REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre :

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Image et Vie Artificielle**

Evaluation parallèle des méthodes de segmentation d'images.

Par :
KATEB NOUR

Soutenu le , devant le jury composé de :

	MAA	Président
DJEROU Leila	Professeur	Rapporteur
	MCB	Examinateur

Remerciement

Je tiens premièrement à remercier ALLAH le tout puissant de m'avoir donné le courage et la patience pour terminer ce modeste travail.

*Je tiens tous d'abord à remercier mon encadreur le professeur « **Djerou Leila** » pour sa patience, ses encouragements, ses conseils, et le temps précieux quelle ma accordé pour l'achèvement de ce travaille.*

Mes grands remerciements à tous les enseignants et collègues pour leur accompagnement, aide, encouragement et soutient tout le long de la réalisation de ce travaille.

Sans oublier tous les membres de ma famille pour leurs compréhensions, leurs soutiens, et leurs encouragements.

Dédicace

Je dédie ce modeste travail à :

Ma mère Fadila

Mon père Mohamed

Mon époux MeftahTarek

Et mes enfants Ranim, Abderrahim, Rinad, Ilaf, et ma petite Layane .

.

A mes sœurs, mes frère et leurs familles, ainsi qu'à toute ma famille et mes amies.

Résumé

Dans le traitement et l'analyse d'images médicales, une tâche assez fréquente est la comparaison des performances de différents algorithmes de segmentation pour détecter des zones d'intérêts dans les images médicales. Sur d'énormes bases de données d'images, cette tâche est généralement longue et fastidieuse. Dans notre travail, nous nous intéressons au problème d'évaluation et de comparaison des performances d'un ensemble d'algorithmes de segmentation à base de seuillage, dans le cadre de détection automatique des zones d'intérêt dans des images médicales, dans une structure de traitement parallèle, en exploitant le paradigme multithreading de programmation parallèle pour réduire le temps de calcul.

Mots cles : Evaluation de qualite de segmentation, Methodes de seuillage, Programmation parallèle, Segmentation d'image.

Table des matières

Table des matières

Table des figures

Introduction générale

En imagerie médicale, la segmentation est une étape primordiale qui consiste à extraire, à partir de l'image, une ou plusieurs régions formant la zone d'intérêt. Les méthodes de segmentation sont diverses et dépendent en grande partie de la modalité d'imagerie et des caractéristiques propres aux structures que l'on souhaite segmenter. La plupart des travaux présentent des approches qui sont bien adaptées à une application particulière ou à un type défini d'images médicales. La comparaison entre ces approches est difficile pour diverses raisons. Par exemple, en imagerie cérébrale, certaines techniques séparent tout simplement le cerveau de la tête, d'autres essaient d'identifier chaque classe de tissu et d'autres visent à extraire la structure anatomique. En plus, certaines techniques font intervenir des connaissances a priori et d'autres ne le font pas. Pour cela, l'évaluation de la qualité de segmentation est indispensable afin de permettre le choix d'un algorithme et de régler ses paramètres. Il serait aussi possible de fusionner des résultats de segmentations en prenant en compte les valeurs des critères d'évaluation comme indice de confiance pour la fusion. Toutefois, l'évaluation de la segmentation reste un sujet très délicat, aux limites floues, aux enjeux multiples, aux protocoles expérimentaux non consensuels.

On distingue deux classes d'approches d'évaluations selon que l'on possède (évaluation non supervisée) ou pas (évaluation supervisée) une segmentation de référence. La plupart des études réalisées dans la littérature favorisent évaluation supervisée qui se base sur les critères utilisant une vérité terrain. La robustesse d'une telle approche dépend essentiellement de l'exactitude de la vérité terrain et de la précision de la métrique d'évaluation utilisée. Pour cela, le tracé interactif de la segmentation désirée par des experts a souvent été la seule approche acceptable. Toutefois, ces critères souffrent de plusieurs limites qui sont principalement liés à la dépendance de la performance humaine des experts (variabilités intra-expert et inter-expert) et à la présence et l'influence du bruit [5]. De là, plusieurs algorithmes ont cherché à enlever la variabilité présentée par des experts, soit en combinant les avis des experts sur la qualité de la segmentation étudiée, en utilisant principalement des évaluations probabilistiques et statistiques , soit en définissant une vérité terrain qui est le plus souvent la moyenne des segmentations de références tracées par un consensus d'experts à plusieurs reprises , et lorsque les scientifiques terminent leurs travaux, il existe un désir général d'obtenir des résultats instantanément, plutôt que de passer un temps disproportionné à exécuter des tests pour comparer leur méthode avec celles d'autres auteurs. Cependant, ces tests sont nécessaires pour prouver leur concept et, dans certains cas, pour obtenir des résultats intermédiaires leur permettant de poursuivre leurs travaux dans la bonne direction.

