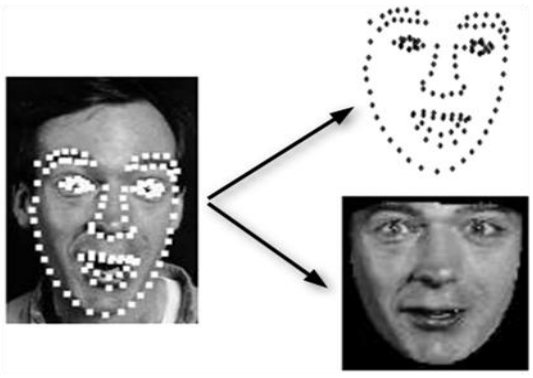


Figure 3.29. La déformation des formes [Cootes et al. 1995].

La déformation des formes ou «Wrapping » est par définition la transformation de la configuration spatiale d'une forme en une autre. Elle consiste à associer un ensemble de points caractéristiques de la première forme à un autre ensemble de points correspondant à la deuxième forme.

La correspondance de déformation se fait dès lors entre les triangles des enveloppes convexes par projection des points d'une forme à une autre. La déformation permet de construire des vecteurs de texture g_{im} de même taille, à savoir celle de la forme moyenne, par échantillonnage de l'intérieur des formes déformées :

$$g_{im} = (g_1, g_2, \dots, g_m)^T$$

m est le nombre de pixels représentant la surface de l'objet.

Pour minimiser l'effet des variations d'éclairage et ainsi pour égaliser les dynamiques de niveaux de gris des textures, les vecteurs de texture obtenus sont normalisés à l'aide d'une translation et d'une remise à l'échelle :

$$g = g_{im} - \beta/\alpha$$

Les valeurs de α et β sont choisies pour transformer au mieux chaque vecteur de texture à la moyenne standardisée des textures. La moyenne standardisée des textures est la moyenne centrée réduite. Etant donnée la moyenne standardisée $\bar{g}$, les valeurs de α et β sont données par :

$$\alpha = g_{im}.\bar{g} \ , \quad \beta = moyenne(g_{im})$$

α étant un paramètre défini en fonction de la moyenne, il est donc évident que le calcul de la moyenne des textures normalisées devient ainsi un processus récursif.

Algorithme de normalisation des textures

1- Estimer la moyenne des textures $\bar{g}$;
2- Standardiser $\bar{g}$ (ramener sa moyenne à 0 et sa variance à 1) ;
3- Pour chaque vecteur de texture g_{im} , calculer
   $\alpha = g_{im}.\bar{g}$
   $\beta = g_{im}.\mathbf{1}/m$
   Normailser g_{im}
4- Jusqu'à ce que $\bar{g}$ soit stable

L'analyse statistique de l'ensemble des vecteurs de texture alignés, par une ACP, permet d'obtenir le modèle statistique de texture :

$$g = \bar{g} + \Phi_g b_g$$

où $\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g_i$ est la texture moyenne (sans forme), g une texture synthétisée (sans forme), Φ_g la matrice des principaux modes de variation de texture et b_g le vecteur de poids associés aux modes de variation, contrôlant la déformation de la texture synthétisée.

3.5 Modèle combiné de forme et de texture

L'idée force de Cootes [Cootes et al. 1998a] est de lier les deux modèles statistiques décrivant la forme et les niveaux de gris par un troisième modèle statistique combiné. La modélisation de la corrélation qui existe entre la forme et la texture a pour but de définir la notion d'apparence.

Ainsi pour chaque instance de l'ensemble d'apprentissage définie par :

$$\begin{cases} X = \bar{X} + \Phi_s b_s \\ g = \bar{g} + \Phi_g b_g \end{cases}$$

les vecteurs de forme b_s et de texture b_g sont concaténés pour obtenir de nouveaux vecteurs de paramètres :

$$b = \begin{pmatrix} W_s . b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s . \Phi_s^T (X - \bar{X}) \\ \Phi_g^T (g - \bar{g}) \end{pmatrix}$$

où W_s est une matrice diagonale des poids associés aux paramètres de forme permettant de les ramener à l'échelle des paramètres de texture. Comme le vecteur des paramètres de forme b_s a une unité de mesure en pixels, et que celui de texture b_g est exprimé en niveaux de gris, ils ne sont pas comparables sans l'utilisation d'une matrice de pondération W_s. Une méthode simple d'estimation consiste à pondérer uniformément les composantes de b_s par un facteur r rapport entre la variance totale des textures et celle des formes.

Afin d'obtenir le paramètre d'apparence c, une ACP est appliqué sur le vecteur b :

$$b = \Phi_c . c$$

où Φ_c est la matrice des vecteurs propres fournis par l'ACP. Le vecteur c est le paramètre d'apparence contrôlant à la fois les paramètres b_s et b_g , c'est-à-dire la forme et la texture.

La forme et la texture peuvent être déduites à partir du vecteur d'apparence c par les équations suivantes :

$$X = \bar{X} + \Phi_s W_s^{-1} \Phi_{cs} c$$

$$g = \bar{g} + \Phi_g \Phi_{cg} c$$

$$\Phi_c = \begin{pmatrix} \Phi_{cs} \\ \Phi_{cg} \end{pmatrix}$$

Le modèle combiné s'écrit alors :

$$X = \bar{X} + \varphi_c . c$$

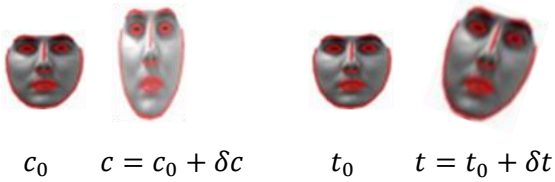
$$g = \bar{g} + \varphi_g . c$$

avec

$$\varphi_c = \Phi_s W_s^{-1} \Phi_{cs}$$

$$\varphi_g = \Phi_g \Phi_{cg}$$

Comme dans le cas du modèle de forme, on rajoute au modèle d'apparence un vecteur de pose *t*. Chaque exemple de la base d'apprentissage peut être synthétisé par une valeur particulière des paramètres d'apparence *c* et de pose *t*. En modifiant ces paramètres de δc et de δt, une nouvelle forme et une nouvelle texture sont alors synthétisées. Le vecteur de texture mis dans la forme moyenne est ensuite déformé en la nouvelle forme.



c_0 $c = c_0 + \delta c$ t_0 $t = t_0 + \delta t$

Figure 3.30. Effets de modification des paramètres d'apparence et de pose [Le Gallou 2007].

- **Image résiduelle**

La notion d'image résiduelle est introduite pour désigner la différence entre une image synthétique de l'objet délivrée par le modèle et l'image réelle :

$$\partial g = g_{image} - g_{modèle}$$

Soit une image i de la base d'apprentissage définie par les paramètres c_0 et t_0, en modifiant c_0 et t_0 respectivement de δc et de δt :

$$c = c_o + \partial c$$

$$t = t_0 + \partial t$$

une nouvelle forme X_m et une nouvelle texture g_m sont alors synthétisées.

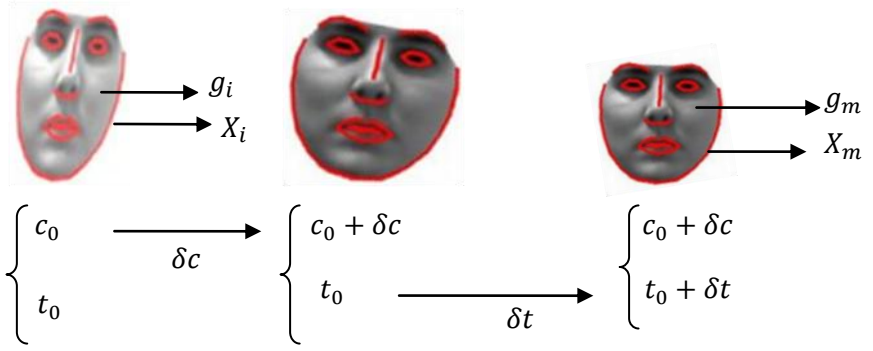


Figure 3.31. Synthèse d'une forme X_m et d'une texture g_m [Le Gallou 2007].

L'image résiduelle $\partial g = g_i - g_m$ et une regression linéaire à variables multilpes sur un certain nombre d'expériences portant sur la modification des objets synthétisés dans la base d'apprentissage de δc et de δt, fournissent une relaion entre δc et δg puis entre δt et δg :

$$\delta c = R_c * \delta g$$

$$\delta t = R_t * \delta g$$

R_c et R_t représentent les matrices d'expérience. Chaque expérience consiste à déplacer les paramètres de la position optimale d'une certaine quantité, et de mesurer la différence entre l'image réelle et l'image synthétique.

$$\partial g = g_{image} - g_{modèle}$$

Figure 3.32. Construction des textures pour la création de l'image résiduelle [Le Gallou 2007].

3.6 Modèle actif d'apparence

Le modèle actif d'apparence (AAM) est la désignation de l'algorithme de mise en correspondance dont le principe est de rechercher dans les images traitées de nouveaux objets (texture et forme) conformes au modèle statistique d'apparence, autrement dit, de trouver les

paramètres de pose et d'apparence du modèle qui décrivent le mieux les formes et les textures des objets présents dans les images. En partant d'une initialisation raisonnable, les modifications à apporter aux paramètres d'apparence et de pose permettent au modèle de s'ajuster à l'objet recherché dans de nouvelles images. Le problème de la recherche d'un objet est un problème d'optimisation dont la résolution suppose une relation de régression linéaire à multiples variables entre l'image résiduelle et les paramètres de pose δt et d'apparence δc.

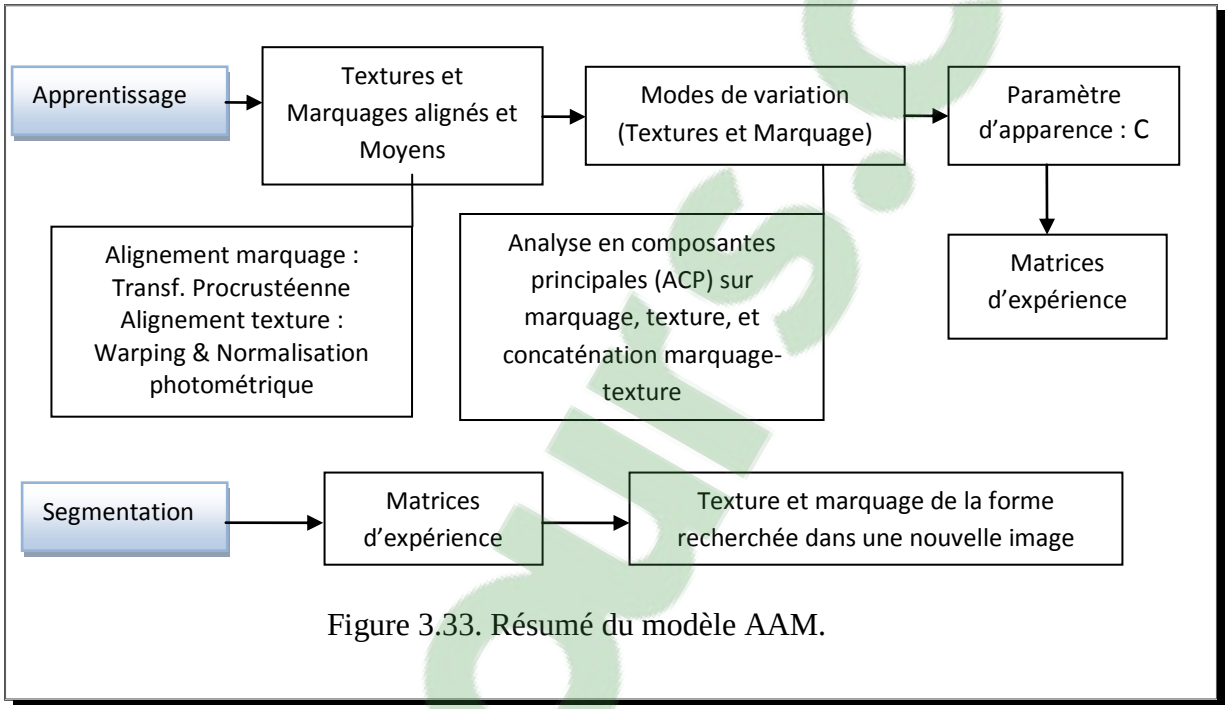


Figure 3.33. Résumé du modèle AAM.

La recherche de l'objet par l'algorithme de l'AAM se déroule comme suit :

1- Initialiser les paramètres du modèle c et t (initialement mis à 0) ;
2- Générer la forme X_m et la texture g_m à partir des paramètres c et t ;
3- Calculer la texture normalisée g_i à l'intérieur de X_m et dans l'image de l'objet recherché ;
4- Calculer l'image résiduelle $\delta g_0 = g_i - g_m$ et l'erreur résiduelle $E_0 = |\delta g_0|$;
5- Prédire les paramètres de pose $\delta t_0 = R_t * \delta g_0$et d'apparence $\delta c_0 = R_c * \delta g_0$;
6- Mettre à jour les paramètres $c_k = c_{k-1} - k * \delta c_0$ et $t_k = t_{k-1} - k * \delta t_0$ (k initialement fixé à 1) ;
7- Générer la forme et la texture, et évaluer l'erreur résiduelle $E_i = |\delta g_i|$;
8- Si $E_i < E_0$ alors accepter les mises à jour et aller à 2 avec $c = c_k$ et $t = t_k$

Lorsque la convergence de l'erreur résiduelle est atteinte, la représentation de la texture et celle de la forme de l'objet recherché sont synthétisées à travers le modèle par g_m et X_m. Comme dans les cas de l'ASM, une approche multi-résolution permet de rendre l'algorithme plus rapide et plus robuste.

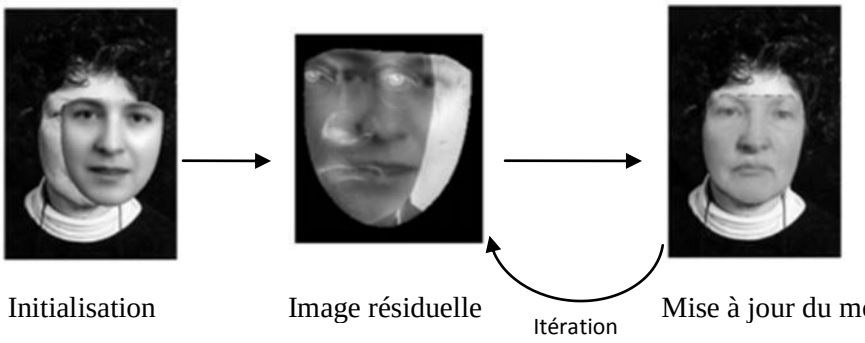


Figure 3.34. Mise en correspondance du modèle à l'image [Cootes et al. 1998a].

3.7 Discussion

Les modèles actifs de forme et d'apparence ont été largement utilisés et améliorés par la suite. Leur succès est fonction de la qualité de la base d'images et de sa variabilité. La base doit être la plus exhaustive possible et les points caractéristiques extraits doivent être significatifs des objets recherchés. L'étude expérimentale sur des données réelles est importante et permettra d'évaluer et de contrôler les performances des modèles dérivés. Les inconvénients majeurs rencontrés correspondent à l'opération d'annotation des images et à la connaissance de la position initiale des modèles. La tâche de l'annotation manuelle est fastidieuse et couteuse en temps, notamment pour de grands ensembles d'apprentissage. Elle doit être faite avec la plus grande précision, pour éviter les imprécisions dans les positions des points caractéristiques. Ces imprécisions induisent des erreurs non contrôlés dans le modèle. L'augmentation du nombre de points caractéristiques améliore généralement les résultats jusqu'à un certain seuil, où la forme étant suffisamment représentée, et l'insertion de nouveaux points donne des résultats moins précis. Un nombre élevé de points caractéristiques risque d'augmenter la sensibilité au bruit car chaque point caractéristique peut prendre une position erronée. L'insertion d'un nouveau point entre deux autres modifie la direction de recherche de ses points voisins car la normale au contour en chaque point est déterminée à partir de ses deux voisins. Le choix du nombre de points caractéristiques doit prendre en compte la représentativité de l'objet, la complexité du modèle et la sensibilité au bruit. Un travail heuristique serait donc indispensable pour faire un tel choix.

Un autre point important qui influence les résultats est celui de la longueur du profil de recherche. L'ASM effectue la recherche pour chaque point le long de la normale au contour qui doit contenir le point recherché ou au moins un point proche du point recherché et ayant un profil similaire. Un profil de recherche doit être assez long pour permettre une meilleure exploration mais il doit être aussi petit que possible pour réduire le temps de calcul. Si le profil est long, le modèle risque de diverger. Une étape préliminaire de localisation du modèle

s'avère indispensable. Cootes et al. [1995] [1998a] [1998b] ont confirmé la dépendance des modèles à l'initialisation.

L'application de l'ACP nécessite le choix du pourcentage de la variance expliquée. Ce choix est intimement liée au nombre de directions principales de variation décrivant l'ensemble des données. La suppression des petites valeurs propres est parfois nécessaire permettant de minimiser le bruit présent dans les données, assimilé aux vecteurs propres de petite variation.

Pour certaines applications, l'information de texture est déterminante et la méthode ASM ne suffit pas, et il convient d'utiliser la méthode AAM. Par contre l'AAM est plus sensible à l'initialisation. L'AAM est basé sur l'hypothèse de régression linéaire valable localement et la convergence est réalisée que si le modèle est initialisé dans un voisinage proche de la position idéale.

La robustesse des modèles actifs est liée à la variabilité introduite dans sa base d'apprentissage. Plus la variabilité de la base d'apprentissage augmente, plus le nombre de paramètres contrôlant les déformations du modèle augmente, ce qui complique la phase de la segmentation ou de la localisation, et augmente le temps de calcul. Ceci est un inconvénient majeur pour une implémentation temps réel. La convergence du modèle déformable créée devient difficile à partir d'une base contenant beaucoup de variabilité [Le Gallou 2007].

3.8 Conclusion

Dans ce chapitre, nous avons dressé les fondements théoriques des deux modèles déformables destinés à construire statistiquement un modèle de forme et un modèle d'apparence. Pour chaque modèle, on doit disposer d'un ensemble d'apprentissage constitué d'images contenant le même objet avec des formes et des apparences différentes. L'application à la segmentation constitue les modèles actifs de forme et d'apparence. Le modèle actif de forme permet de localiser les bords des objets ayant des formes semblables à celles de l'ensemble d'apprentissage et le modèle actif d'apparence permet de prendre en compte l'information de texture, en plus de celle de forme. Il en découle que le modèle AAM est plus fiable que le modèle ASM.

Les versions des modèles présentées sont les versions originales des publications de Cootes et al. [1995], [1997] [1998a] [1998b] [1999]. La démarche et la problématique des deux modèles continue de motiver les équipes de recherche. Elles sont primordiales et chaque point traité a ouvert la voie à d'autres chercheurs pour proposer leurs alternatives pour l'expérimentation et l'amélioration des résultats fournis dans les applications envisagées. Les alternatives proposées sont nombreuses et nous mentionnons que quelques unes comme le *Mixture Model for Representing Shape Variation* [Cootes et al. 1997], *Generalizing the Active Shape Model* [Al-Zubi et al. 2003], *Shape Parameter Optimization for AdaBoosted Active Shape Model* [Li et al. 2005], *Robust Active Shape Model Search* [Rogers et al. 2002], *Multi-view Non-linear Active Shape Model using Kernel PCA* [Romdhani et al. 1999], *le Shape-*

AAM [Cootes et al. 1998b], le *Modèle direct d'apparence* [Hou et al. 2001], l'*AAM par approche compositionnelle* [Matthews et al. 2004], l'*AAM optimisé par Simplexe* [Aidarous et al. 2007]. Les alternatives proposées par les équipes de recherche autre que celle de l'équipe du professeur Cootes ont été reprises par Cootes pour les commenter, pour les infirmer ou les confirmer, comme c'est le cas des AAM revisités où Cootes a démontré que cette approche est plus consommatrice en temps de calcul que la méthode classique à cause des calculs intensifs pour la descente de gradient et la matrice Hessienne [Cootes et al. 2001].

L'implémentation des modèles par certaines équipes de recherche servent souvent comme un banc d'expérimentation et de test. Nous citons *AAM demo* réalisé par l'équipe de Cootes, *AAM Application Program Interface* (*AAM-API*) réalisé par Stegmamm, *AAM toolbox*, Lucas-Kanadeitting algorithm en Matlab, Recognition and Vision Library en C++, AAM building en C++ [Gao et al. 2010].