Le processus d'évaluation par comparaison avec une vérité terrain doit être effectué individuellement. Cela signifie que, pour chaque algorithme à tester, plusieurs évaluations utilisant de métriques à base de régions et de contours [5], sont effectuées, une pour chaque mesure. Si le processus doit être exécuté sur d'énormes ensembles de données, l'évaluation d'un algorithme par plusieurs métriques est une tâche fastidieuse pour le chercheur et entraîne des coûts élevés de traitement informatique. Pour cela, les nouvelles architectures de calcul organisent des cœurs de traitement supplémentaires sur la même puce, qui caractérisent les processeurs multicœurs et les multiprocesseurs à puce. Ils permettent d'exécuter ces segmentations dans les cœurs simultanément à partir de différents threads (processeurs multithreads). Les performances des programmes sur ces processeurs dépendent de la capacité de ces programmes à générer plusieurs threads à exécuter simultanément. Les processeurs multicœurs offrent de nombreuses possibilités pour les programmes parallèles basés sur le traitement multithreads , ceci donne une grande aide pour tout scientifique qui doit passer par une phase de test similaire. Il rend plus facile que jamais le déploiement du parallélisme dans les parties intensives en calcul d'un programme ou, lorsque les performances de différents algorithmes doivent être testées dans le même système, La parallélisation fournit une forme simple de vérification du programme, économise du temps (68,54%) et réduit l'effort global de 67%. Ainsi, nous espérons que les scientifiques et les chercheurs en général profiteront de cette méthode dans leur propre environnement [12].

Dans notre travail, nous nous intéressons au problème d'évaluation et de comparaison des performances d'un ensemble d'algorithmes de segmentation, dans le cadre de détection automatique des zones d'intérêt dans des images médicales, dans une structure de traitement parallèle, en exploitant le paradigme multithreading de programmation parallèle pour réduire le temps de calcul. De ce fait, ce mémoire comporte quatre chapitres organisés comme suit :

- Dans le premier chapitre, nous allons présenter une étude détaillée sur l'évaluation des algorithmes de segmentation d'image,
- Dans le second, nous allons donner un aperçu général sur les architectures parallèles en présentant les différents types de parallélisme
- Ensuite dans le troisième, nous allons décrit la méthode proposée pour l'évaluation de certaines méthodes de segmentation d'image à base de seuillage, la conception du système à réaliser et son architecture globale et détaillée.
- L'implémentation du système et les expérimentations et les résultats sont abordés dans le quatrième chapitre.
- Le mémoire est terminé par une conclusion générale avec les perspectives envisagées.

CHAPITRE 1 :

Evaluation de la segmentation d' images

1. Introduction

La segmentation est une étape essentielle en traitement d'images dans la mesure où elle conditionne l'interprétation qui va être faite sur ces images. De nombreux algorithmes ont ainsi été proposés durant les dernières décénnies,Ils sont basés sur différentes approches contour, région, texture.000 Devant la multitude des méthodes proposées, le problème de l'évaluation de la qualité de la segmentation devient primordial. Habituellement, l'éfficacité d'un nouvel algorithme est illustrée par la présentation de quelques résultats de segmentation, ce qui n'autorise que des conclusions subjectives et qualitatives sur les performances de cet algorithme. Il n'existe pas à l'heure actuelle de méthode absolue et générique pour effectuer cette tâche d'évaluation [1] .

Dans ce chapitre, nous allons faire une étude détaillée sur le problème d'évaluation des algorithmes de segmentation d'image, en spécifiant les différentes techniques utilisées dans ce domaine.

2. Définition de la segmentation

Si l'on reprend le schéma classique d'une chaîne de traitement (voir FIG. 1.1), la segmentation est une étape préalable à l'interprétation.

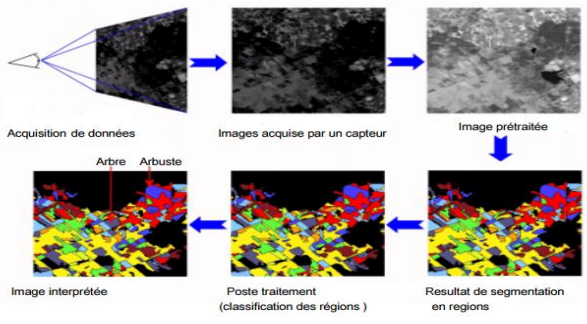


FIG 1.1 La chaîne de traitement d'une image [1] .