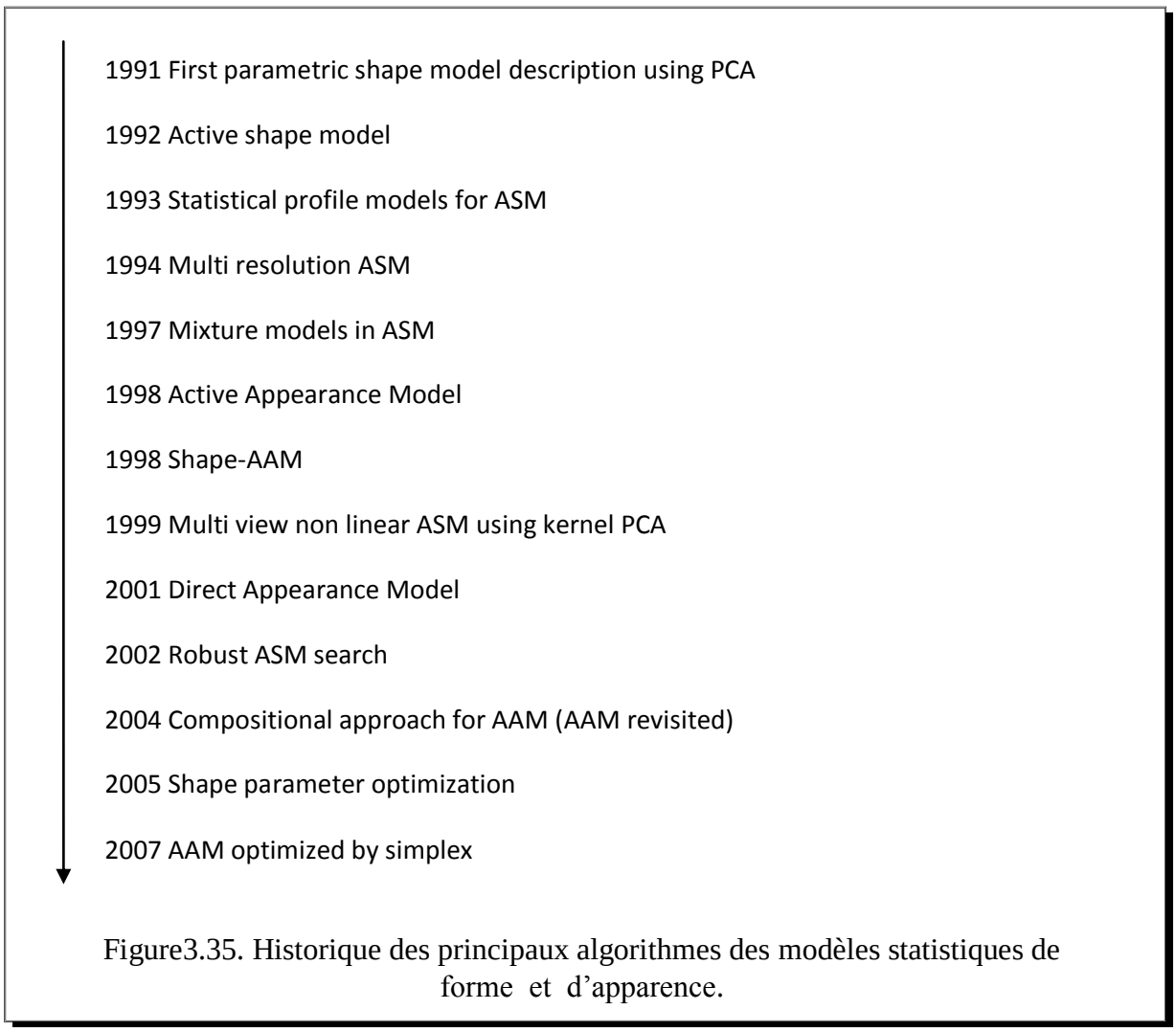


Figure3.35. Historique des principaux algorithmes des modèles statistiques de forme et d'apparence.

Les différents modèles cités sont applicables à une large variété de problèmes et fournissent un cadre intéressant pour l'interprétation automatique des images. Ils ont été largement utilisés dans le domaine de la segmentation d'images médicales et la reconnaissance faciale. De façon générale, ils sont performants dans les domaines de la reconstruction et la reconnaissance d'objets. Les modèles sont des méthodes de segmentation 2 D par excellence, et le passage en 3D peut être très difficile à cause des problèmes soulevés par la mise en correspondance entre les points caractéristiques. L'objectif à travers cette présentation est d'expliciter que le processus de segmentation théorique et pratique de tels modèles est complexe et est fortement dépendant de l'initialisation. Sa mise en œuvre ne semble pas du tout simple et les questions d'optimisation et de convergence des modèles restent toujours posées.

Chapitre 4

Généralités sur le calcul Parallèle

Le calcul parallèle est devenu une technologie clé dans l'augmentation de la capacité d'utilisation des ordinateurs modernes. Il existe un grand nombre de formes d'expression du parallélisme, chacune adaptée à certaines classes de problèmes. Il arrive qu'un même problème puisse être traité par plus d'une approche de parallélisme avec des reflets directs sur les performances obtenues. Cela offre une gamme de possibilités plus vaste que celle offerte par le calcul séquentiel. L'utilisation du parallélisme est compliquée car dans la pratique elle ne dépend pas uniquement des caractéristiques du problème mais aussi des architectures des machines à disposition. Cela rend difficile la tâche de ceux qui veulent proposer un tour d'horizon sur les différentes formes de parallélisme. Nous avons choisi de présenter un sous ensemble des possibilités du calcul parallèle orienté par nos besoins ainsi que les concepts qui nous seront utiles.

4.1 Introduction

De nos jours, les applications informatiques sont de plus en plus complexes et difficiles à réaliser. La performance n'est pas toujours au rendez-vous et dépend d'une adéquation entre l'architecture matérielle, le modèle de programmation et l'implémentation de celui-ci. Différents paradigmes de développement ont vu le jour, introduisant des concepts avancés permettant de structurer les applications informatiques. Les fabricants de processeurs, de mémoires et de systèmes de stockage ont amélioré leurs systèmes d'une manière sensible, à tel point que les logiciels montraient des améliorations de performance sans même que leur code source ne soient réadapté. La fréquence des processeurs est passée de 500 *MHZ* à 1 *GHZ* pour atteindre 3 *GHZ* de nos jours [Saidani 2012]. Les architectures vont évidemment continuer à évoluer et à améliorer leurs performances mais l'amélioration des logiciels ne se fera plus de manière aussi systématique que par le passé. Les révolutions architecturales ont changé de direction et s'orientent vers une multiplication des unités de calcul et une coordination plus efficace de leur exécution concurrente pour améliorer les performances des programmes. Le parallélisme est la nouvelle révolution dans la manière avec laquelle nous écrirons les logiciels. Ce changement est aussi important que l'avènement de la programmation orientée objet dans les années 1990. Des processeurs jusque-là réservés à des machines de calculs dédiés aux calculateurs des grands laboratoires scientifiques se retrouvent de nos jours dans des stations de travail de particuliers et même dans des ordinateurs portables voire embarqués. Les architectures parallèles sont devenues accessibles au grand nombre et n'ont pas cessé de s'améliorer en performance. Le traitement d'images est l'un des domaines qui requiert une grande puissance de calcul et qui se prête bien à la parallélisation. Les architectures parallèles les plus répandues sont :

- les architectures multi-cœurs à mémoire partagée du type *SMP* (*Symetric Multi-Processor*)
- les architectures graphiques GPU (*Graphics Processing Unit*)
- les architectures hétérogènes (Multi-CPU/ Multi-GPU).

Une amélioration de ces techniques induit automatiquement une amélioration aussi bien des applications séquentielles que des applications parallèles. La majorité des applications existantes sont séquentielles. Dans la pratique, l'utilisation du parallélisme n'est pas une tâche facile. Elle ne dépend pas uniquement des caractéristiques du problème mais aussi de celles de l'architecture des machines à disposition.

4.2 Architectures parallèles

La classification de référence des différents modèles de parallélisme est celle de Flynn [Flynn 1972]. Elle distingue quatre modèles :

- SISD pour *Single Instruction Single data* : il représente le modèle conventionnel de l'architecture de *Von-Newmann* où un seul flot d'instructions est traité par un flot séquentiel d'instructions. Ce mode ne fait pas évidemment partie du monde des architectures parallèles.
- SIMD pour *Single Instruction Multiple Data* : est une extension du modèle SISD à plusieurs flots de données. La même instruction est appliquée à un certain nombre de données. Le modèle SIMD est le modèle à *parallélisme de données*.
- MIMD pour *Multiple Instruction Multiple Data* : est le modèle où plusieurs instructions de nature différentes sont appliquées sur plusieurs flots de données, au même moment.
- MISD pour *Multiple Instruction Single Data* : il correspond au modèle dit *pipeline* ou SIMD vectoriel. Le traitement des données correspond à une séquence d'opérations sur plusieurs données de même type.

	Single data	Multiple Data
Single Instruction	SISD	SIMD
Multiple Instruction	MISD	MIMD

Table 4.1. Classification de Flynn.

Les critères de classification de Flynn s'avèrent très insuffisants pour établir une classification fine de toutes les machines parallèles actuelles. La classification suivante distingue les architectures parallèles selon le type d'hiérarchie de mémoire et permet de mieux comprendre les modèles de programmation parallèle pour les machines parallèles :

- Mémoire partagée : une architecture à mémoire partagée présente une mémoire unique accessible directement par l'ensemble des processeurs. L'existence d'une mémoire commune entre les unités de calcul simplifie le travail de parallélisation des algorithmes.

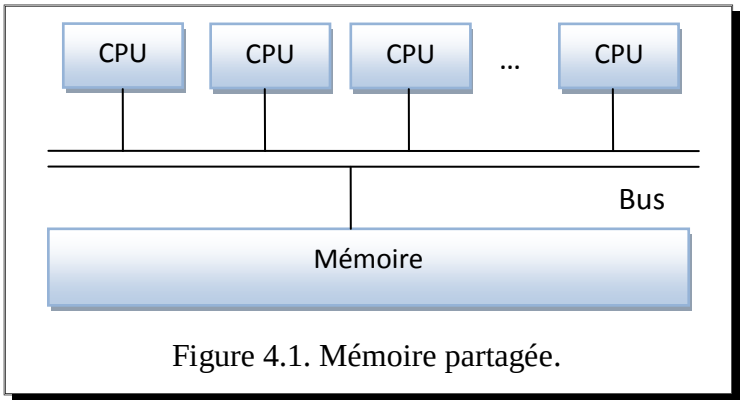


Figure 4.1. Mémoire partagée.

Il existe deux types de mémoire partagée : les mémoires à accès uniforme UMA (*Uniform Memory Access*) et les mémoires à accès non uniforme NUMA (*Non Uniform Memory Access*). Les architectures à mémoire UMA correspondent aux architectures multiprocesseurs symétriques SMP (*Symetric Multiprocessing*) où tous les processeurs ont les mêmes fonctions et se partagent les ressources du système de façon uniforme (figure 4.2). Les architectures à mémoire NUMA correspondent à un ou plusieurs SMP connectés (figure 4.3). Le modèle PRAM (*Parallel Random Acess Memory)* est un cas particulier du modèle à mémoire partagée.

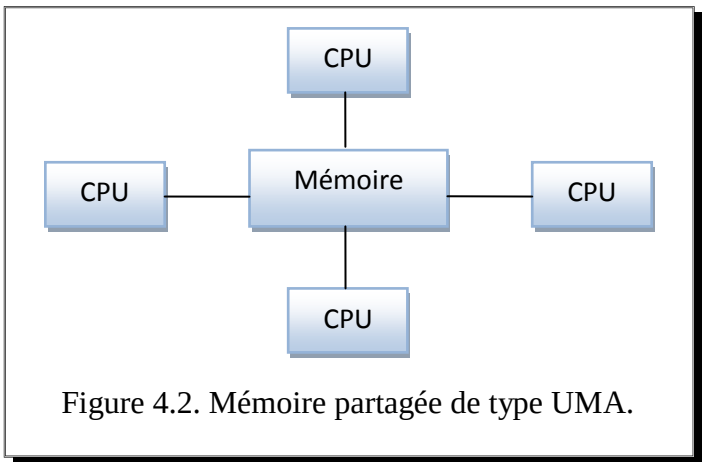


Figure 4.2. Mémoire partagée de type UMA.

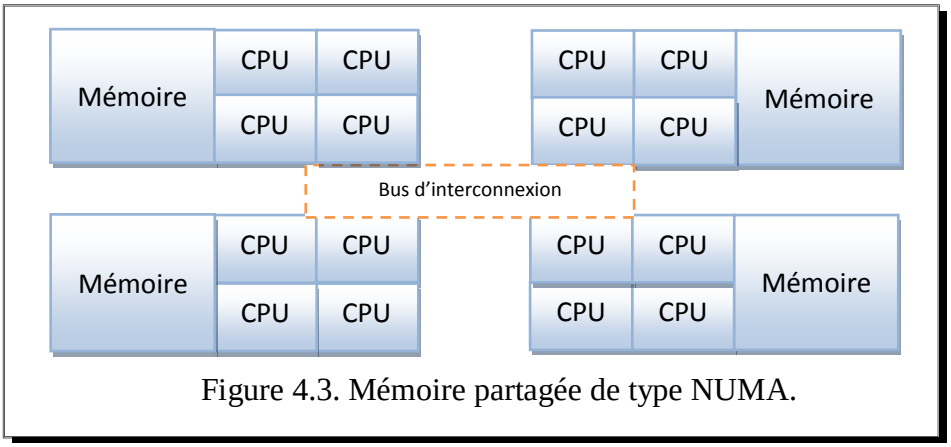


Figure 4.3. Mémoire partagée de type NUMA.

- Mémoire distribuée : une architecture à mémoire distribuée est composée d'un ensemble de nœuds connectés entre eux par un réseau de communication (figure 4.4). Chaque nœud est composé d'une mémoire locale et d'un processeur. Une architecture distribuée peut atteindre une taille importante en termes de nombre de processeur et de capacité mémoire. L'avantage évident d'une architecture distribuée est son adaptation à un parallélisme à grande échelle au contraire d'une architecture partagée. Les architectures massivement parallèles MPP (*Massively Parallel Processing*) sont des architectures distribuées composées de plus de dizaine de milliers de processeurs.

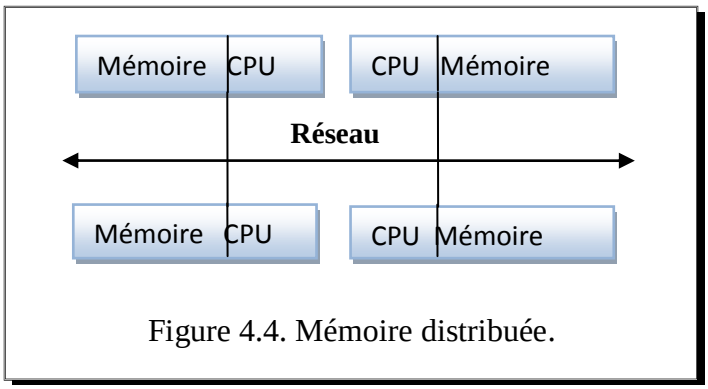


Figure 4.4. Mémoire distribuée.

- Architecture hybride : une architecture hybride regroupe les deux architectures présentées précédemment. Elle est composée d'un ensemble de nœuds multiprocesseurs reliés entre eux par un réseau à haut débit. La composante mémoire partagée est souvent un système SMP et la composante distribuée est la mise en réseau de plusieurs systèmes SMP.

Cas des microprocesseurs

Dans les architectures des micro-processeurs, différentes formes de parallélisme ont été intégrées progressivement dans les mécanismes internes d'exécution d'un programme. D'une conception SISD à l'origine, le développement de la technologie RISC (*Reduced Instruction Set Computing*) a favorisé l'introduction du pipeline dont le principe est basé sur le découpage d'une opération en sous opérations, ce qui permet la création d'une chaine d'exécution parallèle des différentes sous opérations. La tendance depuis 2004 vers le parallélisme niveau *thread* désigné par *Simultaneaous Multithreading* s'apparente à un modèle MIMD [Sicard 2004]. Dans une machine séquentielle, on peut trouver des unités de calcul parallèle (64 bits, carte GPU) et des processeurs multi-cœurs pouvant être considérés comme formant un système mini-distribué. Les outils de programmation utilisés pour tirer profit du parallélisme de ces architectures sont les bibliothèques de threads *Pthread* et *OpenMP* [Saidani 2012] [Boillod-Cerneux 2014] [Paraiso 2014]. Les premiers pas du *GPGPU* (*General Purpose computing on Graphics Processing Units*) ont démontré que l'architecture des processeurs graphiques est bien adaptée au calcul intensif. CUDA (*Compute Unified Device Architecture*) est un modèle de programmation parallèle conçu pour les architectures des cartes graphiques.

4.3 Modèles de programmation parallèle

Un modèle de programmation parallèle permet d'établir le développement d'une application parallèle sur un type d'architecture donné en faisant abstraction de ses détails techniques qui peuvent varier. Il est destiné à structurer les données manipulées et à définir les entités de l'algorithme à paralléliser et leurs interactions. Il doit satisfaire un certain nombre de caractéristiques telles que :

- Méthode précise de développement,
- Simplicité de compréhension,
- Indépendance vis-à-vis de l'architecture,
- Implémentation efficace,
- Estimation du coût du modèle.

L'aspect architectural et l'aspect programmation du modèle définissent le degré d'explicitation du parallélisme. On distingue ainsi plusieurs catégories de modèle [Skillicorn et al. 1998] :

- Modèle à parallélisme implicite : la parallélisation est effectuée par la transformation du code séquentiel en code parallèle par des compilateurs spécifiques (exemple : Prolog Parallèle), et l'exécution parallèle est complètement cachée au programmeur.
- Modèle à parallélisme explicite : la parallélisation est faite d'une manière explicite sans pour autant définir la structure parallèle de l'application.
- Modèle à décomposition explicite : correspond aux langages qui proposent des primitives explicites pour définir les tâches à exécuter en parallèle. La réalisation du point de vue ordonnancement et communication est totalement cachée au programmeur.
- Modèle à placement explicite : la définition et la mise en place des différentes tâches de l'application sont à la charge du programmeur. La communication et la synchronisation sont totalement transparentes.
- Modèle guidé par évènements : les évènements sont programmés pour déclencher la parallélisation des traitements.
- Modèle à parallélisme totalement explicite : tous les aspects de décomposition, de placement, de communication et de synchronisation de l'application, sont à la charge du programmeur de l'application parallèle.

La source du parallélisme est un deuxième critère pour catégoriser les modèles de programmation parallèle. Dans cette optique, on distingue deux groupes de modèles : le *parallélisme de données et le parallélisme de tâche*.

Dans le parallélisme de données, une même opération est appliquée en même temps à plusieurs éléments d'un ensemble de données. Cette source de parallélisme est fréquemment utilisée dans le cas des structures de données vectorielles. Dans une architecture à mémoire partagée, les opérations ont accès à la structure de données via la mémoire globale. Par contre dans une architecture à mémoire distribuée chaque ensemble de données réside dans la mémoire locale de chaque opération. Les données sont transmises à chaque unité qui les copie dans sa propre mémoire locale. La programmation avec ce modèle se fait en général en écrivant du code avec des constructions de parallélisme de données.

Le modèle S*tream Computing* est un type particulier de modèle basé sur le parallélisme de données [Saidani 2012]. Ce modèle est le modèle dominant sur les unités de calcul graphique. Le parallélisme utilisé est le modèle SIMD, où plusieurs unités de calcul exécutent la même

instruction sur un ensemble de données en parallèle. Parmi les architectures qui supportent le S*tream C*omputing, les processeurs GPU (*Graphic Processing* Unit) et les technologies des composants reprogrammables à haute densité FGPA (*fonctional Graphics processor*). Plusieurs langages ont été développés dans ce contexte, parmi lesquels CUDA (Compute Unified Device Architecture) et OpenCL (Open Computing Language) [Ospici 2013].

Le parallélisme de contrôle, appelé aussi parallélisme de tâche, consiste à découper le problème en sous-problèmes dans le but d'exécuter le maximum de tâches dans un même laps de temps. La mise en œuvre consiste à attribuer au mieux les tâches aux unités de calcul disponibles. L'écriture d'un programme consiste à écrire les différentes tâches, à spécifier les variables qu'elles partagent et les communications qui existent entre elles. Dans le parallélisme de contrôle, on trouve plusieurs modèles et leurs paradigmes de programmation :

- Modèle à mémoire partagée : dans ce type de modèle, les tâches partagent un espace d'adressage commun sur lequel ils peuvent lire et écrire des données de manière indépendante et asynchrone. L'accès à la mémoire partagée est contrôlé par des mécanismes tels que les sémaphores et les verrous.
- Modèle par échange de message : dans ce modèle, la programmation parallèle se fait par passage de messages. Les tâches échangent des données au travers des communications en envoyant et recevant des messages. Pour une communication synchrone, l'émetteur et le récepteur doivent être autorisés pour que la transmission ait lieu. Pour une communication asynchrone, l'émetteur envoie son message à n'importe quel moment, mémorisé et délivré au récepteur quand celui-ci le demandera. L'échange de messages est ainsi la technologie de base qui permet la réalisation des communications entre les unités d'une architecture à mémoire distribuée. Les tâches peuvent résider sur la même machine physique ou sur un nombre arbitraire de machines. Les implémentations du *Message Passing* prennent la forme d'une bibliothèque de routines. Le MPI (*Message Passing* Interface) est un standard pour l'interface de programmation par échange de messages. Il a été élaboré afin d'assurer la portabilité et la facilité d'utilisation. Actuellement, le MPI est le modèle de programmation le plus utilisé pour le *Message Passing*.
- Modèle à appel de procédure à distance : ce modèle permet d'utiliser les paramètres de procédure pour transmettre les données, ce qui supprime la nécessité de construire explicitement les messages. L'appel de procédure à distance RPC (*Remote Procedure Call*) est un appel de procédure entre deux processeurs distants, l'appelant et l'appelé. Le processus de l'appelé reçoit les paramètres d'entrée de la procédure, exécute le corps de la procédure et renvoie les résultats vers l'appelant. Cette technologie fut ensuite associée à la programmation orientée objet pour gérer des objets distribués sur le réseau. L'invocation de méthode à distance RMI (Remote Method Invocation) est la version orientée objet du modèle RPC. L'un des standards d'objets distribués est CORBA qui est multiforme et supporte la plupart des langages de programmation. D'autres standards existent comme le RMI (*Remote Method Invocation*) en Java.
- Modèle à thread : un thread ou processus léger est une unité d'exécution autonome qui peut effectuer une tâche en parallèle avec d'autres threads. Le flot de contrôle d'un

thread est purement séquentiel. L'utilisation des processus légers au lieu de processus comme unité de parallélisme permet d'aborder un parallélisme à grain fin. Chaque thread défini possède ses données locales, et un accès à la mémoire globale. Les threads ont une durée de vie variable et peuvent être crées et détruits tout au long du déroulement du programme. La communication entre les threads nécessite une synchronisation afin de garantir l'exclusivité de l'accès à un emplacement donné, à un instant donné pour un seul thread. Le modèle de programmation par *thread* est souvent associé au modèle à mémoire partagée. Les implémentations des threads sont généralement réalisées par l'utilisateur à l'aide d'une bibliothèque de fonctions, ou d'une série de directives intégrées dans le code parallèle. Il existe plusieurs versions d'implémentation des threads utilisables avec des langages de programmation. Deux implémentations standards les plus connues sont POSIX (*Portable Operating System Interface for uniX*) et OpenMP (*Open specifications for MultiProcessing*) [Saidani 2012]. L'avantage d'OpenMP par rapport à POSIX est sa capacité à cacher les détails de gestion des threads.

Le modèle auquel nous ferons souvent référence dans ce document est le parallélisme de données. Il est considéré comme étant la forme la plus directe du parallélisme, et il est probablement le modèle le plus simple. Il dérive directement du modèle de programmation procédural. Sa mise en œuvre consiste à définir les structures de données homogènes et une séquence d'instructions *data-parallèles*, qui s'appliquent à tous les éléments de la structure en même temps.

4.4 Méthodologie de parallélisation

Le développement d'une version parallèle d'une application nécessite avant tout l'étude de sa faisabilité. Il existe certains problèmes pour lesquels, il n'existe aucune forme de parallélisme exploitable. Dans le cadre de ce travail, nous nous focalisons sur le modèle à parallélisme totalement explicite. Dans ce type de modèle, le concepteur est libre de construire une application parallèle spécifique parfois différente de l'application séquentielle. Pour la plupart des problèmes, la parallélisation explicite est la seule envisageable pour obtenir de bonnes performances.

Une décomposition préalable de la structure de l'application doit être établie afin d'identifier :

- la structure des données,
- les différentes tâches formant des unités de parallélisation,
- les communications entre les tâches,
- la synchronisation.

Une fois la faisabilité validée, la source de parallélisme que le problème présente doit être identifiée. Le problème doit être ensuite partitionné selon la nature du parallélisme contenu dans l'application (figure 4.5). La décomposition exploitant le parallélisme de données consiste à diviser la structure de données de l'application en plusieurs parties égales de taille

quelconque et d'assigner à chacune des tâches une part des données sur laquelle elle effectue des traitements. Dans la décomposition fonctionnelle, les tâches effectuent des traitements différents sur les mêmes données.

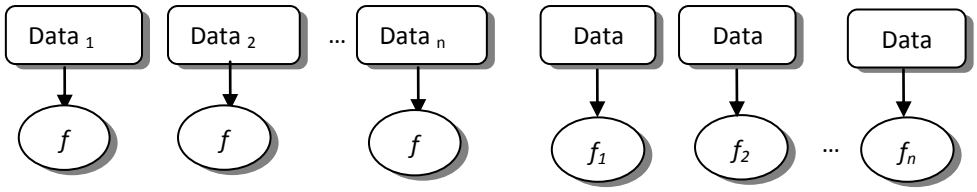


Figure 4.5. Parallélisme de données (à gauche) et parallélisme de tâches (à droite).

Toutes ces étapes de la décomposition font du parallélisme totalement explicite un travail d'expertise. Il n'est pas possible de chercher une approche type car chaque problème possède ses propres caractéristiques. Les points suivants résument les questions importantes du parallélisme totalement explicite d'un problème :

- Choix de la granularité : la granularité fine ou parallélisme à grain fin (*fine-grain*) correspond à la décomposition des tâches en sous tâches. La granularité grosse ou parallélisme à grain gros (*coarse-grain*) correspond au regroupent de tâches. En pratique, la diminution de la granularité augmente le parallélisme potentiel et permet d'utiliser plus d'unités de traitement. L'augmentation de la granularité permet de réduire le temps nécessaire à la gestion du parallélisme. Le choix de la granularité dépend de l'architecture et de l'algorithme.
- Ordonnancement des tâches : la stratégie d'ordonnancement dépend des caractéristiques du problème. Si le problème est régulier, l'ordonnancement est complètement déterminé. Lorsqu'il s'agit d'un problème irrégulier, dépendant des données en entrée, l'ordonnancement statique n'est pas possible. L'ordonnancement peut être effectué au fur et à mesure de la création des tâches.
- Placement des tâches : la stratégie de placement des tâches de l'application aux sites d'exécution dépend des caractéristiques du problème donné. Pour un problème à caractère régulier, la répartition est statique et pour un problème à caractère irrégulier, la répartition doit être dynamique.
- Equilibrage de charge : ou *load* balancing est une nécessité afin de minimiser les durées d'inactivité des sites d'exécution. Lorsque les tâches effectuent le même traitement, l'équilibrage de charge est trivial, puisqu'il suffit d'attribuer aux tâches les mêmes quantités de données. Si les tâches exécutent des traitements différents, un ajustement de la charge de travail est parfois nécessaire. Cependant, il n'est pas aussi facile de définir ce qu'est la charge d'une unité de traitement. En pratique, chaque problème intègre ses propres mécanismes d'équilibrage de charge.
- Le temps d'exécution : est le principal critère d'évaluation pour mesurer la performance d'une application parallèle. Il dépend de plusieurs paramètres tels que la taille de l'application, et de l'architecture de son exécution. Il peut être défini comme

étant le maximum des temps d'exécution mesurés sur chacun des unités ou des sites de traitement. La loi d'Amdahl [Fernandes 2002] [Saidani 2012] [Sicard 2004] définit l'accélération de l'application comme étant le rapport du temps d'exécution séquentiel sur le temps d'exécution parallèle :

$$Accélération = \frac{Temps\ d'exécution\ séquentielle}{Temps\ d'exécution\ parallèle}$$

L'efficacité de l'application est mesurée par le rapport de l'accélération sur le nombre de sites d'exécution.

- Qualité du parallélisme : bien souvent, lorsque l'on veut paralléliser une application séquentielle, les modifications que l'on doit faire sont tellement importantes que l'on se rapproche de la création d'une nouvelle application. Il est donc indispensable de préserver la qualité du résultat de l'application séquentielle.

4.5 Parallélisme et traitement d'image

Le traitement d'image est sans aucun doute un domaine privilégié pour l'étude et la mise en œuvre du parallélisme. Les systèmes de vision disposent souvent d'une chaîne de traitement, en partant de l'acquisition de l'image jusqu'au résultat final. Le traitement d'images combine deux contraintes fortes [Sicard 2004] [Saidani 2012]:

- La première contrainte est liée au très grand nombre d'opérations nécessaires même pour un calcul simple : une image couleur numérique de 720 x 576 points en composantes RVB (24 bits) représente environ 1,2 Mo (Méga octets) de données à l'état brut. Une opération simple de convolution sur un voisinage 3x3 (figure 4.6) représente un peu moins de 300 millions d'opérations de multiplication et d'addition en une seconde.

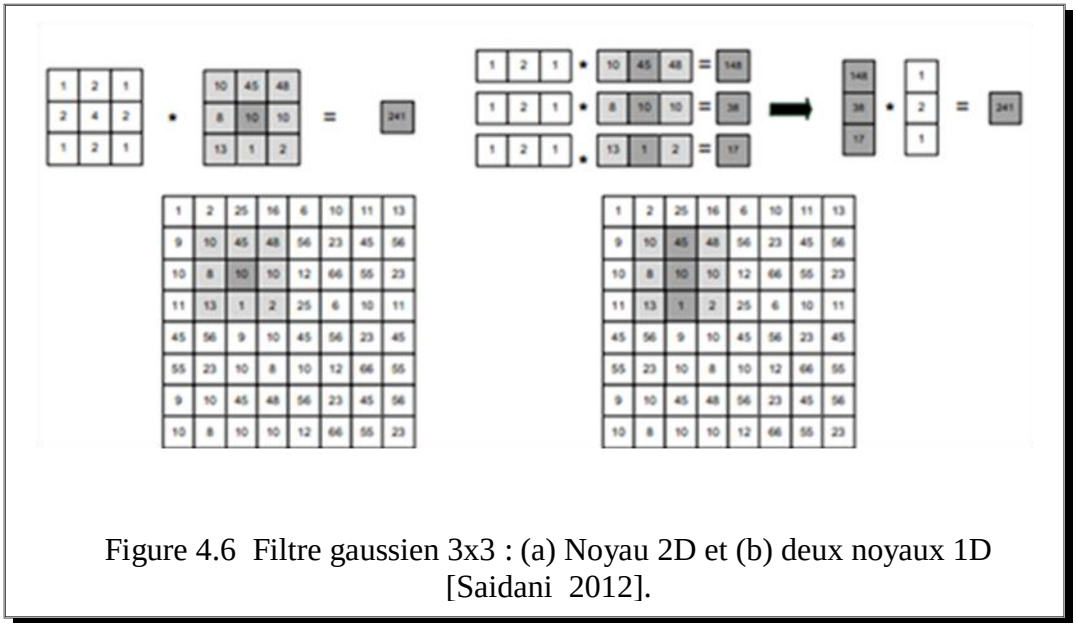


Figure 4.6 Filtre gaussien 3x3 : (a) Noyau 2D et (b) deux noyaux 1D [Saidani 2012].

- La seconde contrainte correspond à un besoin de temps de réponse très faible : pouvoir reconnaître avec une vitesse correcte peut être très utile dans de nombreux domaines. Dans le domaine médical, les cas urgents nécessitent une reconnaissance rapide. Dans le cas d'une chaîne de traitement pour l'assistance à la conduite automobile (figure 4.7), la machine doit être en mesure de prendre une décision, pour éviter le danger d'une situation d'accident par exemple, en une fraction de seconde.

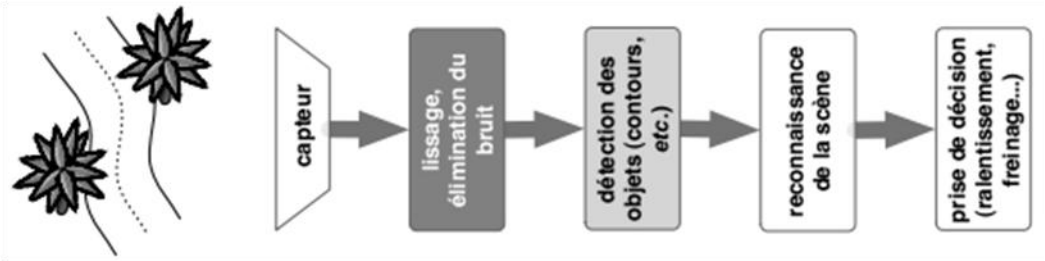


Figure 4.7. Chaîne de traitement dans un système d'assistance automobile [Sicard 2004].

Le parallélisme est donc une réponse naturelle aux contraintes du traitement d'image. En termes de modèle de programmation, Sicard, dans sa thèse [Sicard 2004], a prouvé que le data-parallélisme est une solution efficace et simple pour les traitements d'image de bas niveau. Il est très bien adapté aux structures dites régulières. Cependant, d'autres mécanismes doivent être mis en œuvre pour manipuler les structures irrégulières et d'autres formes d'irrégularité caractéristiques de l'analyse d'image. En effet, les calculs de niveau intermédiaire peuvent présenter des irrégularités dans la forme des données, dans leur

évolution, ou dans les traitements qui leur sont appliqués. Par exemple, lors d'un processus de segmentation, la taille, la surface ou la géométrie des régions peuvent être modifiées pour certaines régions qui se partagent ou se fusionnent, alors que d'autres ne changent pas.

L'évolution irrégulière des données est ainsi caractérisée par des degrés d'activité différents pour des sous-ensembles de données. L'irrégularité des traitements est aussi caractérisée par le fait que certains ensembles de données ne requièrent pas toujours le même type de traitement au même moment. En général, l'irrégularité d'un problème peut se concevoir par la forme et la représentation des données, qui nécessitent un codage adapté, ou par l'évolution de ces données, liée à l'exécution de l'algorithme, ou bien par la nature des traitements appliqués à ces données.

4.6 Conclusion

Dans ce chapitre, nous avons présenté au lecteur les possibilités existantes dans le calcul parallèle et distribué. L'idée de base est de présenter les différents aspects du domaine sans trop entrer dans les détails pour pouvoir insérer ensuite les contributions de cette thèse dans ce contexte. Le lecteur intéressé pourra consulter les références [Ospici 2013] [Saidani 2012] [Sicard 2004] [Fernandes 2002] [Boillod-Cerneux 2014] [Paraiso 2014]. L'architecture que nous prétendons utiliser est une architecture hybride composée d'une architecture à mémoire distribuée combinée à une architecture à mémoire partagée. Le modèle de programmation principal que nous envisageons d'utiliser est le data-parallélisme. La programmation par *threads* est le modèle de base pour la mise en œuvre des codes parallèles.

Chapitre 5

Méthodologie de parallélisation : Nos contributions

Nous avons vu dans les chapitres précédents le contexte dans lequel ce travail s'intègre. Nous allons discuter et présenter les propositions que nous avons conçues pour aborder progressivement le paradigme d'implémentation du modèle conçu pour la réalisation de la segmentation. On souhaite montrer les hypothèses de base en termes de réorganisation du modèle et de centralisation des données (voir figure 1.4 du chapitre 1), qui nous ont servi de point de départ pour notre étude et démarche de distribution et de parallélisation. Nous avons initialement proposé une version distribuée du modèle. Ensuite, nous avons proposé une démarche de parallélisation basée sur la parallélisation de chaque opération du modèle. Nous avons procédé à l'étude de faisabilité des trois premières opérations à savoir la procédure d'alignement, le calcul des profils d'intensité et l'analyse en composantes principales.

5.1 Nos motivations

Le chapitre 3 a mis en évidence les points fondamentaux des modèles statistiques de forme et d'apparence. Ces modèles sont exploités à des fins de segmentation. Chacune de leurs étapes est cruciale et un mauvais choix pour l'une d'elles compromet la chaine entière. Ils constituent un sujet complexe comme le soulignent la majorité des travaux réalisés qui portent sur l'étude de la robustesse et de la convergence des modèles dans des situations réelles. Les implémentations proposées concernent généralement l'algorithme de segmentation et sont réalisées en C/C++ et/ou Matlab. La majorité des applications réalisées sont séquentielles. A notre avis, aucun paradigme de développement n'a été cité quant à l'expérimentation et l'implémentation complète des modèles. Il nous a donc paru intéressant de concevoir une démarche dans laquelle nous empruntons des facilités de notation proches de celles utilisées pour les systèmes distribués et parallèles. Nous employons un schéma méthodologique pour représenter les modèles statistiques. Le schéma proposé constituera un cadre de développement flexible et cohérent pour une mise en œuvre progressive des différents traitements des modèles. Nous avons pensé qu'une version parallèle (figure 5.1) de la méthode ne serait intéressante que si elle pouvait être mise en œuvre de façon distribuée sur plusieurs processeurs connectés via un réseau rapide et avec un gain de performance suffisant pour que la méthode soit viable dans la pratique. En effet, les modèles statistiques de forme et d'apparence ne traitent pas que les images 2D, mais évoluent dans le sens du traitement des différents objets multimédia. Son paradigme d'implémentation doit aussi évoluer dans la même direction que celle de ses modèles.

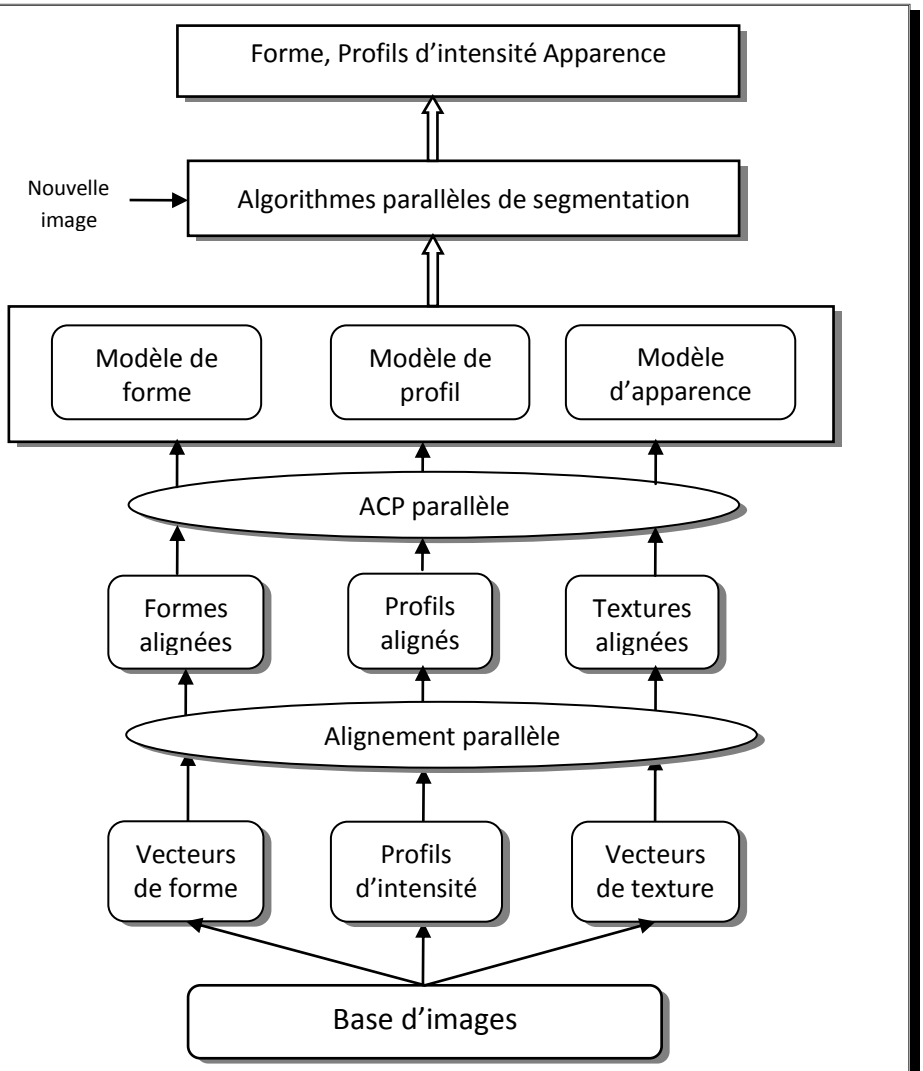


Figure 5.1. Version parallèle proposée pour les modèles de forme et d'apparence.

5.2 Parallélisation distribuée

Dans le modèle AAM, les données initiales sont constituées essentiellement de vecteurs de forme, de vecteurs d'intensité et des vecteurs de texture. L'ensemble de tous ces vecteurs forme une base d'information initiale et centrale du modèle. Dans la structure du modèle, elles sont traitées séquentiellement. Le modèle est donc fortement sérialisé à cause des dépendances de données. Les données du modèle forment aussi un domaine de données hétérogènes. Le passage d'une version séquentielle vers une version partiellement ou totalement parallèle nous semble possible. Le schéma du modèle semble à première vue couvrir les deux formes de parallélisme : données et tâches (parallélisation multithread).

Pour l'étude du degré de parallélisme du problème, nous avons initialement décidé de restructurer le modèle dans un schéma distribué, et de répartir l'ensemble de ses opérations principales autour de sa base de données (figure 5.2).

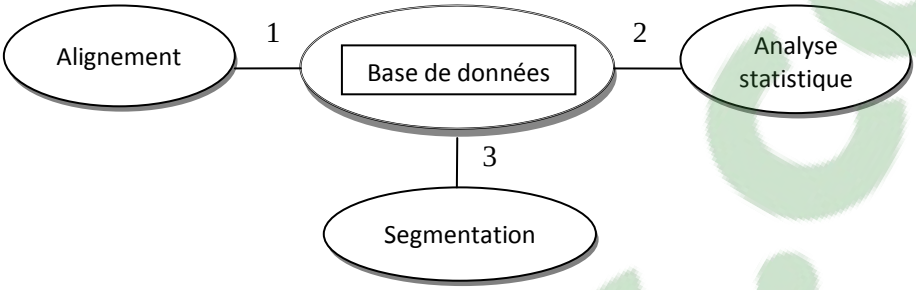


Figure 5.2. Version distribuée simplifiée du modèle de forme et d'apparence.

Plusieurs approches sont disponibles pour le faire. Pour notre solution distribuée, nous avons considéré le modèle classique de type client-serveur (figure 5.3). Cette technique met en œuvre un processus « client » chargé d'invoquer des processus chargés d'exécuter les opérations du modèle. Dans cette répartition, un traitement local de parallélisme peut être envisagé chez le client et chez le serveur. On peut aussi envisager de lui associer un modèle de type maître-esclave. La problématique du modèle s'inscrit parfaitement dans ce contexte. La technique maître-esclave met en œuvre un processus maître chargé de coordonner l'exécution des différentes opérations du modèle en les déléguant à d'autres processus. Pour notre solution distribuée, le modèle d'exécution envisagé repose sur le concept des appels de procédure à distance RPC (*Remote Procedure Call*) et plus précisément les appels de méthodes à distance RMI (*Remote Method Invocation*). Le paradigme client-serveur est similaire au paradigme maître-esclave sauf que le serveur est au service du client.