Un point de désaccord entre les différents chercheurs de la communauté est la définition des objectifs de la segmentation. Si nous revenons à l'origine historique de la segmentation, elle a

grandement été inspirée par le système de perception visuelle humaine car l'objectif était de faciliter une interprétation automatique d'une image de manière similaire à une interprétation humaine. Le système de perception visuelle humaine utilise les notions de similarité et de différence afin de localiser les objets d'une scène ainsi que leurs frontières ou contours. Ces deux notions ont donné naissance aux deux approches couramment utilisées dans le domaine de la segmentation, à savoir respectivement les approches dites "région" et "frontière", La segmentation fait référence aux notions de différence et de similarité comme les perçoit le système visuel humain et ceci donne naissance à deux approches couramment qualifiées d'approche frontière et d'approche région.

Un bon résultat de segmentation doit vérifier les quatre principes suivants :

a.Les régions d'un résultat de segmentation d'une image doivent être uniformes et homogènes par rapport à des caractéristiques comme le niveau de gris ou la texture.

b.L'intérieur des régions doit être simple et sans trop de trous.

c.Les régions adjacentes d'un résultat de segmentation doivent avoir des valeurs significativement différentes par rapport aux caractéristiques pour lesquelles elles sont uniformes.

d.Les frontières de chaque contour doivent être simples, régulières et doivent être précises spatialement".

La segmentation fait partie de l'interprétation et n'est pas une étape à part ,La segmentation d' une image consiste à la subdiviser en parties la constituant et à extraire ces régions d'intérêt (les objets). Les objectifs de la segmentation sont liés à l'interprétation réalisée ultérieurement , La segmentation d'images est un processus de bas niveau qui a pour objectif de partitionner une image en régions homogènes. La manière dont l'homogénéité des régions est définie dépend de l'application , l'étape de segmentation et celle d'interprétation ne sont pas uniquement liées, mais confondues .En fait, les applications d'interprétation d'images sont juste des algorithmes de segmentation complexes .

En fait ,il existe différentes catégories de méthodes de segmentation (voir figure FIG1.2) donc autant de définitions. La différence fondamentale entre toutes ces méthodes est la quantité de connaissance a priori exploitée pour la segmentation d'une image. Ainsi, la frontière entre la segmentation et l'interprétation est assez difficile à définir [1] .

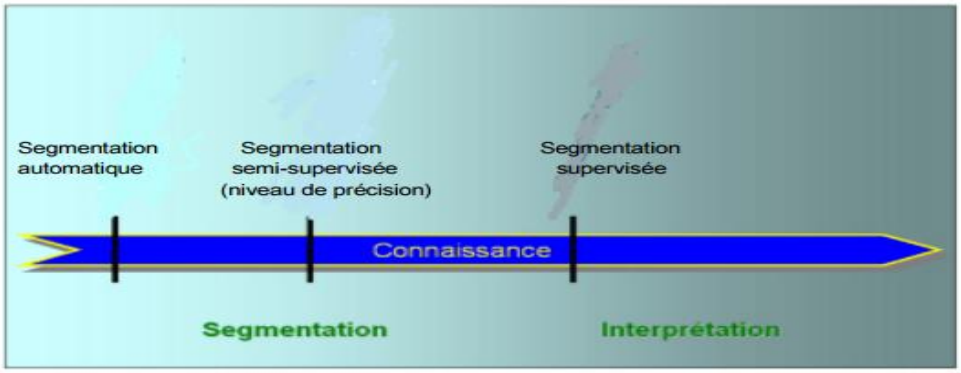


FIG. 1.2 Les différentes catégories de méthodes de segmentation [1].

3. Approches d'évaluation des algorithmes de segmentation d'image

Deux catégories de critères d'évaluation, à savoir l'évaluation supervisée et non supervisée.

3.1. Evaluation supervisée

Un résultat de segmentation est évalué en prenant en compte des connaissances a priori. Ces connaissances peuvent être une vérité terrain (segmentation de référence) ou bien des connaissances sur les éléments à identifier. Les vérités terrains peuvent être obtenues grâce à des experts ou bien en créant des images synthétiques . Cette connaissance pour l'évaluation permet de prendre en compte de façon explicite les objectifs éventuels de la segmentation. Cette approche permet une évaluation d'un résultat pour les différentes catégories de méthodes de segmentation détaillées précédemment [5] .