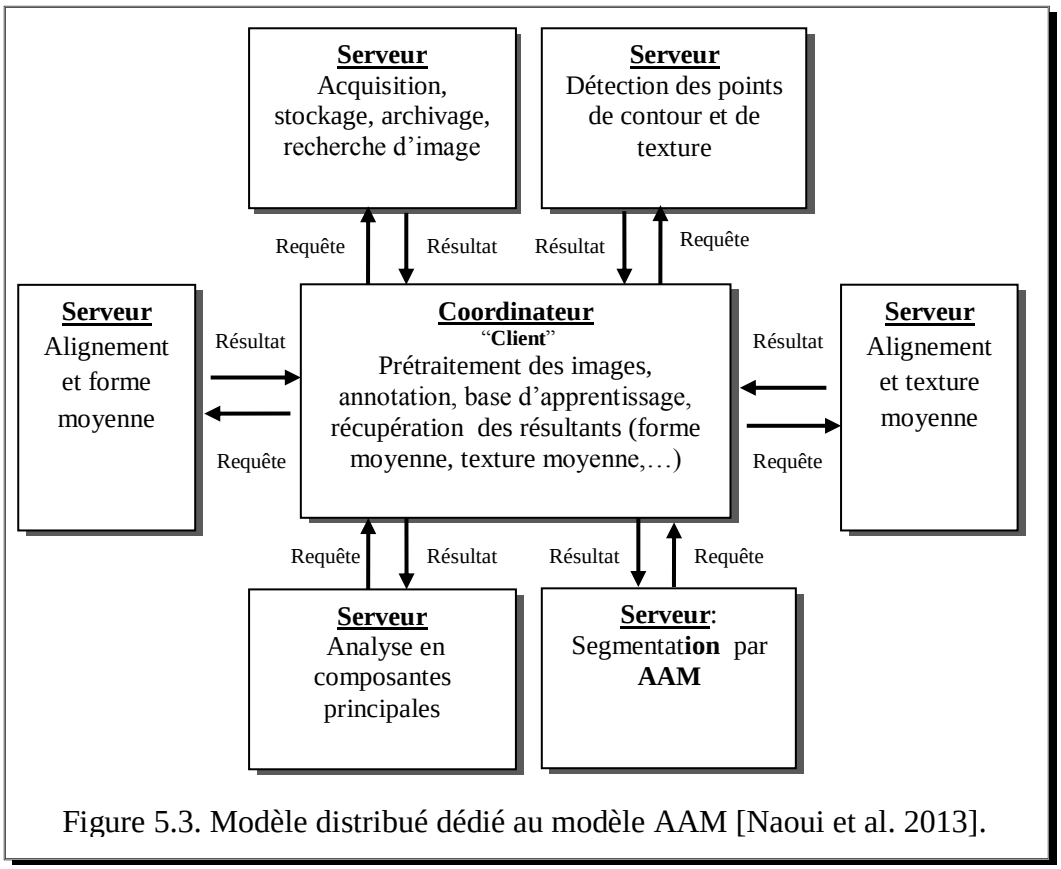


Figure 5.3. Modèle distribué dédié au modèle AAM [Naoui et al. 2013].

Le système proposé repose donc sur un schéma méthodologique. Il est conçu pour bâtir et utiliser les modèles statistiques, à des fins d'une perspective d'application réelle dédiée à la segmentation d'images. Le système proposé est composé d'une chaîne de traitements qui se décompose en modules d'acquisition d'images, de prétraitement (convertissant les images en niveaux de gris, réduisant la sensibilité aux variations de l'illumination, etc.), de détection des points caractéristiques servant à améliorer l'annotation des images, d'alignement de forme, d'alignement de texture, d'analyse statistique, et du module de segmentation. L'efficacité de chaque module contribuera sans aucun doute à l'efficacité globale du système proposé.

Sur ce type d'architecture, nous avons relevé les opérations purement data-parallèles. Pour les autres opérations, nous proposons d'intégrer le parallélisme à base du multithreading (figure 5.4).

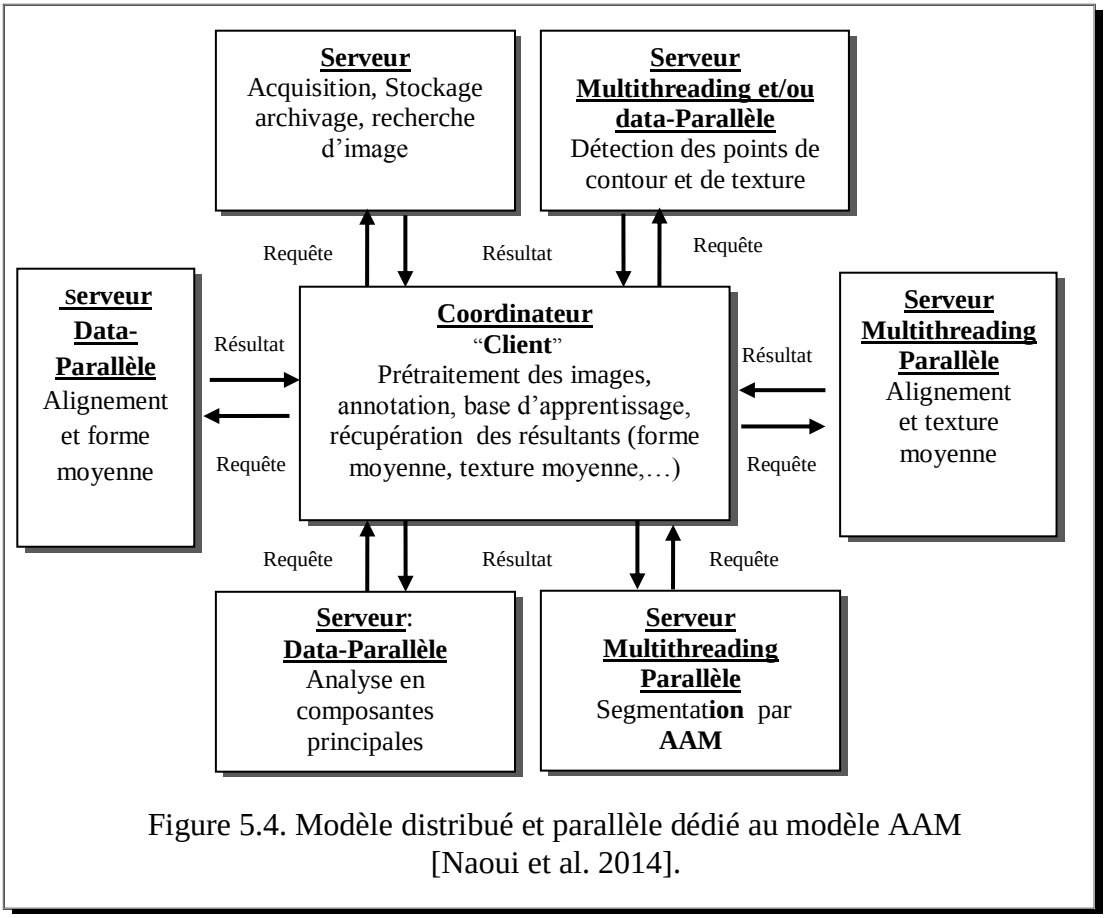


Figure 5.4. Modèle distribué et parallèle dédié au modèle AAM [Naoui et al. 2014].

5.3 Version parallèle : Etude de faisabilité

Le descriptif proposé dans la section précédente constitue le point de départ qui nous a permis de tracer nos propositions et de chercher le type de parallélisme pouvant convenir à la totalité ou à une partie du modèle. L'approche procédurale du modèle prouve la dépendance entre les données, ce qui ne privilégie pas une approche totalement parallèle. Une solution simple à envisager est d'explorer les opérations du modèle une par une (table 5.2), et d'identifier le type de régularité des données manipulées.

Opération	Données d'entrée	Données de sortie
Annotation des formes	Images	Points d'annotation
Calcul des profils	Points d'annotation	Intensités
Texture des formes	Profils (Intensités)	Profils alignés et profil moyen
Alignement des formes	Points d'annotation	Points d'annotation alignés et forme moyenne
Alignement des profils	Profils	Profils alignés
Alignement des textures	Intensités des points de forme, la forme moyenne	Intensités alignées dans la forme moyenne
Analyse statistique de formes	Vecteurs alignés	Composantes principales de forme
Analyse statistique de textures	Intensités alignées	Composantes principales de texture
Segmentation	Forme et texture	Forme et texture

Table 5.1. Liste des opérations et des données du modèle.

Comme nous l'avons déjà mentionné, nous avons procédé à l'étude des premières opérations du modèle, que sont l'alignement des formes, le calcul des profils et la procédure d'analyse en composantes principales. Les données principales de ces opérations sont essentiellement les points de forme et leurs intensités. Dans notre cas, les m points de forme sont des points à deux dimensions représentés par des abscisses et des ordonnées (formule 1). L'ensemble constitue une structure régulière :

$$X_i = \begin{pmatrix} (x_j^i) \\ (y_j^i) \end{pmatrix}, \text{ pour } i = \overline{1,n} \text{ et pour } j = \overline{1,m} \qquad (1)$$

Nous proposons de répartir tout vecteur de forme $\mathbf{X_i}$ composé de m points en deux vecteurs $\mathbf{X_{i_x}}$ **et** $\mathbf{X_{i_y}}$, l'un représentant toutes les abscisses des points et l'autre représentant toutes les ordonnées des points :

$$X_{i_x} = \{x_j^i\}\,;\; X_{i_y} = \{y_j^i\},\; \text{pour } j = \overline{1,m} \qquad (2)$$

Tout traitement **P** effectué sur ces points est à la fois effectué sur les abscisses et les ordonnées, que nous formulons ainsi :

$$P(X_i) \equiv P(X_{i_x}) \parallel P(X_{i_y}) \qquad (3)$$

Cette information complémentaire semble donc intéressante à exploiter. On désigne par P_1 et P_2 deux copies du traitement P, et on réécrit la formule (3) en formule (4) :

$$P(X_i) \equiv P_1(X_{i_x}) \parallel P_2(X_{i_y}) \qquad (4)$$

C'est sur ce principe (formule 4) que nous avons traité les trois premières opérations du modèle [Naoui et al. 2014]. Nous décrivons ici la façon dont nous avons mis en œuvre ce principe.

5.3.1 Cas de la procédure d'alignement [Naoui et al. 2014]

Comme décrit au paragraphe 3.2.2 du chapitre 3, l'alignement des vecteurs de forme est déterminé sur la base de l'analyse de Procrustes. Nous reprenons la formule de base de l'analyse de Procrustes (formule 5) à partir de laquelle est conçue l'alignement de deux formes, et telle qu'elle est présentée dans [Cootes et al. 1995] :

$$E = \left(X_1 - M(X_2)\right)^T W\left(X_1 - M(X_2)\right) \qquad (5)$$

La somme pondérée E (formule 5) représente la distance de Procrustes entre deux formes X_1 et X_2. W est la matrice diagonale des poids associée à chaque vecteur de forme. M est la matrice des paramètres de rotation θ, de translation (t_x, t_y) et de facteur d'échelle s, appliqués à X_2. Selon les formules proposées en (1) et (2), X_2 est composé de deux vecteurs, celui des abscisses $\{x_i^2\}$ et celui des ordonnées $\{y_i^2\}$. Ainsi $M(X_2)$ devient :

$$M\begin{pmatrix} x_i^2 \\ y_i^2 \end{pmatrix} = \begin{pmatrix} (s\cos\theta)\, x_i^2 - (s\sin\theta)\, y_i^2 + t_x \\ (s\sin\theta)\, x_i^2 + (s\cos\theta)\, y_i^2 + t_y \end{pmatrix} \qquad (6)$$

On réécrit la matrice M d'une manière plus simple, on obtient :

$$M\begin{pmatrix} x_i^2 \\ y_i^2 \end{pmatrix} = s\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x_i^2 \\ y_i^2 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \qquad (7)$$

On note par R :

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad (8)$$

On obtient ainsi :

$$M\begin{pmatrix} x_i^2 \\ y_i^2 \end{pmatrix} = s\,R\begin{pmatrix} x_i^2 \\ y_i^2 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \qquad (9)$$

Par des décompositions et des transformations simples, on obtient les formules suivantes:

$$M\begin{pmatrix} x_i^2 \\ y_i^2 \end{pmatrix} = s\,R\begin{pmatrix} x_i^2 \\ 0 \end{pmatrix} + s\,R\begin{pmatrix} 0 \\ y_i^2 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \qquad (10)$$

$$M\begin{pmatrix} x_i^2 \\ y_i^2 \end{pmatrix} = s\,R\begin{pmatrix} x_i^2 \\ 0 \end{pmatrix} + \begin{pmatrix} t_x \\ 0 \end{pmatrix} + s\,R\begin{pmatrix} 0 \\ y_i^2 \end{pmatrix} + \begin{pmatrix} 0 \\ t_y \end{pmatrix} \qquad (11)$$

$$M\begin{pmatrix} x_i^2 \\ 0 \end{pmatrix} = s\,R\begin{pmatrix} x_i^2 \\ 0 \end{pmatrix} + \begin{pmatrix} t_x \\ 0 \end{pmatrix} \qquad (12)$$

$$M\begin{pmatrix} 0 \\ y_i^2 \end{pmatrix} = s\,R\begin{pmatrix} 0 \\ y_i^2 \end{pmatrix} + \begin{pmatrix} 0 \\ t_y \end{pmatrix} \qquad (13)$$

$$M\begin{pmatrix} x_i^2 \\ y_i^2 \end{pmatrix} = M\begin{pmatrix} x_i^2 \\ 0 \end{pmatrix} + M\begin{pmatrix} 0 \\ y_i^2 \end{pmatrix} \quad (14)$$

$$M(X_2) = M(X_x^2) + M(X_y^2) \quad (15)$$

On réécrit la formule E à l'aide de la matrice **M** transformée, ce qui nous permet d'obtenir :

$$E = ((X_x^1, X_y^1) - M(X_x^2, X_y^2))^T W((X_x^1, X_y^1) - M(X_x^2, X_y^2)) \quad (16)$$

On décompose ainsi la formule **E** (16) en deux nouvelles formules (17) et (18) de même type que (5), l'une relative aux abscisses et l'autre correspondant aux ordonnées, ce qui nous donne au final :

$$E_x = (X_x^1 - M(X_x^2))^T W(X_x^1 - M(X_x^2)) \quad (17)$$

$$E_y = (X_y^1 - M(X_y^2))^T W(X_y^1 - M(X_y^2)) \quad (18)$$

$$E = (E_x, E_y) \quad (19)$$

A partir de la formule (19), on estime qu'il est possible d'appliquer une forme de data-parallélisme à la procédure d'alignement, et on prouve ainsi théoriquement la faisabilité d'une telle solution. En effet, chaque vecteur de forme X^i est décomposé en deux vecteurs X_x^i et X_y^i, le premier correspondant à ses abscisses et le second correspondant à ses ordonnées. La procédure d'alignement de deux vecteurs de formes X^i et X^j est équivalente à la même procédure appliquée aux vecteurs X_x^i et X_x^j d'une part et d'autre part aux vecteurs X_y^i et X_y^j .

$$\text{Aligner } (X^i, X^j) : \begin{Bmatrix} \text{Aligner } (X_x^i, X_x^j) \\ \text{Aligner } (X_y^i, X_y^j) \end{Bmatrix}$$

Une telle procédure peut être généralisée à l'ensemble des vecteurs de forme, comme le montre le schéma de la figure 5.5.

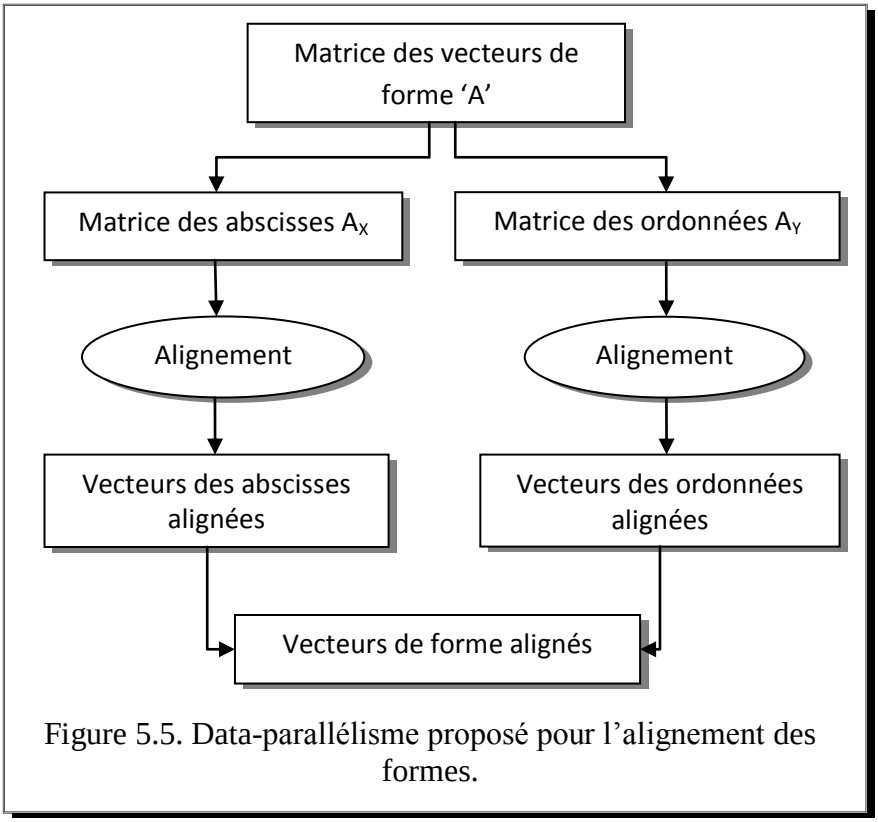


Figure 5.5. Data-parallélisme proposé pour l'alignement des formes.

5.3.2 Cas des profils des points [Naoui et al. 2014]

Comme décrit au paragraphe 3.2.8 du chapitre 3, le profil d'un point d'annotation est conçu sur la base de l'approche dérivée de l'image en ce point. Nous décrivons ici la méthode générale, baptisée *méthode gradient*, que nous avons adoptée pour mettre au point le calcul de profil et de son gradient, et qui s'articule autour de trois points essentiels (figure 4.13) :

- Calcul du gradient de la fonction intensité I de l'image au point d'annotation i(x, y) : le calcul du gradient au point d'annotation traduit une forte variation de la fonction d'intensité de l'image autour de ce point. En termes de vecteur gradient, il traduit la présence locale d'un module élevé en ce point. La direction du gradient est orthogonale à la frontière qui passe au point considéré et maximise la dérivée directionnelle. Le gradient est orienté dans le sens de la croissance du champ des intensités.
- Génération d'un profil de longueur p : sur la base du gradient calculé, on extrait p intensités le long de la direction du gradient : $I_i = (I_i^1, I_i^2, \dots, I_i^p)$.
- Calcul du gradient du profil : la dernière étape est de déterminer, pour chaque profil d'un point, son gradient : $Y_i = \frac{dI_i}{\sum_k dI_i^k}$, tel que $dI_i = (dI_i^1, dI_i^2, \dots dI_i^p)$. dI_i représente l'ensemble des gradients calculés aux points du profil.

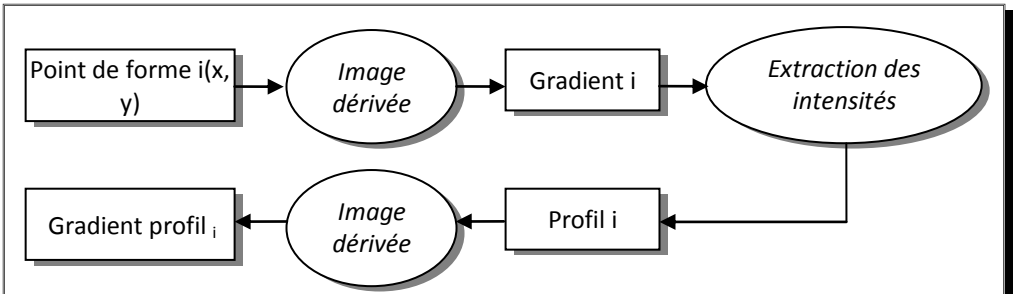


Figure 5.6. Calcul du gradient de profil d'un point d'annotation i.

Le gradient représente la dérivée première de la fonction d'intensité de l'image. Il est utilisé souvent dans la détection des contours qui se matérialisent par une rupture d'intensité de l'image dans une direction donnée. Plusieurs méthodes permettent de le déterminer. L'approche la plus classique consiste à choisir deux directions privilégiées orthogonales sur lesquelles on projette le gradient. Les principales méthodes dérivatives sont les filtres de dérivée première, conçues sur la base d'opérateurs discrets appliqués à une image par opération de filtrage.