3.1.1 Évaluation de la détection de frontières

Dans le cas de résultats de segmentation sous la forme de frontières, l'évaluation est généralement réalisée à l'aide d'une métrique de similarité entre les contours extraits par la méthode de segmentation et ceux de la vérité terrain. L'erreur quadratique moyenne (Root Mean Square) est sans doute l'une des premières mesures de divergence .ces mesures de distance peuvent être complétées par différentes distances provenant de l'interprétation probabilistique d'image.Ces mesures prédisent de façon très imprécise les déformations perceptibles. Comme elles n'intègrent pas l'information

spatiale sur les pixels, elles peuvent donner lieu à des valeurs aberrante [1] .

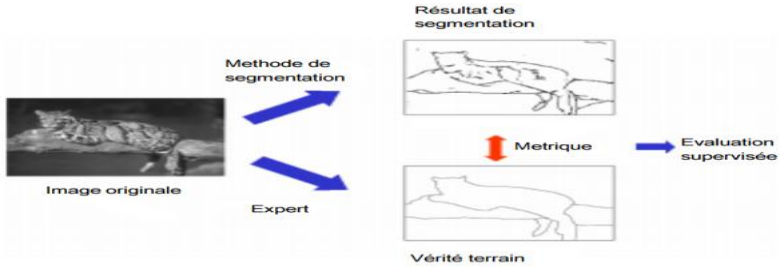


FIG. 1.3 Évaluation supervisée d'un résultat de segmentation à l'aide d'un tracé expert [1].

Le séquençage L'expression mathématique de ce critère de qualité a été obtenue de manière simple et intuitive et possède une très bonne efficacité de calcul. Cette mesure est très souple car elle peut être adaptée à des critères plus spécifiques d'appréciation visuelle grâce à ses sept coefficients mais un entraînement préalable est nécessaire

3.1.2 Évaluation de la segmentation en régions

Des critères permettant de mesurer la dissimilarité entre un résultat de segmentation en régions/ classes par rapport à un résultat de segmentation de référence . Une série de mesures d'erreur de classi fication lorsque les classes sont supposées connues, c'est à dire qu'il faut connaître pour chaque classe d'un résultat de segmentation la classe qui doit lui être associée dans un autre résultat de segmentation. Ces indicateurs permettent de rendre compte classe par classe des erreurs de classification, mais ne tiennent pas compte des informations spatiales sur les pixels mal classés. Un autre critère permettant de mesurer la dissimilarité entre deux résultats de la segmentation . Il consiste à apparier la région du résultat de segmentation avec une région de la vérité terrain maximisant le recouvrement des régions. Ce critère est facile à mettre en œuvre.

Cependant, il n'apparie pas toutes les classes, donc il ne prend pas en compte toute l'information (entre autre la dispersion spatiale des pixels). Il part de l'hypothèse que deux classes sont à apparier si elles ont un ensemble maximal commun de pixels. Cette hypothèse est restrictive et privilégie les grandes régions. D'autres mesures d'erreur, basées sur le principe de recouvrement entre deux résultats de segmentation en régions ou classes [1] .

3.2. Evaluation non supervisée

Les critères d'évaluation non supervisée permettent d'évaluer la qualité de résultats de segmentation d'une image sans aucune information (voir figure FIG. 1.4). Ils s'adressent donc plus particulièrement aux deux catégories de méthodes de segmentation nécessitant peu de connaissances a priori afin d'éviter l'introduction d'un biais dans l'évaluation. Il faut noter que ces critères mesurent en quelque .

sorte la cohérence du résultat de segmentation en se basant sur des conditions nécessaires que doit respecter toute méthode de segmentation [5] .

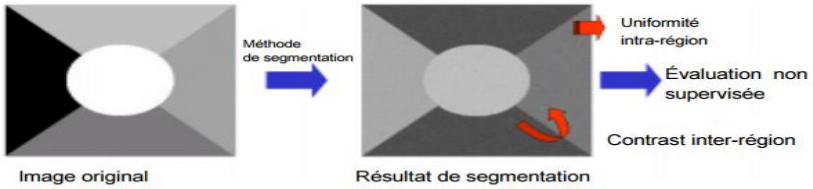


FIG. 1.4 Évaluation non supervisée d'un résultat de segmentation [1]

3.2.1 Évaluation de la détection de frontières

Le critère d'évaluation de cartes de frontières fondée sur la cohérence locale des frontières. Cette cohérence est évaluée sur la base de deux critères de caractérisation des frontières : la continuité et la minceur.

Les frontières sont supposées être générées par un détecteur de frontières de type gradient, suivi par un seuillage. Des critères permettent d'évaluer deux types de frontières : les frontières séparant deux objets de la scène et les lignes qui traversent un objet, sans qu'il y ait de différence de couleur ou de texture entre les régions autour de cette ligne. Le premier type de frontière est évalué par le contraste entre les régions, alors que le second est évalué par le gradient moyen le long de la ligne. Dans le cas d'une image couleur, les calculs sont effectués séparément sur chaque canal et fournissent un vecteur de mesures. Une distance d'ambiguïté pouvant être utilisée dans la détermination de la justesse de détection des frontières. Trois mesures sont utilisées pour calculer l'ambiguïté : l'existence, la localisation et la disposition. L' ambiguïté d'existence et l'ambiguïté de localisation sont dérivées d'une généralisation de la représentation des frontières en utilisant des ensembles flous[1].

3.2.2 Évaluation de la segmentation en régions

L'un des premiers critères pouvant servir à qualifier un résultat de segmentation en régions est l'uniformité intra-région. En effet, sur une image faiblement ou non texturée, les régions segmentées doivent apparaître comme les plus homogènes possibles. Ainsi, un critère d' évaluation avec seuillage qui utilise une mesure de bruit. L'image de départ doit comporter des objets de forme compacte et un fond faiblement texturé. La quantité de bruit pour une image seuillée est calculée en utilisant la matrice de co-occurrence de niveaux de gris de l'image.

Les éléments de la matrice de co-occurrence représentant le pourcentage d'adjacence entre les objets et le fond sont additionnés. Le résultat indique alors une quantité de bruit [1] .

4. Métriques d'analyse de performances d'algorithmes

L'évaluation des performances permet un jugement qualitatif et quantitatif sur un algorithme de traitement d'images. Une propriété d'un algorithme est appréciée par un critère caractéristique, comme par exemple le *PSNR*, le taux de classification correcte ou un jugement subjectif.

La pertinence du critère d'évaluation utilisé est importante car elle influence de façon conséquente celle du jugement réalisé. Dans la mesure où l'évaluation des performances est souvent prépondérante , la littérature disponible dans le domaine du traitement d'images utilise de nombreuses métriques permettant de calculer les performances des algorithmes de segmentation d'images médicales. Ces métriques permettent de quantifier les erreurs entre deux modélisations en terme de distance entre des surfaces ou des volumes.

Pour mesurer les qualités de segmentation, nous avons sélectionné les mesures largement utilisées suivantes : l'erreur de cohérence mondiale (*CME*), évalue la surestimation et la sous-évaluation; l'erreur de cohérence du niveau d'objet (*OCE*),évalue la surestimation où les valeurs plus proches de zéro sont considérées comme meilleures ; l'indice aléatoire (*RI*), calcule une comparaison des segmentations (*GT*)par rapport à la production) en évaluant les points de coïncidence; les valeurs métriques plus proches d'une seule représentent un match parfait; et l'analyse de la Variance de l'Information (*VoI*), où les valeurs inférieures de *VoI* représentent un meilleur résultat [2] .

La relation comparative de l'information valide entre les deux segmentations conduit à une plus grande certitude de l'information valide, pour atteindre une plus grande homogénéité, réduisant ainsi l' entropie.

CME *(1)*

GCE *(2)*

OCE(Se, Sa) *(3)* *(Se) représente la segmentation GT (par un expert humain).*

RI (Sa , Se) *(4)* *(Sa) la segmentation automatique .*

VoI (Sa , Se) *(5)*

4.1. Répartition des données

Cette section présente différentes techniques permettant d'interpréter la répartition de données obtenus par une méthode de segmentation ou de classification .

Elles vont donc permettre de définir s'il y a un comportement récurrent au niveau de la répartition de données.

l'exactitude de la détection dépend de la qualité de la répartition qui est couramment évalué par une matrice de confusion.dans le cas de répartition de pixels entre les classes, , les vrais positifs (*TP*) signifient des pixels correctement détectés ; les faux positifs (*FP*) détectent les pixels n'existe pas vraiment dans la classe; les vrais négatifs (*TN*) sont la détection correcte de la maladie et les faux négatifs (*FN*) sont une détection incorrecte de la maladie. Les valeurs obtenues par *TP, FP, TN* et *FN* sont utilisées pour valider l'exactitude de la détection par des mesures telles que la précision (*p*), le rappel (*r*), les faux taux positifs (*FPR*), l'indice Jaccard (*JI*), l'indice de distance Jaccard (*JDI*) et l'indice Dice (*DI*) également appelé *F-mesure* (*Fm*) ou *F-score* (*Fs*).