Ainsi, au point i(x, y) le gradient $\nabla I_i(x,y)$ est le vecteur des dérivées horizontale et verticale de l'intensité de l'image :

$$\nabla I_i(x,y) = \left(\frac{\partial I_i(x,y)}{\partial x}, \frac{\partial I_i(x,y)}{\partial y}\right) \qquad (20)$$

Le gradient est caractérisé par son module et sa direction. Les expressions usuelles de ces grandeurs en norme euclidienne sont :

$$|\nabla I_i(x,y)| = \sqrt{\left(\frac{\partial I_i(x,y)}{\partial x}^2 + \frac{\partial I_i(x,y)}{\partial y}^2\right)} \quad (21)$$

L'évaluation de ces expressions exige des calculs longs et précis en type réel, elles sont souvent remplacées par un calcul simplifié :

$$|\nabla I_i(x,y)| \approx \left|\frac{\partial I_i(x,y)}{\partial x}\right| + \left|\frac{\partial I_i(x,y)}{\partial y}\right| \qquad (22)$$

La direction du gradient est donnée par :

$$\theta = \tan^{-1}\left(\frac{\partial I_i(x,y)}{\partial y} / \frac{\partial I_i(x,y)}{\partial x}\right) \qquad (23)$$

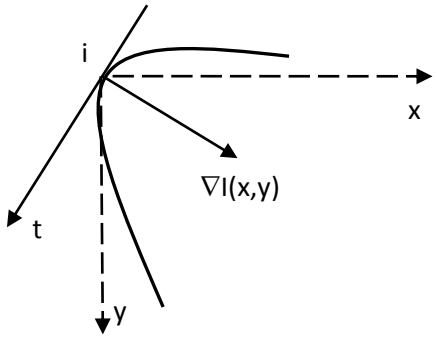


Figure 5.7. Vecteur gradient.

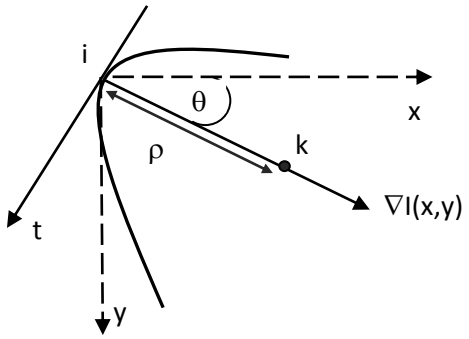


Figure 5.8. Un point 'k' du gradient.

Pour tout point k du gradient, nous définissons ses coordonnées x' et y' dans le repère de l'image de la façon suivante :

$$x' = x \pm \rho * \cos(\theta) \qquad (24)$$

$$y' = y \pm \rho * sin(\theta) \qquad (25)$$

ρ est la distance (en pixels) qui sépare le point k du point d'annotation i. La tangente t dans la figure 5.8 représente la ligne de niveau.

Le gradient du profil correspond à :

$$dI_i^k(x', y') = \nabla I_i^k(x', y') = \left(\frac{\partial I_i^k(x', y')}{\partial x'}, \frac{\partial I_i^k(x', y')}{\partial y'}\right) \qquad (26)$$

La norme du gradient du profil au point k est :

$$\left|\nabla I_i^k(x', y')\right| = \sqrt{\left(\frac{\partial I_i^k(x',y')}{\partial x'}^2 + \frac{\partial I_i^k(x',y')}{\partial y'}^2\right)} \qquad (27)$$

Ou tout simplement :

$$\left|\nabla I_i^k(x', y')\right| \approx \left|\frac{\partial I_i^k(x',y')}{\partial x'}\right| + \left|\frac{\partial I_i^k(x',y')}{\partial y'}\right| \qquad (28)$$

La direction du gradient du profil au point k est :

$$\theta_k = \tan^{-1}\left(\frac{\partial I_i^k(x',y')}{\partial y'} / \frac{\partial I_i^k(x',y')}{\partial x'}\right) \qquad (29)$$

Ainsi, le gradient du profil au point i(x, y) est :

$$dI_i = \left\{dI_i^k\right\} \text{ for } k = \overline{1, p} \qquad (30)$$

Autrement dit :

$$dI_i = \begin{Bmatrix} \frac{\partial I_i^k(x',y')}{\partial x'} \\ \frac{\partial I_i^k(x',y')}{\partial y'} \end{Bmatrix} \text{ for } k = \overline{1, p} \qquad (31)$$

Nous pouvons conclure que la méthode proposée pour le calcul du gradient et de son profil se prête bien à un parallélisme de données. Pour chaque point d'annotation, son intensité est dérivée en abscisse et en ordonnée. La dérivation en ce point est une opération composée de deux opérations de dérivation sur deux données différentes. La procédure est la même pour tous les points d'annotation d'un même vecteur de forme (figure 5.7). Pour chaque point de forme, elle est également la même et itérative pour tous les points de son profil.

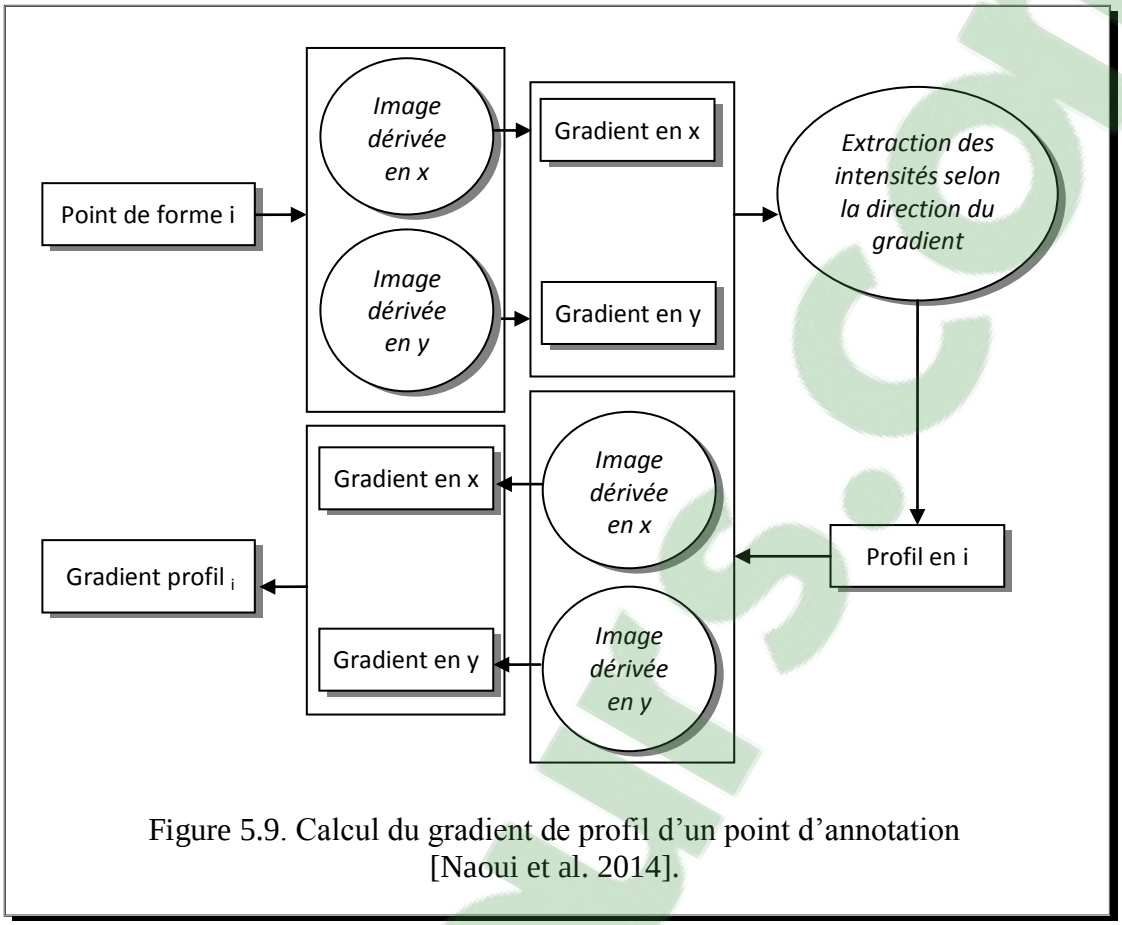


Figure 5.9. Calcul du gradient de profil d'un point d'annotation [Naoui et al. 2014].

5.3.3 Cas de l'analyse en composantes principales

De la même façon que la section précédente, nous traitons la méthode d'analyse en composantes principales décrite au paragraphe 3.2.3 du chapitre 3 et qui traite la totalité des vecteurs de forme alignés et regroupés dans une matrice de la manière suivante :

$$A = \begin{pmatrix} \begin{pmatrix} x_1^1 & \dots & x_1^n \\ & \vdots & \\ x_m^1 & \dots & x_m^n \end{pmatrix} \\ \begin{pmatrix} y_1^1 & \dots & y_1^n \\ & \vdots & \\ y_m^1 & \dots & y_m^n \end{pmatrix} \end{pmatrix} \qquad (32)$$

La matrice A est composée de deux sous matrices, une sous matrice des abscisses et une sous matrice des ordonnées :

$$A_X = \begin{pmatrix} x_1^1 & \dots & x_1^n \\ & \vdots & \\ x_m^1 & \dots & x_m^n \end{pmatrix} \qquad (33)$$

$$A_Y = \begin{pmatrix} y_1^1 & \dots & y_1^n \\ & \vdots & \\ y_m^1 & \dots & y_m^n \end{pmatrix} \qquad (34)$$

Nous reprenons les opérations fondamentales de l'analyse en composantes principales, et nous introduisons le principe de décomposition de la matrice des données dans l'expression de leurs formules :

$$\overline{A_X} = \frac{1}{n} \sum_{i=1}^{n} A_{X_i} \qquad (35)$$

$$\overline{A_Y} = \frac{1}{n} \sum_{i=1}^{n} A_{Y_i} \qquad (36)$$

Les formules (35) et (36) représentent respectivement le vecteur moyen des abscisses et le vecteur moyen des ordonnées, d'où les représentations centrées suivantes :

$$dA_{X_i} = A_{X_i} - \overline{A_X} \qquad (37)$$

$$dA_{Y_i} = A_{Y_i} - \overline{A_Y} \qquad (38)$$

Les formules suivantes correspondent au calcul des matrices de covariance des abscisses et des ordonnées :

$$S_{A_X} = \frac{1}{n} \sum_{i=1}^{n} dA_{X_i} \, dA_{X_i}^{T} \qquad (39)$$

$$S_{A_Y} = \frac{1}{n} \sum_{i=1}^{n} dA_{Y_i} \, dA_{Y_i}^{T} \qquad (40)$$

Par conséquent, on peut donc prétendre déterminer deux catégories de vecteurs propres correspondant à deux catégories de valeurs propres :

$$S_{A_X} P_{X_k} = \lambda_{A_k} P_{A_k} \qquad (41)$$

$$S_{A_Y} P_{Y_k} = \lambda_{Y_k} P_{Y_k} \qquad (42)$$

Nous définissons deux types de variances totales :

$$V_{A_{X_T}} = \sum_i \lambda_{X_i} \qquad (43)$$

$$V_{A_{Y_T}} = \sum_i \lambda_{Y_i} \qquad (44)$$

Le choix des deux types de valeurs propres par rapport à un pourcentage de variance f_v doit vérifier :

$$\sum_{i=1}^{t} \lambda_{X_i} \geq f_v V_{A_{X_T}} \qquad (45)$$

$$\sum_{i=1}^{t} \lambda_{Y_i} \geq f_v V_{X_{Y_T}} \qquad (46)$$

Nous émettons donc l'hypothèse suivante :

$$X = \begin{pmatrix} X_X \\ X_Y \end{pmatrix} \cong \begin{pmatrix} \overline{X_X} \\ \overline{X_Y} \end{pmatrix} + \begin{pmatrix} P_{X_X} \\ P_{X_Y} \end{pmatrix} \begin{pmatrix} b_{X_X} \\ b_{X_Y} \end{pmatrix} (47)$$

La formule (47) correspond à l'équation fondamentale du modèle statistique de forme.

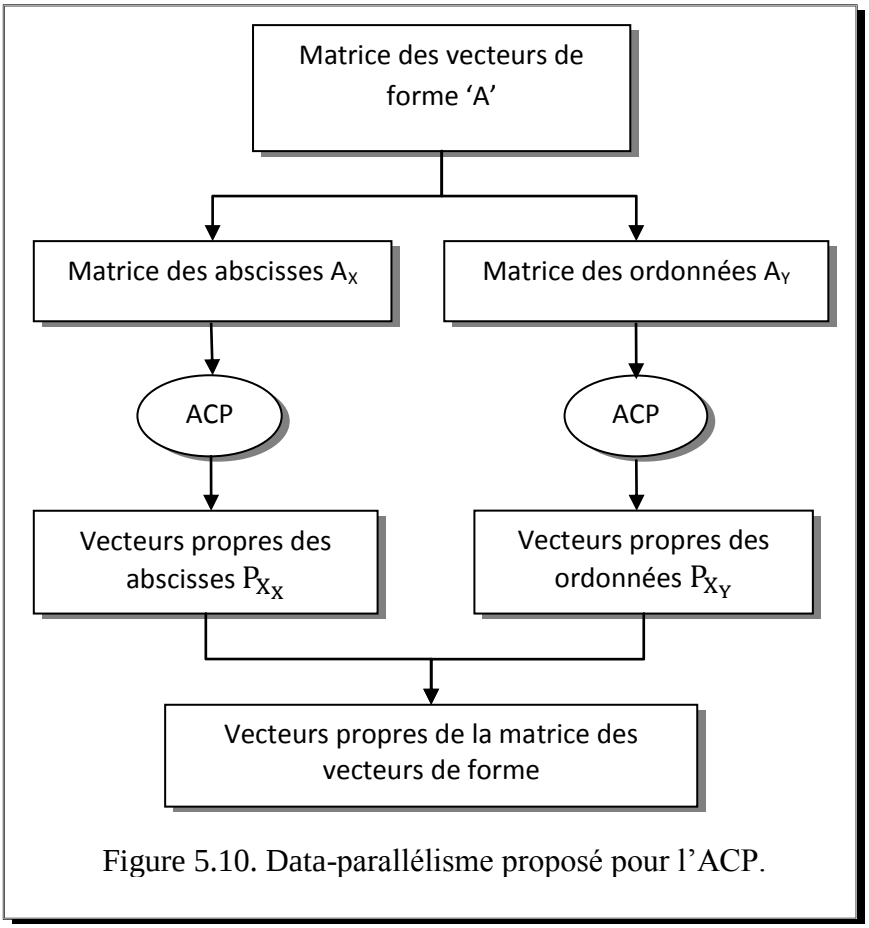


Figure 5.10. Data-parallélisme proposé pour l'ACP.

La méthode de décomposition ainsi proposée se prête aussi à un parallélisme de données (figure 5.10). L'analyse statistique faite par l'ACP sur tous les vecteurs de forme traite statistiquement les abscisses et les ordonnées. Les abscisses et les ordonnées constituent deux données différentes pour la même procédure statistique.

5.4 Conclusion

Dans ce chapitre, nous avons présenté nos contributions. Elles sont formulées en termes de faisabilité des solutions proposées. Nous avons voulu explorer le potentiel de parallélisation du modèle dont la finalité est l'élaboration d'une version parallèle et distribuée dédiée au modèle AAM qui profiterait certainement des avantages des approches proposées. Pour y arriver, nous avons repris les opérations initiales du modèle, et nous nous sommes attachés à montrer qu'une parallélisation des opérations du modèle est faisable, basée sur une

représentation régulière des données. Nous constatons que le parallélisme de données est très présent dans les opérations du modèle. Nous nous sommes orientés naturellement vers un découpage des données pour aboutir à un schéma fixe de décomposition des données. La mise en œuvre de nos contributions dans un environnement d'exécution adéquat et l'utilisation des bibliothèques de programmation et d'un langage tel que le C++, pour l'expression des algorithmes, permettra sans aucun doute d'espérer un paradigme d'implémentation structuré de haut niveau et d'implémenter les schémas que nous avons proposés au cours de notre étude. Dans cet aspect de parallélisation, l'idée nous semble extensible à d'autres opérations du modèle. Les principales difficultés résideraient certainement dans la phase de collecte des résultats et leur traitement. Il s'agit certainement d'une voie qui mériterait à terme d'être explorée.

Chapitre 6

Implémentation et expérimentation

Nous décrivons dans ce chapitre les implémentations développées au cours de cette thèse à partir des travaux décrits précédemment. Celles-ci ont été développées dans le but de démontrer la pertinence des travaux dans un contexte applicatif. Pour chaque application, nous concevons un schéma d'implémentation qui ne sera validé que dans un environnement d'exécution adéquat. Les premières applications ont été écrites en Java-RMI, afin de tester leur faisabilité dans un environnement distribué. Nous avons constitué un ensemble d'apprentissage composé de dix images de visage. Ces images nous ont permis de tester notre première version distribuée. Les autres applications ont été développées en C++ 2010, via la bibliothèque OpenCV (Open source Computer Vision library) version 2.4.5, et la bibliothèque Qt version 5.1.1 pour le développement des interfaces graphiques. Nous avons utilisé cette fois-ci une base d'apprentissage composée de six images issues de la base de données JAFFE (Japanese Female Facial Expression). Pour chaque application, nous avons intégré des opérations de prétraitement d'images, visant à améliorer la qualité visuelle de l'image. Il est important de rappeler au lecteur que nous sommes dans un cadre d'un travail expérimental et nous exposons un certain nombre de résultats expérimentaux.

6.1 Alignement de formes

L'application décrite ici vise à aligner un ensemble de vecteurs de forme, par la méthode d'analyse de Procrustes. Nous avons écrit et implémenté l'algorithme d'alignement décrit au chapitre 3 figure 3.11 en Java-RMI [Agueb et al. 2011]. Nous avons développé notre application sur deux PCs Pentium (R) Dual CPU 2*2.17 GHZ chacun doté d'une vitesse de 3.0 Mhz, d'une mémoire centrale de 3 Go chacun, et l'un fonctionne sous Windows Seven et l'autre sous Windows XP. L'application est développée en Java sous MyEclipse 8.0. La base de données est composée d'images de visage contenant la variabilité en pose et en expression (figure 6.1). Nous nous sommes intéressés à la forme des lèvres décrite par son contour externe. L'annotation des images est manuelle. Nous avons choisis les coins et les points intermédiaires sur le contour des lèvres (figure 6.4). La chaîne de traitements de l'application est détaillée sur la figure 6.2. Nous avons effectué deux tests. Pour le 1èr test, nous avons positionné manuellement huit points sur le contour des lèvres, et dix points pour le 2ème test.

Figure 6.1. Base d'images utilisée dans la procédure d'alignement.

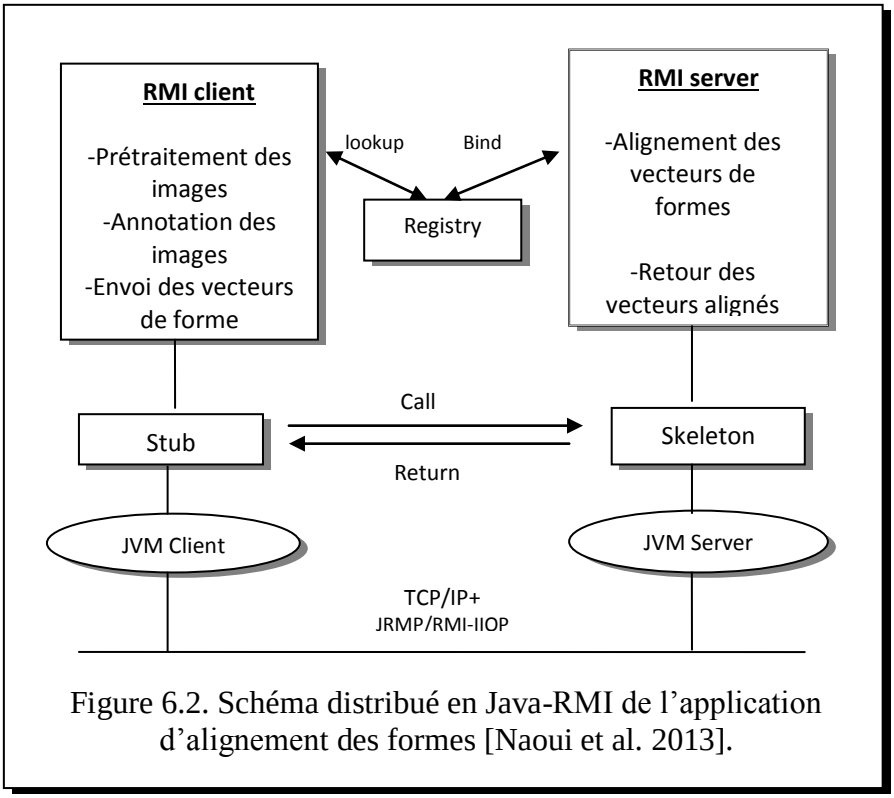


Figure 6.2. Schéma distribué en Java-RMI de l'application d'alignement des formes [Naoui et al. 2013].

L'application est composée de deux modules :

- Module Client : chargé des opérations de lecture, de prétraitement, d'annotation des images, et envoi des vecteurs de forme au serveur.
- Module serveur : chargé d'exécuter la procédure d'alignement.

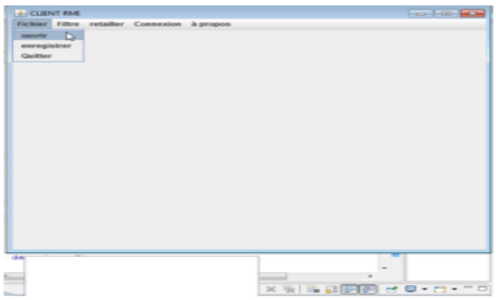


Figure 6.3. Interface client (à gauche) et interface client avec affichage d'image (à droite).

Figure 6.4. Annotation manuelle du contour des lèvres d'une image.

Figure 6.5. Interface client et connexion au serveur.