4.2 L'exactitude

En particulier, la précision (ou l'exactitude) et le « rappel » (r) sont deux variables statistiques fondamentales qui nous permettent d'établir l'efficacité d'un algorithme de segmentation ou de détection . La précision (p) mesure la proximité entre le résultat de la détection, basé sur un algorithme, et une comparaison standard ou le résultat attendu de la détection.

En termes généraux, c'est le pourcentage de visites. La relation décrivant ce paramètre s'exprime comme suivant

$$p = TP/(TP + FP) \quad (6)$$

4.3 La sensibilité

La sensibilité nous permet d'établir le niveau de performance de l'algorithme, car le nombre d'incidents de détection efficaces sont évalués par rapport au nombre total d'incidents de détection

attendus. La sensibilité est calculée par l'expression suivan

$$r = TP/(TP + FN) \quad (7)$$

4.4 Le taux de faux positifs (FPR)

Est le rapport entre les erreurs et les coups. Lorsque FPR tend à zéro, cela signifie que la détection est meilleure avec un minimum de faux positifs. La relation est présentée comme suit

$$FPR = FP/(TP + TN) \quad (8)$$

4.5 Indice Jaccard

Il s'agit d'une mesure visant à établir le degré de similitude entre l'algorithme proposé et la *GT*. Cet indice décrit la relation entre la détection correcte et les défaillances. Il est représenté ici comme suit

$$JI = TP/(TP + FP + FN) \quad (9)$$

4.6 L'indice de distance Jaccard

Contrairement à *JI*, l'indice de distance Jaccard *(JDI)* est une mesure de la dissemblance. Cette mesure est obtenue en soustrayant *JI* d'un *(1-JI)*, de sorte que le meilleur résultat est obtenu lorsque la valeur *JDI* s'approche de zéro.

$$JDI = 1 - JI \quad (10)$$

4.7. L'indice Dice Coefficient

Ou Dice Index *(DI)* est une autre mesure qui mesure la similitude entre un algorithme de segmentation ou de détection et la *GT*. Le *DI* détermine généralement les performances de l'algorithme en tenant compte à la fois de la précision et de la sensibilité dans l'évaluation. Le *DI* est également connue sous le nom de coefficient de raffinement de précision (mesure *F* ou *F-score*) et son expression est montrée comme suit

$$Fm = 2 * (p.r)/(p + r) \quad (11)$$

Le (*Fm*) établit l'exactitude globale d'un système et, dans ce cas, montre le degré d'exactitude de la segmentation ou de la détection; lorsque la valeur de *Fm* approche 1 (100%), il y'a un plus grand degré de raffinement d'une méthode ou d'un algorithme.

4.8. Scores de classification d'algorithmes

De manière à pouvoir classer différents algorithmes, de nombreux concours (ISBI, MICCAI, Kaggle etc) [6] . sont régulièrement organisés. Ils permettent aux différentes unités de recherche de se positionner par rapport à ce qui se fait en terme d'algorithmie ailleurs dans le monde ainsi que d'avoir accès à des bases de données communes pour se comparer. En ce sens, de nombreuses méthodologies sont mises en place pour proposer un classement le plus objectif possible.Par exemple, lors d'un concours sur la segmentation 3D du foie,des critères fournissant un score global φ sur 100 pour classer les différents algorithmes en compétition [3] .

Ces métriques sont utilisées pour comparer les sorties des algorithmes en compétition par rapport àune référence.

5. Importance de la référence

Pour pouvoir valider les comportements d'algorithmes, ceux-ci sont comparés à une référence considérée comme étant le modèle à obtenir. Bien souvent, la référence est obtenue par un expert clinique qui segmente manuellement le modèle final à obtenir. Cela signifie que la référence de l'expert est considérée comme étant Ground Truth ou vérité terrain[3] (voir FIG.1.5)

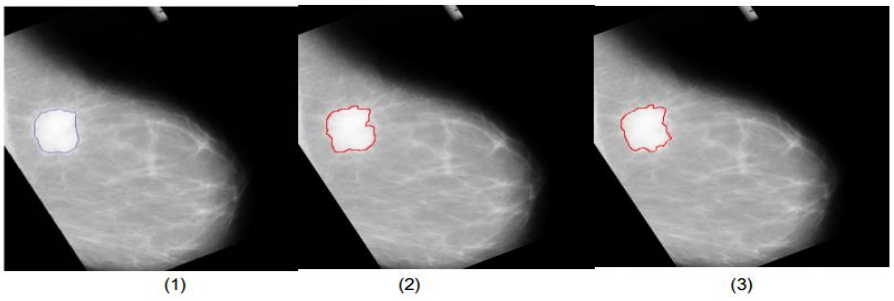


FIG 1.5 (1) Contour détecté par algorithme et (2) ,(3) contours tracés par deux experts [4] .