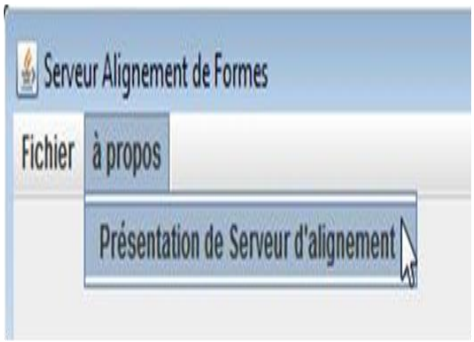


Figure 6.6. Interface serveur (à gauche) et interface serveur avec affichage d'images (à droite).

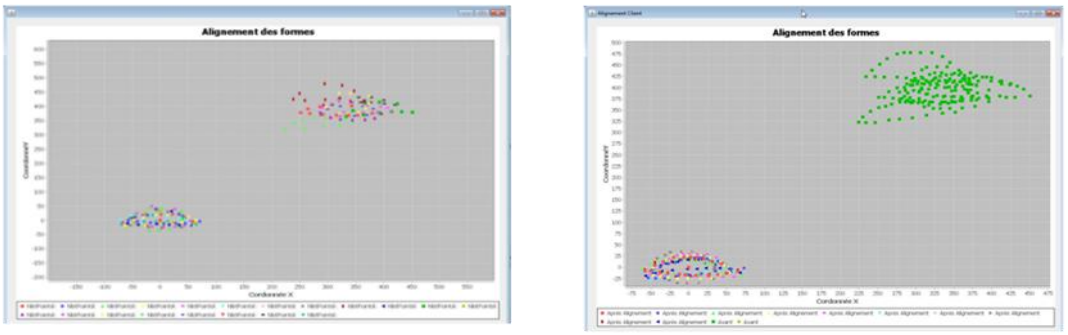
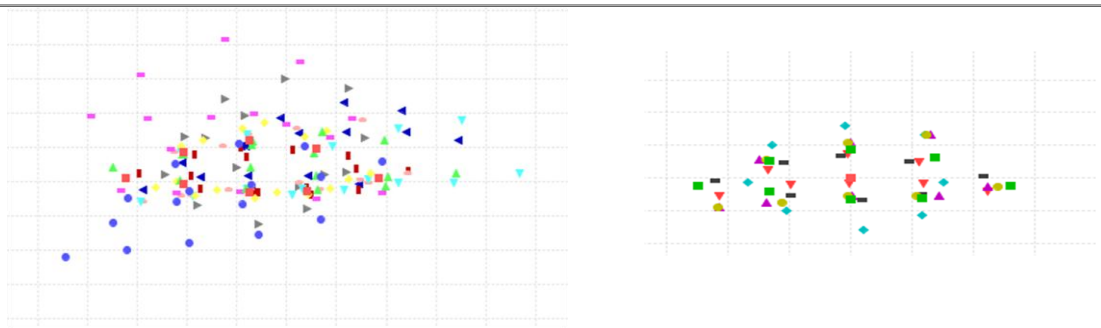


Figure 6.7. Représentation graphique des formes alignées. A gauche un ensemble représenté par 10 points et à droite un ensemble constitué de 20 points.

Le premier test effectué sur la base d'images avec huit points d'annotation nous donne de bons résultats. Nous avons confronté notre application à un second test, en annotant chaque image par dix points. Les résultats fournis révèlent le bon alignement des vecteurs de forme. La figure 6.7 et 6.8 illustrent les résultats obtenus.



(a) (b)

Figure 6.8. (a) Vecteurs de forme non alignés. (b) Vecteurs de forme alignés.

6.2 Analyse en composantes principales : version distribuée

L'application décrite ici est une première version test de l'ACP effectuée sur une matrice quelconque. Nous l'avons implémentée en Java-RMI selon le schéma de la figure A.9 dans le même environnement que celui de l'implémentation de la procédure d'alignement. La chaîne de traitements est composée des calculs de :

- la matrice réduite ;
- la matrice de corrélation ;
- les vecteurs propres ;
- les nouvelles coordonnées.

Les résultats de l'exécution (figure A.10) obtenus représentent les vecteurs propres de la matrice donnée et ne présentent aucun intérêt puisque la matrice de données est quelconque. L'objectif est de valider l'implantation de la méthode dans un environnement distribué orienté objet.

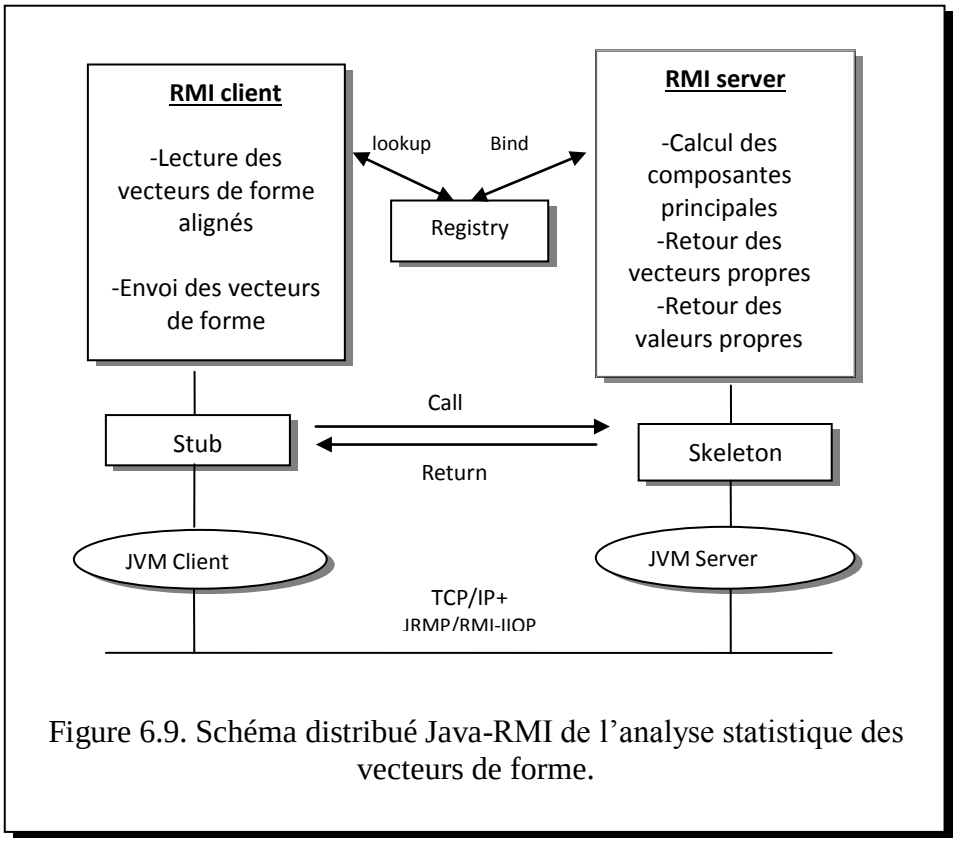
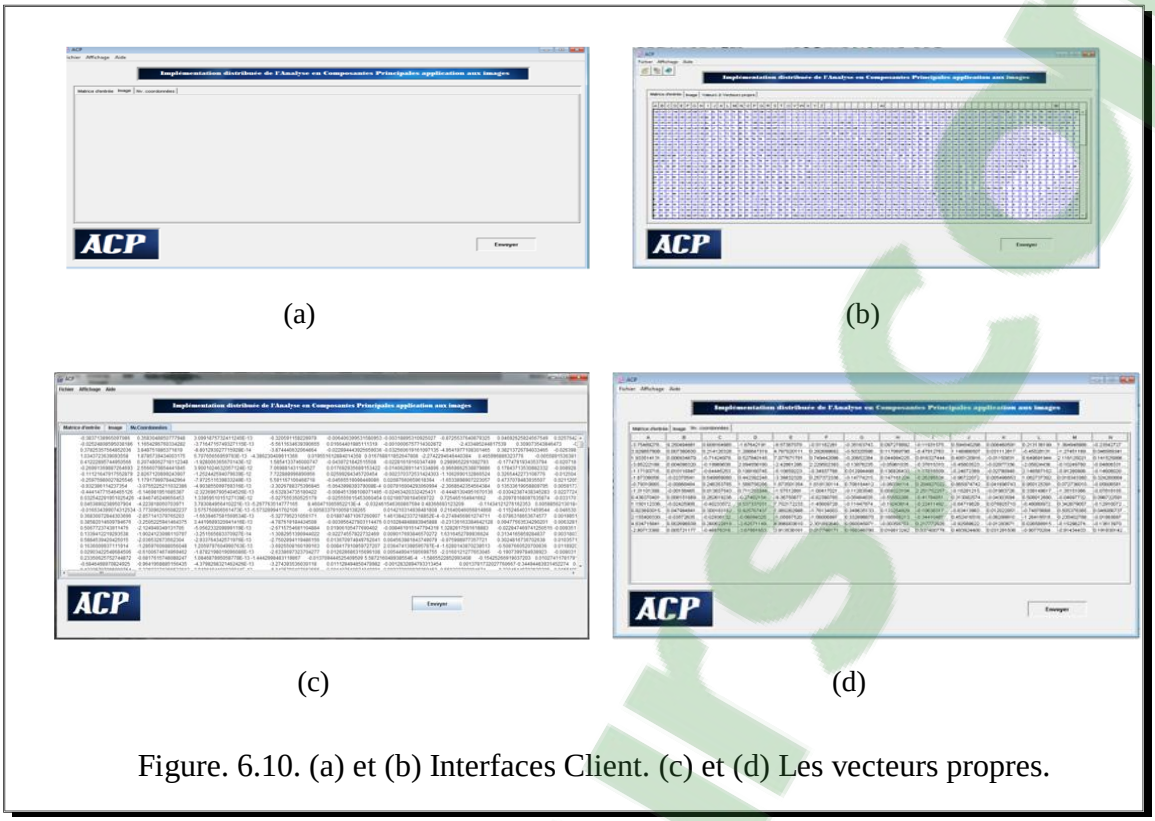


Figure 6.9. Schéma distribué Java-RMI de l'analyse statistique des vecteurs de forme.



(a) (b)

(c) (d)

Figure. 6.10. (a) et (b) Interfaces Client. (c) et (d) Les vecteurs propres.

6.3 Profils des points d'annotation

Deux versions de l'application du calcul des profils des points d'annotation ont été réalisées dans le même environnement que celui de l'ACP version C++ et version Qt, afin de vérifier, de corriger et de valider les formules (24) et (25) proposées pour le calcul des profils. Nous utilisons la même base de données provenant de la base JAFFE. Le gradient de l'image est calculé sur la base du filtre de Sobel (figure 6.12). Nous avons aussi utilisé le filtre de Roberts et celui de Prewitt. La validité des profils est en fonction de la précision des points d'annotation. Pour cette raison, nous avons intégré dans l'application la détection des contours des images traitées avant d'entamer l'opération d'annotation. L'annotation des images de contour devra aider à apporter plus de précision dans le tracé des profils. En vue d'une application multithreading, nous avons conçu un schéma d'implémentation illustré dans la figure 6.11 mais qui n'a pas été mis en exécution.

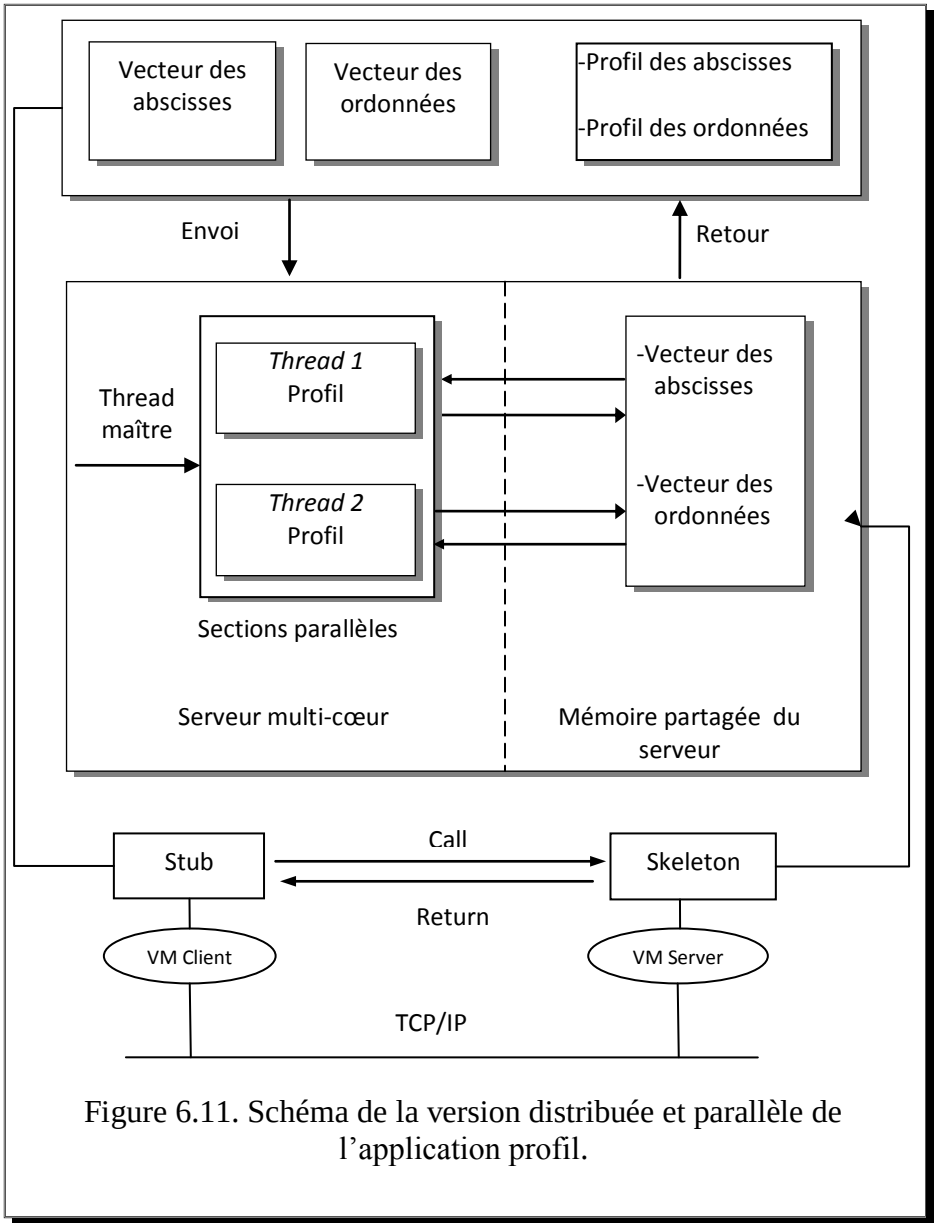
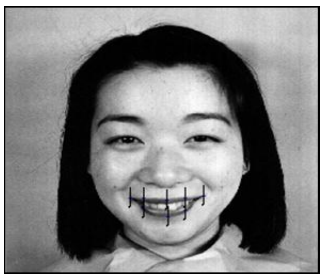


Figure 6.11. Schéma de la version distribuée et parallèle de l'application profil.

(a)

(b) **(c)**



(d) **(e)**

Figure 6.12. (a) Interface principale de l'application, (b) Annotation des contours des lèvres, (c) Image gradient en X, (d) Image gradient en Y, (e) Tracé des intensités des profils.

Pour la majorité des points d'annotation, les formules proposées pour le calcul des coordonnées des points de profil sont vérifiées et confirmées en effectuant le tracé de la perpendiculaire au segment passant par le point d'annotation (figure 6.20). Nos efforts sur la validation des formules nous ont un peu éloignés de l'exécution multithreading de la procédure de parallélisation du calcul des profils.

6.4 Analyse en composantes principales : version multithreading

Nous présentons dans cette section l'application ACP réalisée en C++ version 2010. Nous utilisons la bibliothèque OpenCV spécialisée dans le traitement d'image en temps réel. Nous avons réintégré ensuite l'application dans l'environnement Qt version 5.1.1 afin de doter

l'application d'interfaces graphiques. Qt est une bibliothèque logicielle multiplateforme orientée objet et développé en C++ par Nokia. Nous réutilisons la bibliothèque OpenCV dans l'environnement Qt. Dans cet environnement, nous avons réalisé deux versions de l'application ACP. La première version consiste à traiter toute la matrice des vecteurs de forme. La chaîne de traitement de la première version est détaillée sur la figure 6.13. La deuxième version est dédiée au multithreading afin de traiter séparément la procédure ACP sur deux matrices différentes l'une représentant les abscisses des vecteurs de forme et l'autre représentant la matrice de leurs ordonnées (figure 6.14). Le schéma d'implémentation de la deuxième version est détaillé sur la figure 6.15. La base de données utilisée dans les deux versions est composée de six images de visages de la base de données JAFFE (figure 6.16). La matrice des vecteurs de forme est construite à partir de l'annotation des images de la base de données (figure 6.17). Elle est composée de six vecteurs de forme dont chacun contient huit points d'annotation (figure 6.18) pour la première exécution. Les figures 6.18 et 6.19 illustrent les interfaces de l'application séquentielle. La figure 6.20 illustre l'exécution de la matrice de l'ensemble des vecteurs de forme. La figure 6.21 illustre l'exécution de l'application parallèle et l'estimation du temps d'exécution. Les applications sont exécutées sur une machine portable Corei5 (2.50GHZ et 4 GO). Le paramètre CLOCKS_PAR_SEC est égal à 1000. Nous avons obtenu le temps d'exécution de chaque version à l'aide de la fonction Clock(). Les résultats obtenus sont sauvegardés dans des fichiers textes (figure 6.22). Le pourcentage de variation retenu est de 80%.

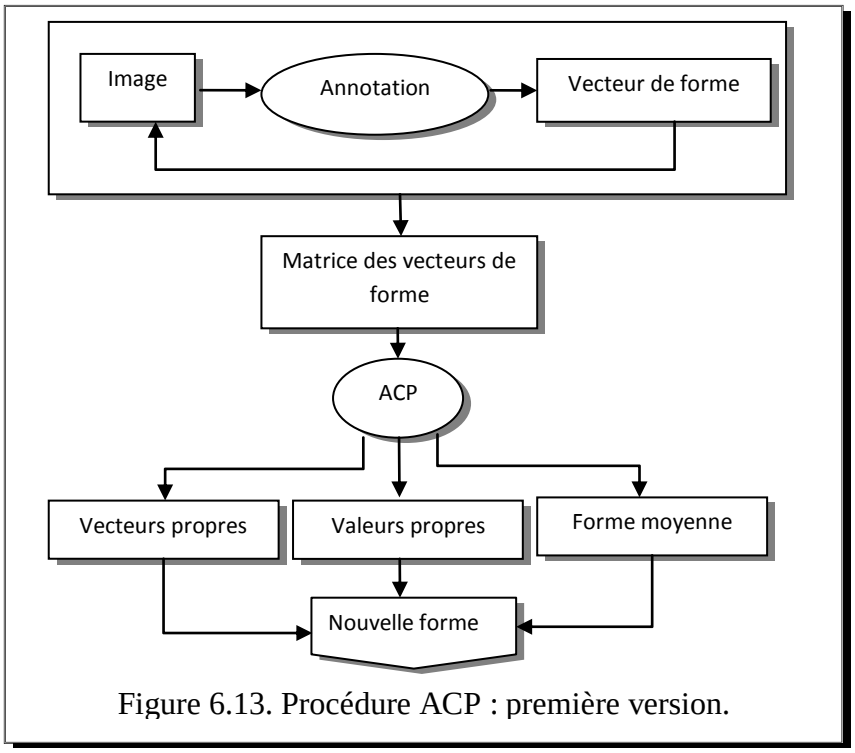


Figure 6.13. Procédure ACP : première version.

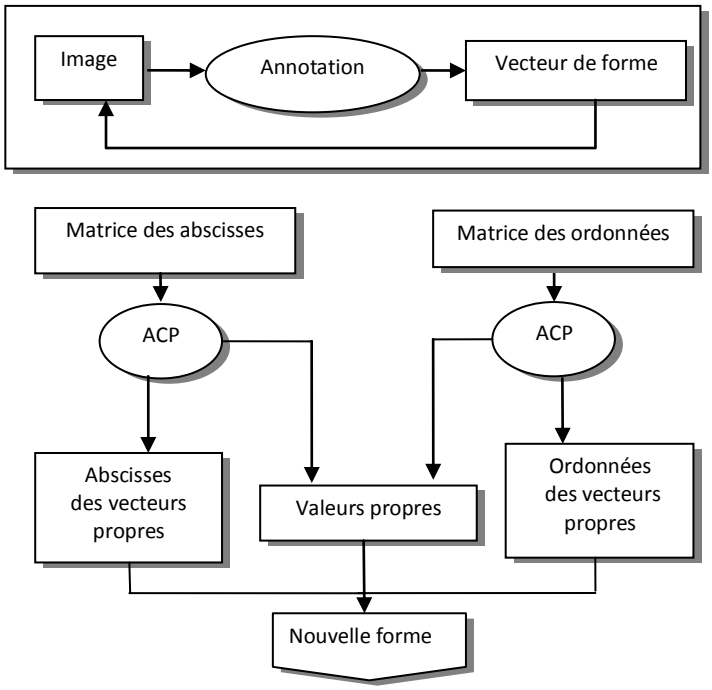


Figure 6.14. Procédure ACP : deuxième version.

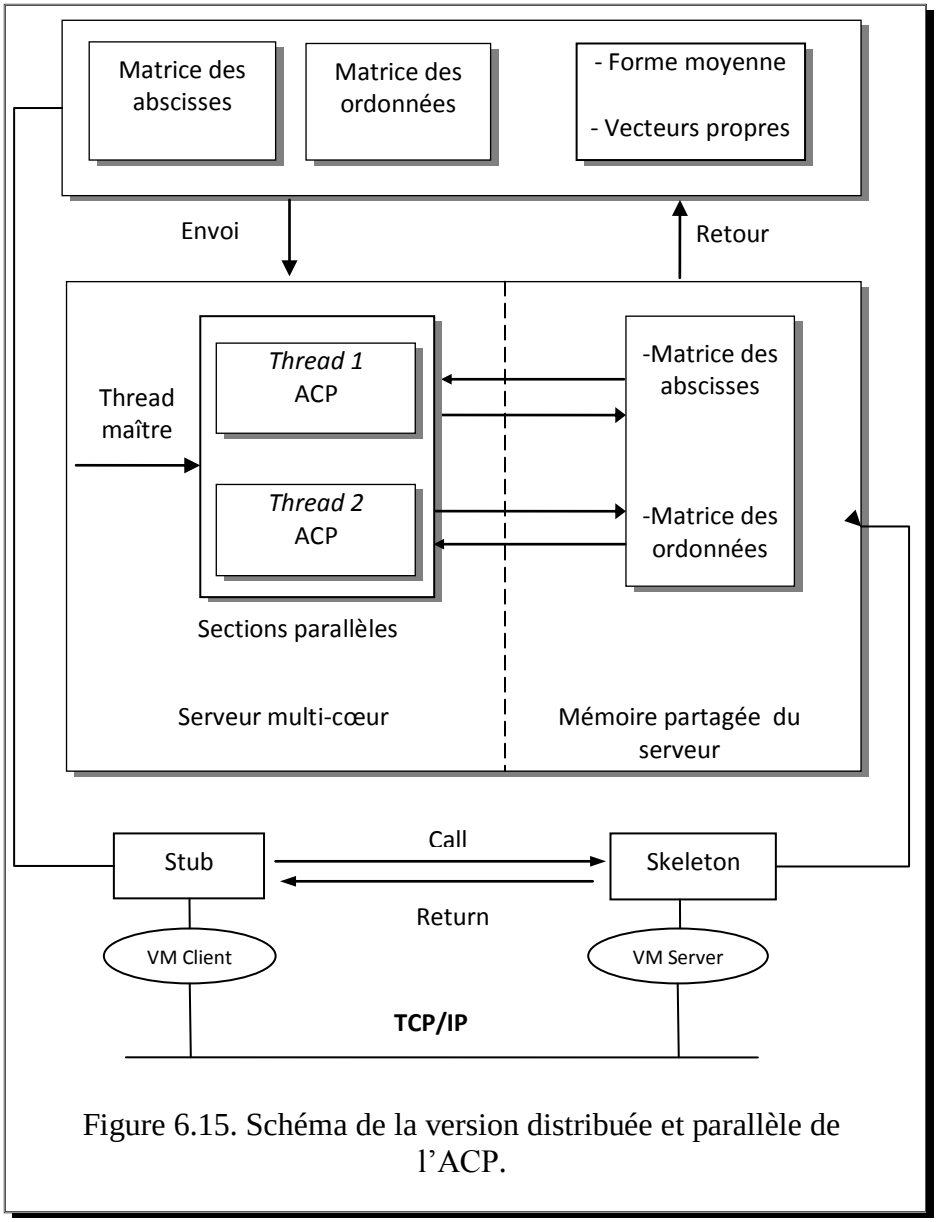


Figure 6.15. Schéma de la version distribuée et parallèle de l'ACP.

Figure 6.16. Base d'images utilisée dans la procédure ACP.

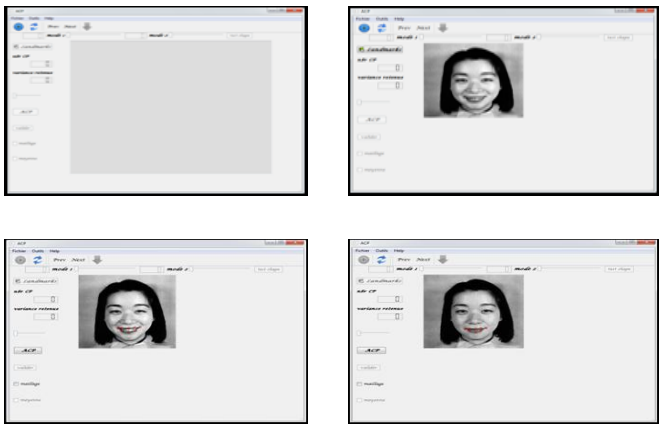


Figure 6.17. Interfaces de l'application ACP : Annotation des lèvres.

Figure 6.18. Les images annotées par huit points d'annotation au niveau du contour des lèvres.

Figure 6.19 Forme moyenne.

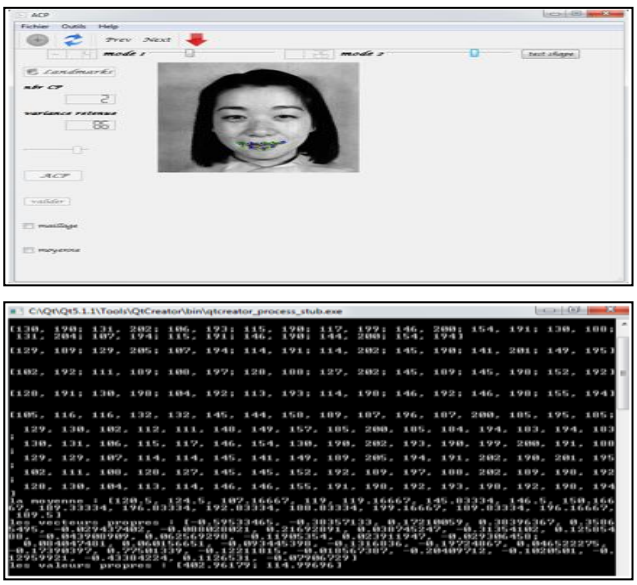


Figure 6.20. Exécution de l'ACP sur une matrice globale.

La table suivante résume un exemple d'application composé des données d'annotation (en lignes), de la forme moyenne, des valeurs propres, des vecteurs propres (en ligne), et d'une nouvelle forme calculée dans la nouvelle base.

Points d'annotation :							
(104,182)	(116,187)	(117,196)	(131,186)	(131,198)	(144,187)	(147,194)	(155,183)
(108,193)	(115,188)	(115,198)	(132,189)	(132,203)	(145,190)	(144,197)	(152,192)
(107,194)	(114,192)	(116,200)	(130,189)	(130,203)	(142,191)	(140,201)	(149,195)
(102,191)	(113,188)	(114,198)	(129,189)	(129,202)	(142,189)	(142,199)	(151,192)
(101,192)	(117,192)	(117,197)	(129,192)	(130,200)	(141,192)	(141,198)	(154,193)
(101,185)	(114,184)	(113,195)	(130,186)	(131,199)	(146,184)	(147,195)	(159,181)
Forme Moyenne :							
(103.83	(114.83	(115.33	(130.17	(130.5	(143.33	(143.5	(153.33
189.5)	188.5)	197.33)	188.5)	200.83)	188.83)	197.33à	189.33)
Valeurs propres : 88.29, 12.23							
Vecteurs propres :							
(-0.127	0.007	-0.049	0.02	0.04	0.138	0.280	0.301
-0.444	-0.261	-0.146	-0.179	-0.172	-0.263	-0.230	-0.563)
(-0.643	0.186	0.145	-0.239	-0.153	-0.307	-0.181	0.235
-0.128	0.252	-0.138	0.269	-0.271	0.149	0.002	0.023)
Valeurs de b : 0.24, 0.06							
Nouvelle forme							
(102,187)	(115,187)	(115,195)	(129,188)	(129,199)	(142,188)	(143,196)	(154,188)

Table 6.1. Données et résultats d'un 1èr exemple d'application.

Les coordonnées obtenues dans cet exemple pour la nouvelle forme calculée semblent significatives et proches des coordonnées des vecteurs de forme donnés, bien que les données initiales n'ont pas été alignées.

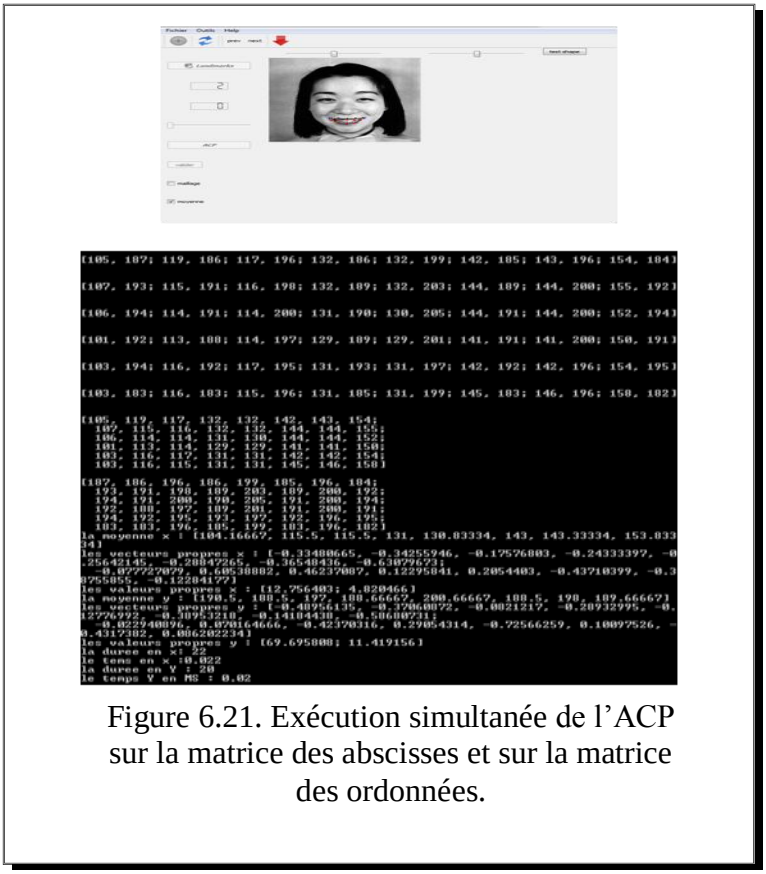


Figure 6.21. Exécution simultanée de l'ACP sur la matrice des abscisses et sur la matrice des ordonnées.

La table 6.2 résume un autre exemple d'application mais cette fois ci appliqué simultanément à deux matrices l'une correspondant aux abscisses des points d'annotation et l'autre correspondant aux ordonnées des mêmes points. La forme moyenne est déterminée à partir de la matrice des moyennes des abscisses et celle des moyennes des ordonnées. Pour chaque matrice, les valeurs propres et les veceturs propres sont calculés. Dans la nouvelle base calculée, une nouvelle forme est déduite et correspond à l'ensemble des nouvelles abscisses et des nouvelles ordonnées.

Abscisses des points d'annotation :							
104	116	117	132	132	144	146	156
107	114	117	131	131	145	145	154
109	116	114	130	130	144	143	151
105	114	112	128	128	142	142	153
102	114	114	129	129	142	142	156
101	113	111	128	132	147	147	159
Moyennes des abscisses :							
104.67	114.50	114.17	129.67	130.33	144	144.17	154.83
Ordonnées des points d'annotation :							
185	186	197	188	200	185	194	184
194	189	200	188	203	191	201	191
193	190	200	189	204	190	202	194
190	189	197	189	200	189	198	192
193	191	195	192	198	193	196	193
183	182	193	185	201	183	196	182
Moyennes des ordonnées :							
189.67	187.83	197	188.5	201	188.5	197.83	189.33
Valeurs propres des abscisses : 16.96, 9.83							
Vecteurs propres des abscisses :							
(-0.609	-0.172	-0.229	-0.131	0.173	0.222	0.292	0.603)
(-0.21	-0.148	-0.544	-0.408	-0.402	-0.309	0.449	-0.104)
Valeurs de b : 0.24, 0.06							
Nouvelles abscisses :							
104.51	114.45	114.08	129.61	130.35	144.03	144.27	154.97
Valeurs propres : 66.42 ; 12.53							
Vecteurs propres des ordonnées :							
(0.512	-0.355	-0.208	-0.192	-0.055	-0.404	-0.243	-0.553)
(-0.046	0.191	-0.407	0.358	-0.545	0.239	-0.555	0.072)
Valeurs de b : 0.24, 0.06							
Nouvelles ordonnées :							
189.78	187.75	196.93	188.48	200.95	188.42	197.74	189.20
Nouvelle forme :							
(104.51	(114.45	(114.08	(129.61	(130.35	(144.03	(144.27	(154.97
189.78)	187.75)	196.93)	188.48)	200.95)	188.42)	197.74)	189.20)

Table 6.2. Données et résultats d'un 2ème exemple d'application.

Les coordonnées obtenues dans cet exemple, pour la nouvelle forme calculée séparément en abscisses et en ordonnées, semblent significatives et proches des coordonnées des vecteurs de forme donnés.

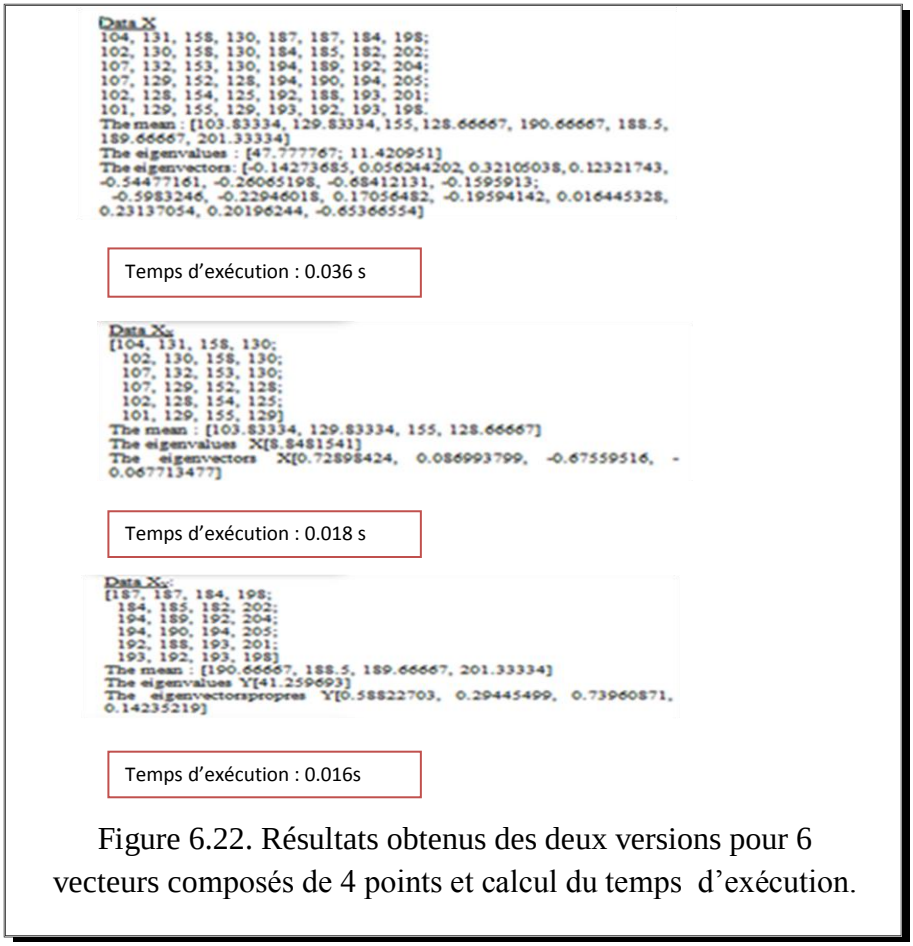


Figure 6.22. Résultats obtenus des deux versions pour 6 vecteurs composés de 4 points et calcul du temps d'exécution.

Dans la version séquentielle de l'ACP appliquée à la matrice des vecteurs de forme de taille 6*16, nous avons relevé un temps d'exécution estimé en moyenne à 0.036 secondes (figure 6.19). Dans la version parallèle multithreading de la procédure ACP effectuée simultanément sur deux matrices chacune de taille 6*8, le temps d'exécution moyen relevé est de 0.018 secondes pour la matrice des abscisses et 0.016 s pour la matrice des ordonnées (figure 6.18). L'accélération pour le test effectué est double (le rapport 0.036/0.018). Nous avons effectué trois tests (table 6.3) et nous remarquons la différence en temps d'exécution entre l'exécution séquentielle et l'exécution simultanée de l'ACP. Cependant il est utile d'envisager plusieurs tests afin d'estimer approximativement la mesure de l'accélération de la procédure d'analyse et de pouvoir relever l'efficacité de la parallélisation de la procédure d'analyse.

Dimension	ACP global	ACP des abscisses	ACP des ordonnées
6*6	0.026	0.015	0.012
6*8	0.036	0.018	0.016
6*12	0.048	0.022	0.032

Table 6.3. Mesures du temps d'exécution.

6.5 Conclusion

Les implémentations que nous avons programmées sont réalisées. Les premières implémentations ont concerné la procédure d'alignement et la procédure d'analyse en composantes principales, dans un environnement distribué orienté objet de type Java-RMI, sans aborder le parallélisme que nous avons traité séparément. Nous avons mené une étude expérimentale sur une base d'images de visage et nous avons ciblé le contour des lèvres. Nous avons ensuite changé d'environnement pour traiter les autres applications. Nous avons utilisé le C++, les bibliothèques Qt et OpenCV avec lesquels nous avons programmé et testé nos différentes applications. Pour l'application calcul des profils, notre objectif primaire était de tester la validité des formules de calcul des points de profil proposées. Dans la deuxième version du calcul de l'ACP, nous avons procédé à une analyse globale de l'ensemble des points, ensuite à une analyse simultanée effectuée sur les abscisses et les ordonnées du même ensemble de points. L'ACP appliquée simultanément aux abscisses et aux ordonnées est équivalente à une ACP appliquée d'une manière globale à l'ensemble des points. Pour chaque version, il est possible de déduire de nouveaux vecteurs admissibles et qui représentent la même donnée. Les données dont nous disposons sont en très petit nombre et il est difficile de tirer des conclusions sur la qualité des opérations traitées à partir de si peu de données. Nous ne prétendons donc pas valider rigoureusement ceux que nous utilisons. Cependant, les résultats obtenus ont tendance à montrer la faisabilité des solutions proposées. Une version préliminaire du travail correspondant à ce chapitre a été publiée dans [Naoui et al. 2013] et [Naoui et al. 2014].

Chapitre 7

Conclusion et Perspectives

7.1 Conclusion et discussion

Dans le cadre de cette étude, nous nous sommes intéressés à la segmentation d'images par les modèles statistiques de forme et d'apparence. Parmi nos travaux, nous pouvons distinguer une partie méthodologique dans laquelle nous présentons la problématique et le contexte de notre mise en œuvre et une partie applicative justifiée par l'implémentation de quelques opérations. Dans la première partie nous avons présenté un état de l'art des méthodes de segmentation d'images. Cette partie nous a permis de constater que le problème de la segmentation d'image n'est pas nouveau et est très vaste. Il en résulte un très grand nombre de méthodes qui ont été et seront à l'origine des travaux scientifiques, et dont la comparaison en termes de structure et/ou de performance est très difficile. Nous avons mis l'accent sur la méthodologie standard mise en jeu pour l'élaboration des modèles statistiques de forme et d'apparence. En aval, de l'extraction des formes, la méthodologie concerne d'une part la représentation des formes et d'apparence et leur mise en correspondance permettant la formation d'un ensemble d'apprentissage, et d'autre part applique des techniques d'analyse statistique de données, permettant de dériver les modèles représentatifs de l'ensemble d'apprentissage. Les modèles obtenus sont génératifs, indiquant qu'il existe une bijection entre l'espace des objets et l'espace des paramètres de forme et d'apparence. La segmentation traitée par cette méthodologie est une approche active de reconnaissance d'un objet, conçue sur la base de mise en correspondance de l'objet recherché et d'une instance du modèle dérivé. Segmenter une image inconnue, revient en fait à déterminer les paramètres des modèles statistiques correspondant au mieux à la réalité. Les différentes extensions proposées et conçues pour les modèles statistiques de forme et d'apparence apportent des réponses aux diverses questions posées à l'étape de modélisation et/ou à celle de la segmentation. Les modèles statistiques de forme et d'apparence tirent leur puissance lors de leur application à des images traitant un problème particulier dans des domaines tels que la reconnaissance faciale et l'imagerie médicale. Ils ont été évalués en termes de précision permettant de conclure qu'ils peuvent atteindre une grande précision de la localisation des points d'annotation, à condition que la base d'apprentissage soit bien adaptée au problème, i.e. que l'objet recherché fasse partie de la base d'apprentissage. Ceci permet d'envisager des applications intéressantes dans le domaine de la synthèse d'images. Le passage en 3D n'est pas aisé et ce cas est hors contexte d'étude de cette thèse.