6. Evaluation des algoritmes de segmentation d'image médical

Les algorithmes de segmentation d'images médicales permettent la détection de structures anatomiques. Chaque année, nombre d'entre eux sont développés dans le but d'améliorer et d'automatiser cette détection. Pour valider ces algorithmes dans le cas de contexte clinique, ils doivent être appliqués sur des images de réelles dont le comportement est comparé à une référence. Il s'agit très souvent de les comparer à une seule référence manuelle générée par un expert clinique [2].

Le fait que chaque algorithme de segmentation est validé en utilisant ses propres critères de validation, généralement en calculant un lot de métriques définies . Il est ainsi difficile de pouvoir comparer différents algorithmes entre eux puisqu'aucune validation objective et commune n'existe. Malgré le fait que certains concours offrent des classements entre différents algorithmes pour une problématique donnée, il est toutefois très difficile d' obtenir une comparaison objective des différents algorithmes développés. Par ailleurs, ces résultats ne permettent pas de proposer des voies d'amélioration.

En effet, les algorithmes sont uniquement classés et les résultats de validation sont toujours présentés sous la forme d'un tableau comprenant de nombreuses valeurs numériques dont l 'interprétation est souvent délicate. À partir de valeurs numériques seulement, il est également difficile d'avoir une vue d'ensemble du comportement global des algorithmes et de pouvoir en extraire leurs points forts et leurs points faibles.

En outre, le processus doit être exécuté sur d'énormes ensembles de données, ce qui entraîne des coûts élevés de traitement informatique et est une tâche fastidieuse pour le chercheur.

7. Conclusion

Dans ce chapitre, nous avons fait un survol sur les méthodes d'évaluation des algorithmes de segmentation d'image ,d'après cette étude, nous avons constaté que :

- Aucun critère d'évaluation de résultat de segmentation ne se révèle satisfaisant pour toutes les images segmentées .

- Pour évaluer correctement des algorithmes de segmentation, il est pertinent d'utiliser le maximum de techniques d' évaluation et combiner leurs résultats. Cependant, cette tâche est fastidieuse pour le chercheur et entraîne des coûts élevés de traitement informatique. Pour l'améliorer. Elle peut être réalisée sur une architecture multithreading, qui sera le sujet du chapitre suivant .

CHAPITRE 2 :

Architectures parallèles multithreading

1. Introduction

Aujourd'hui pour garantir la satisfaction du consommateur dans de nouveaux domaines ,tels que le médical, l'aéronautique, la robotique, le multimédia et la sécurité; les méthodes développées deviennent de plus en plus élaborées et complexes. Ainsi, le traitement séquentiel de certains algorithmes sur des processeurs monocœurs devient de plus en plus obsolète car ne répondant plus aux exigences du temps réel. Les limites concernant l'augmentation de la fréquence du processeur monocœur et du pipeline étant atteintes, nous sommes donc contraints de chercher d'autres solutions pour contourner les difficultés liées aux limites techniques de conception et de vérification tel que l'échauffement. Les surcoûts engendrés nous ont poussés à adopter la solution du parallélisme qui consiste à traiter simultanément plusieurs instructions, données ou tâches ; de nouvelles architectures de traitement parallèles, multiprocesseurs, multicoeurs ou multi-composants sont alors développées , plusieurs niveaux de parallélisme sont sollicités au niveau des instructions et des données .

Dans ce chapitre, nous allons donner un aperçu général sur les architectures parallèles en présentant les différents types de parallélisme, les diverses technologies de traitement parallèle pour les systèmes généralistes , les Intérêts et les difficultés de mise en œuvre.

2. Les différents types de parallélisme

Il existe différents niveaux de parallélisme, chacun possède un mode de fonctionnement adapté à une architecture bien déterminée [9].