Dans la deuxième partie, nous avons présenté les travaux passés et présents dans le domaine des modèles de programmation. A notre connaissance aucune approche de segmentation basée sur les modèles statistiques de forme et d'apparence n'a considéré dans son ensemble un paradigme clair qui justifie son implémentation et son application aux images traitées, malgré la maîtrise et l'expertise développées pour la mise en œuvre théorique et pratique de ces approches capables de traiter des formes complexes. Nous nous sommes donc donnés pour objectif la recherche d'une démarche qui puisse exprimer autrement la structure de la méthodologie en vue de la projeter dans un environnement de programmation et de mettre en avant plusieurs aspects liés à son développement pour exploiter pleinement les potentialités offertes par les nouvelles architectures. Nous avons développé plusieurs idées, et nous avons exposé nos contributions sous leur forme générale. Il nous a donc fallu proposer

une version distribuée dédiée aux modèles statistiques de forme et d'apparence. Le schéma proposé est composé de plusieurs blocs correspondant aux opérations essentielles du modèle. Ce schéma permet de renforcer la lisibilité de la méthodologie et de constituer un cadre de développement original pour l'implémentation des algorithmes des modèles statistiques de forme et d'apparence. Dans cette version distribuée, nous avons exploré les potentialités de parallélisation des opérations des modèles à travers quelques propriétés mathématiques. Nous avons procédé à une étude de faisabilité qui a fait l'objet d'une étude approfondie. Le data-parallèle nous a paru en adéquation avec la représentation des données utilisée dans le modèle. Nous avons formalisé une méthode de parallélisation régulière par découpage de la structure des données en deux blocs de même dimension. La méthode a été appliquée aux premières opérations de la phase de modélisation. Nous avons développé les premières briques en implémentant les premières opérations. Les implémentations réalisées sont loin d'être exhaustives et sont destinées uniquement à fournir un point de référence pour notre démarche. Notons ici que la deuxième étape de la méthodologie n'a pas été abordée dans ces travaux et nos travaux sont loin d'être un aboutissement. Nous insistons sur le fait qu'il s'agit d'une étude préliminaire et que de plus amples investigations sont nécessaires à l'amélioration et la validation d'une telle démarche. Nous sommes confiants que ce travail se réalisera graduellement (et collectivement) dans les années à venir et que la conception d'une démarche méthodologique donnera un nouvel élan à l'étude de la segmentation.

7.2 Perspectives

Dans le futur nous souhaitons explorer différentes idées pour compléter notre démarche et donner suite à ces travaux. Les perspectives de ce travail sont donc nombreuses, aussi bien en pratique qu'en théorie. Il nous semble donc raisonnable de continuer nos recherches sur la deuxième étape de la méthodologie de segmentation, et procéder à la parallélisation des autres algorithmes, en particulier l'algorithme de triangularisation et l'algorithme de segmentation. Il serait utile d'identifier le type de parallélisme à appliquer à ces algorithmes dans leur version standard ou dans leur version optimisée. Cela implique la recherche des mécanismes de parallélisation et l'étude de faisabilité pour ces algorithmes. Le cadre d'une parallélisation globale serait donc possible dans un milieu distribué et leur combinaison pourrait donc conduire au développement d'un environnement de traitement dédié aux opérations des modèles. Un développement important de notre démarche dans cette direction paraît donc une évolution naturelle.

Pour notre étude préliminaire, les différents blocs du paradigme proposé peuvent constituer une bibliothèque de fonctions qui peut servir à terme à construire efficacement le processus de segmentation. Chaque bloc peut bénéficier d'une amélioration permettant ainsi de renforcer la qualité l'ensemble du processus de segmentation. La rapidité d'exécution du processus de segmentation est un argument en faveur d'opérations temps réel. La possibilité pour notre paradigme de fonctionner en temps réel permet d'envisager de multiples applications. La finalité est évidemment d'expérimenter notre démarche dans un cadre applicatif, et de reprendre les applications privilégiées des modèles statistiques de forme et d'apparence, à

savoir la segmentation des images médicales et/ou la reconnaissance faciale. Le test de notre démarche sur des bases de données de plus grande taille nous permettra de dégager clairement les points forts et les limitations avant de l'exporter dans un contexte purement applicatif. Nous aurons accès à une vérité terrain plus fiable et nous aurons à affronter la problématique de la segmentation par le biais des images traitées.

Dans Cette même perspective, nous pouvons envisager d'intégrer une opération de vérification et de validation des modèles. En effet, la partie modélisation et la partie segmentation constituent deux problématiques dépendantes l'une de l'autre, et il serait intéressant d'estimer le degré de validité des modèles avant de procéder à l'étape de segmentation. Une validation effective nécessite la mise à disposition de larges bases de données. Il serait intéressant d'appliquer un détecteur lors de la phase d'annotation ou lors de la phase d'initialisation de l'opération de segmentation. L'extraction automatique des informations à priori du modèle serait un atout pour un processus de segmentation efficace. En effet, plusieurs travaux décrivent le concept de points d'intérêts et de multiples méthodes robustes sont développées pour les détecter. Cela serait particulièrement pertinent pour l'annotation automatique des images.

Table des figures

Liste des tables

Bibliographie

[Agueb et al. 2011] Amel Agueb, Dalila Benhamed. 'Implémentation distribuée de la procédure d'alignement des formes d'une base d'image', Mémoire de master, Institut d'Informatique, Université d'Es_Sénia, Oran, 2011.

[Aidarous et al. 2007] Y. Aidarous, S. Le Gallou, A. Sattar, R. Séguier. 'Face alignment using active appearance model optimized by simplex'. International Conference on Computer Vision Theory and Applications (VISAP'07), 2007.

[Allard 2005] Jeremy Allard, 'FlowVR : Calculs intéractifs et visualisation sur grappe', Thèse de Doctorat, Institut national polytechnique de Grenoble, 25 novembre 2005.

[Al-Zubi et al. 2003] Stephen Al-Zubi, Klaus Toennie. 'Generalzing the Active Shape Model by integrating Structural Knowledge to Recognize hand Drawn Sketches', Computer Analysis of Images and Patterns, 10th International Conference, CAIP 2003, Groningen, The Netherlands, 2003.

[Angella 2001] Franck Angella. 'Modèles déformables et systèmes particulaires. Application à l'extraction de structures arborescentes en analyse d'images', Thèse de Doctorat, Université de Bordeaux I, janvier 2001.

[Arbelaez 2005] Pablo Andrès Arbeláez Escalente. 'Une approche métrique pour la segmentation d'images', Thèse de Doctorat, Université Paris Dauphine, novembre 2005.

[Bader et al. 1996] D.A.bader J.Jaja. 'Parallel Algorithms for Image Histogramming and Connected Components with an Experiment Study', Journal of Parallel and Distributed Computing 35, 173-190, 1996.

[Bartoli 2008] Adrian Bartoli. 'Contribution au recalage d'images et à la reconstruction 3 D de scènes rigides et déformables', synthèse des travaux sur 2004-2007, Habilitation à diriger des recherches, Université Blaise Pascal, juin 2008.

[Beaudoin 2001] Christian Beaudoin. 'Une approche semi automatique pour la segmentation des tumeurs hépatiques dans les images de résonnance magnétique', Thèse de Doctorat, Université du Québec, août 2001.

[Besbes 2010] Ahmed Besbes. 'Image segmentation using MRFs and Statistical Shape Modeling', Thèse de Doctorat, Ecole centrale Paris, septembre 2010.

[Beucher 1990] Serge Beucher. 'Segmentation d'images et morphologie mathématique', Thèse de Doctorat, Ecole supérieure des mines de Paris, décembre1990.

[Boillod-Cerneux 2014] France Bollod-Creneux. 'Nouveaux algorithmes numériques pour l'utilisation efficace des machines multi-cœurs hétérogènes', Thèse de doctorat, Université de Lille, 13 octobre 2014.

[Bordei 2013] Cristina Bordei, Pascal Bourdon, Bertrand Augereau, Philippe Carré. 'Une texture polynomiale pour les modèles actifs d'apparence', XXIVème colloque, Brest, France, 2013.

[Bricq 2008] Stéphane Bricq. 'Segmentation d'images IRM anatomiques par inférence bayésienne multimodale et détection de lésions', Thèse de Doctorat, Université de Louis Pasteur – Strasbourg I, novembre 2008.

[Capri 2007] Arnaud Capri. 'Caractérisation des objets dans une image en vue d'une aide à l'interprétation et d'une compression adaptée au contenu application aux images échographiques', Thèse de Doctorat, Université d'Orléans, 2007.

[Chabrier 2005] Sébastien Chabrier. 'Contribution à l'évaluation de performances en segmentation d'images', Thèse de doctorat, Université de Bourges, 14 décembre 2005.

[Ciofolo 2005] Cybèle Ciofolo. 'Segmentation de formes guidées par des modèles en neuro-imagerie. Intégration de la commande floue dans une méthode de segmentation par ensembles de niveau', Thèse de Doctorat, Université de Rennes I, décembre 2005.

[Cocquerez et al. 1995] J.-P. Cocquerez et S. Philipp. ' Analyse d'Images : filtrage et segmentation'. Masson, 1995.

[Cootes et al. 1995] T.F Cootes, Taylor, Cooper, Graham. 'Active Shape Models – Their Training and Applications', Computer Vision and Image Understanding, 61(1), pp 38-59, 1995.

[Cootes et al. 1997] T.F Cootes, C.J.Taylor. 'A Mixture Model for Representing Shape Variation', Image and Vision Computing, 17(8), pp 567-574, 1997.

[Cootes et al. 1998a] T.F Cootes, G.J Edwards, and C.J Taylor, 'Active appearance models', Proc. IEEE European Conference on Computer Vision (ECCV '98), p. 484, 1998.

[Cootes et al. 1998b] T.F Cootes, G.J. Edwards, C.J.Taylor. 'A comparative evaluation of Active Appearance Model algorithms', British Machine Vision Conference Proceeding, Vol.2, pages 680-689, 1998.

[Cootes et al. 1999] T.F Cootes, C.J.Taylor. 'Comparing Active Shape Models with Active Appearance Models', British Machine Vision Conference Proceeding, Vol.1, 1999.

[Cootes et al. 2001] T.F Cootes, G.J Edwards, and C.J Taylor, 'Active appearance models', Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.23, No.6, June 2001, pp. 681-685.

[Cootes et al. 2002] TF Cootes, Kittipanya-ngam. 'Comparing variations on the active appearance model algorithm', British Machine Vision Conference, Vol.2, pp. 837-846, 2002.

[Corouge 2003] Isabelle Corouge. ‘Modélisation statistique de formes en imagerie cérébrale’, Thèse de Doctorat, Université de Rennes I, avril 2003.

[Cousty 2007] Jean Cousty. ‘Ligne de partage des eaux discrètes : théorie et application à la segmentation d’images cardiaques’, Thèse de Doctorat, Université de Marne–la-Vallée, octobre 2007.

[Cristinacce 2008] D. Cristinacce and T. Cootes, ‘Automatic feature localization with constrained local models’ Pattern Recognition, vol. 41, No. 10, pp. 3054–3067, 2008.

[Dupas 2009] Alexandre Dupas. ‘Opérations et algorithmes pour la segmentation topologique d’images 3D’, Thèse de Doctorat, Université de Poitiers, décembre 2009.

[Faux 2009] Francis Faux. ‘Détection et suivi de visage par la théorie de l’évidence’, Thèse de doctorat, Université de Pau et des Pays de l’Adour, octobre 2009.

[Fernandes 2002] L .G.L Nicolas. ‘Parallélisation d’un algorithme d’appariement d’images quasi-dense’, Thèse de Doctorat, Institut National Polytechnique de Grenoble, 08 juillet 2002.

[Fiorio 1995] Christophe Fiorio. ‘Approche interpixel en analyse d’images, une topologie et des algorithmes de segmentation’, Thèse de Doctorat, Université de Montpellier, novembre 1995.

[Flynn 1972] M.Flynn. ‘Some computer organizations and their effectiveness’ IEEE Transactions on Computers, Vol. C-21(9), pages 948-960, 1972.

[Gacon 2006] Pierre Gacon. ‘Analyse d’images et modèle de formes pour la détection et la reconnaissance. Application aux visages en multimédia’, Thèse de Doctorat, Institut national polytechnique de Grenoble, juillet 2006.

[Gao et al. 2010] Xinbo Gao, Xuelong Li, Dacheng Tao, T., Kim, D. ‘A Review of Active Appearance Models’, IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, Vol. 40, No. 2 , March 2010, pages 145–158.

[Gastaud 2005] Muriel Gastaud. ‘Modèle de contours actifs pour la segmentation d’images et de vidéos’, Thèse de Doctorat, Université de Nice-Sophia Antipolis, décembre 2005.

[Ginneken et al. 2002] B. van Ginneken, A.F frangi, J.J. Stall, B. ter Haar Romeny. ‘Active Shape Model Segmentation with Optimal Features’, IEEE-TMI, Vol.21, pp. 924-933, 2002.

[Goodall 1991] C. Goodall. ‘Procrustes methods in the statistical analysis of shape’, Jour. Royal Statistical Society, Series B, 53(2):285–339, 1991.

[Grand-Brochier 2011] Manuel Grand-Brochier. ‘Descripteurs 2D et 2D+t de points d’intérêt pour des appariements robustes’, Thèse de Doctorat, Université de Clermond-Ferrant, 18 novembre 2011.

[Guigues 2003] Laurent Guigues. ‘Modèles multi-échelle pour la segmentation d’images’, Thèse de Doctorat, Université de Cergy Pontoise, décembre 2003.

[Hachama 2008] Mohamed Hachama. ‘Modèles de recalage classifiant pour l’imagerie médicale’, Thèse de Doctorat, Université de Paris 5, 2008.

[Hanifi 2000] Djamel Hanifi. ‘Générateur d’architectures pour plate forme reconfigurable dédié au traitement d’images’, Thèse de Doctorat, Université de Grenoble, 2000.

[Herbulot 2007] Ariane Herbulot. ‘Mesures statistiques non-paramétriques pour la segmentation d’images et de vidéos et minimisation par contours actifs’, Thèse de Doctorat, Université de Nice, 2007.

[Hou et al. 2001] X.W. Hou, S.Z. Li, H.J. Zhang, Q.S. Cheng. ‘Direct appearance models’, Conference on Computer Vision and Pattern Recognition (CVPR'01), Vol.1, pages 828-833, 2001.

[Houeix 1988] Pierre Houeix. ‘Evaluation de performances d’une architecture parallèle pour le traitement d’images’, Thèse de Doctorat, Institut national polytechnique de Grenoble, septembre 1988.

[Kittipanya et al. 2006] P.Kittipanya-ngam, T.F.Cootes. ‘The effect of texture representations of aam performance’, International Conference on Pattern Recognition, Vol.1, pp. 328-331, 2006.

[Lambert 2002] Patrick Lambert. ‘Etudes méthodologiques du filtrage et de la segmentation d’images multi-composantes’, Habilitation à diriger des recherches, Université de Savoie, juillet 2008.

[Landré 2005] Jérôme Landré. ‘Analyse multirésolution pour la recherche et l’indexation d’image par le contenu dans les bases de données images. Application à la base d’images paléontologique Trans ‘Tyfipa’, Thèse de Doctorat, Université de Bourgogne, décembre 2005.

[Lecœur et al. 2007] Jérôme Lecoeur, Christian Barillot. ‘Segmentation d’images cérébrales : Etat de l’art’, Rapport de recherche, Institut national de recherche en informatique et en automatique, juillet 2007.

[Le Gallou 2007] Sylvain Le Gallou. ‘Détection robuste des éléments faciaux par modèles actifs d’apparence’, Thèse de doctorat, Université de Rennes I, novembre 2007.

[Li et al. 2005] Yuanzhong Li, Wataru Ito. ‘Shape Parameter Optimization for AdaBoosted Active Shape Model’, 10th IEEE International Conference on Computer Vision, Vol.1, pp. 251-258, 2005.

[Martins et al. 2008] Pedro Martins, Joana Sampaio and Jorge Batista. ‘Facial Expression Recognition using Active Appearance Model’, VISAPP 2008 - International Conference on Computer Vision Theory and Applications.

[Matthews et al. 2004] L. Matthews and S. Baker. 'Active appearance models revisited', International Journal of Computer Vision, 60(2), pages 135-164, 2004.

[Mercier 2007] Hugo Mercier. 'Modélisation et suivi des déformations faciales. Applications à la description des expressions du visage dans le contexte de la langue des signes', Thèse de doctorat, Université de Toulouse III, mars 2007.

[Meurie 2005] Cyril Meurie. 'Segmentation d'images couleur par classification pixellaire et hiérarchie de partitions', Thèse de doctorat, Université de Caen/ Basse-Normandie, octobre 2005.

[Millborrow 2008] S. Milborrow and F. Nicolls, 'Locating facial features with an extended active shape model', Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR '08), p. 504, 2008.

[Montagnat 1999] Jean Montagnat. 'Modèles déformables pour la segmentation et la modélisation d'images médicales en 3D et 4D', Thèse de Doctorat, Université de Nice-Sophia Antipolis, décembre 1999.

[Naoui et al. 2013] Oumelkheir Naoui, Ghalem Belalem, Mahmoudi Saïd. 'A reflexion on Implementation version for active appearance model', International Journal of compuer Vision and image processing, 3(3), 16-30, July-September 2013.

[Naoui et al. 2014 to appear] M.Naoui, S. Mahmoudi, G. Belalem. 'Towards a Distributed and Parallel Schema for Active Appearance Model Implementation'. International Journal of computational Vision and Robotics,Inderscience. Special Issue on: "Recent Advances in Signal and Image Processing", To appear.

[Paraiso 2014] Fawaz Paraiso. 'Une plate forme multi-nuages distribuée pour la conception, le déploiement et l'exécution d'applications distribuées à large échelle', Thèse de doctorat, Université de Lille, 18 juin 2014.

[Ospici 2013] [Mattieu Ospici. 'Modèles de Programmation et d'exécution pour les architectures parallèles et hybrides', Applications à des codes de simulation pour la physique'', Thèse de Doctorat, université de Grenoble, 03 juillet 2013.

[Rhomdani et al. 1999] S. Rhomdani, S. Gong, A.Psarrou. 'A Multi-view Non-linear Active Shape Model using Kernel PCA', 10th British Machine Vision Conference, Vol.2, pp 483-492, 1999.

[Rogers et al. 2002] M. Rogers, J. Graham. 'Robust Active Shape Model Search', 7th European on Computer Vision, Vol.4, pp 517-530 Springer, 2002.

[Rousselle 2003] Jean-Jacques Rousselle. 'Les contours actifs, une méthode de segmentation. Application à l'imagerie médicale', Thèse de Doctorat, Université de Tours, juillet 2003.

[Saidani 2012] Tarik Saidani. ‘Optimisation multi-niveau d’une application de traitement d’images sur machines parallèles’, Thèse de Doctorat, Université de Paris-sud, 06 novembre 2012.

[Schmid et al. 1998] C. Schmid, R. Mohr et C. Bauckhage. ‘Comparing and Evaluating Interest Points’, IEEE International Conference on Computer Vision, pages 230–235, 1998.

[Schmid et al. 2000] C. Schmid, R. Mohr, C. Bauckhage. ‘Evaluation of Interest Point Detectors’, International Journal of Computer Vision, 37(2) :151–172, 2000.

[Séguier et al. 2008] Renaud Séguier, Sylvain Le Gallou, Gaspard Breton, Christophe Garcia. ‘Modèles actifs d’apparences adaptés’, Traitement du signal, Vol 25, No 5, 2008.

[Serra 2003] Jean Serra, ‘Connexions et segmentation d’images’, Traitement du signal, Vol 20, No 3, p. 243-254, 2003.

[Sicard 2004] Nicolas Sicard. ‘ANET : un environnement de programmation à parallélisme de données pour l’analyse d’image’, Thèse de Doctorat, Université de Paris-sud, Orsay, 12 juillet 2004.

[Skillicorn et al. 1998] D. Skillicorn, D. Talia. ‘Models and languages for parallel computation’, ACM Computing Surveys, 30(2): pp. 1239-169, 1998.

[Stegmann 2000] Mikkel Bille Stegmann. ‘Active Appearance Model. Theory, extensions & cases’, Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, August 2000.

[Stegmann, 2004] Mikkel Bille Stegmann. ‘Generative interpretation of medical images’, PHD thesis, Informatics and Mathematical Modeling, Technical University of Denmark, DTU, 2004.

[Sung et al. 2007] Sung, J., Kanade, T., Kim, D. ‘A Unified Gradient-Based Approach for Combining ASM into AAM’, International Journal of Computer Vision 75(2), 297–309, 2007.

[Velut 2007] Jérôme Velut. ‘Segmentation par modèles déformables surfaciques localement régularisé par spline lissante’, Thèse de Doctorat, Université de Cergy Pontoise, décembre, 2007.

[Zhou et al. 2003] Y. Zhou, L. Gu, H.Z. Zhang. ‘Bayesian Tangent Shape Model: Estimating Shape and Pose Parameters via Bayesian Inference’, Computer Vision and pattern Recognition, 2003.

[Zouagui 2004] Tarik Zouagui. ‘Approche fonctionnelle générique des méthodes de segmentation d’images’, Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon, 08 octobre 2004.

Publications

1.M. Naoui, G. Belalem, S. Mahmoudi. ‘A reflexion on Implementation version for active appearance model’. International Journal of computer Vision and image processing (IJCVIP), 3(3): 16-30, July-September 2013.

2.M. Naoui, S. Mahmoudi, G. Belalem. ‘Towards a Distributed and Parallel Schema for Active Appearance Model Implementation’. International Journal of computational Vision and Robotics,Inderscience. Special Issue on: "Recent Advances in Signal and Image Processing”, To appear.

http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijcvr

3. M. Naoui, S. Mahmoudi, G. Belalem, O. Kemmar. ‘Distributed and Parallel Environment to implement the Standard Version of AAM Model’, Journal of Information Processing Systems (JIPS), ISSN: 1976-913X(Print), ISSN: 2092-805X(Online), Accepted with Revisions.

4. M.Naoui, M.Saïd, G.Belalem. O.Kemmar ‘A Data parallelism to implement the standard version of AAM model: Some contributions’. Submitted. Journal WSCG.