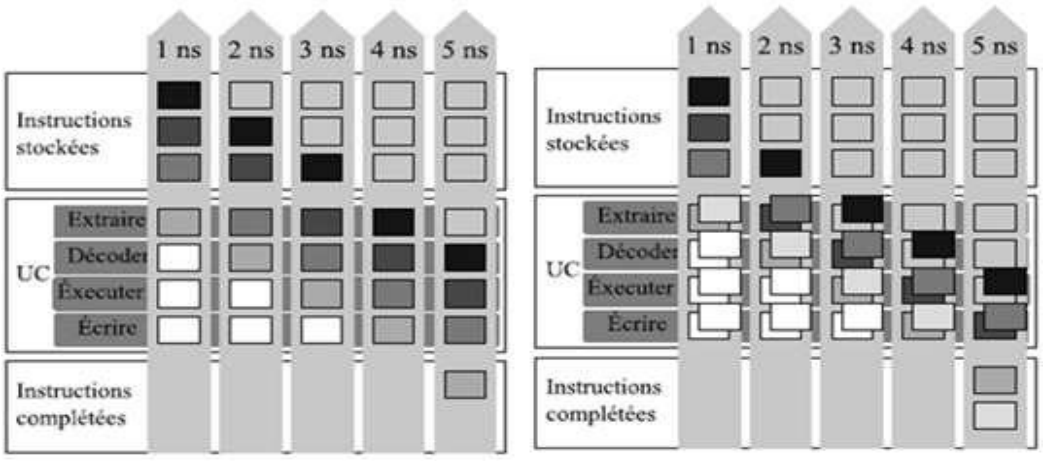
2.1 Parallélisme au niveau des threads

Cette solution consiste à diviser l'application en sous programmes indépendants qu'on nomme «thread», ensuite exécuter ces threads en parallèle sur différentes unités de traitement. Le problème logiciel, provenant de la décomposition de l'application en threads peut être résolu par Le program- meur ou par des compilateurs. Dans les langages de programmation où on note la présence de mécanismes permettant de partager l'algorithme en des threads indépendants qui selon la disponibilité du matériel, peuvent être exécutables en parallèle. Les nouveaux compilateurs évolués peuvent, dans certains cas, réaliser ce travail. Les systèmes multiprocesseurs et multicoeurs s' adaptent avec ce type de décomposition.

En outre, le partitionnement de programmes en threads dépend du nombre de processeurs ou de cœurs afin d'obtenir un parallélisme efficace [9].

2.2. Parallélisme au niveau d'instructions

Dans le cas de figure 2.1, l'idée est la suivante : au lieu de fractionner le programme en sous programmes qu'on pourrait exécuter en parallèle, il y a lieu de paralléliser le programme au niveau de ses instructions. Conséquence, un seul processeur exécute plusieurs instructions simultanément d'un même programme, ceci nécessite des architectures spéciales qui tiendraient compte de ce type de parallélisme. Des techniques ont été développées par les concepteurs permettant au processeur d'exécuter simultanément plusieurs instructions, pas dans l'ordre prévu par le programmeur tel que le pipeline. Effectivement, en pipeline, le traitement d'instructions divise ces dernières en étapes simples d'un nombre fixe qu'il exécute à la façon d'une chaine de montagne, illustré par la Figure 2.1. Le traitement super scalaire en pipeline est une technique améliorée du pipeline et peut être perçue comme étant plusieurs chaines de montagne [9].



.**FIG .2.1** Architecture superscalaire combinée avec un pipeline à quatre étages [9]

2.3 Parallélisme au niveau des données

Le principe consiste à exécuter le même programme sur des données différentes et indépendantes. Ainsi, N données peuvent être exécutées simultanément sur N unités de traitement (processeur, cœur). Le même programme est traité par toutes les unités chacune sur une donnée différente.

Ce parallélisme est caractérisé par une très importante scalabilité, cependant il n'est pas contenu dans toutes les applications [9].

Aujourd'hui, un grand nombre de types de processeurs disposent d'unités dédiées à ce type de traitement appelé *SIMD* (Single Instruction Multiple Data). Elles s'appellent *SSE* (Streaming *SIMD* Extensions) chez *INTEL*, *NEON* chez *ARM* etc. Enfin, les processeurs graphiques *GPU* (Graphics Processing Unit) sont totalement conçus pour exploiter le parallélisme des données

.

3. Les systèmes généralistes parallèles

Après avoir vu les différents types de parallélisme, nous présenterons ici technologies de traitement parallèle pour les systèmes généralistes. La notion de « multi unités » devient une syntaxe universelle et indispensable dans la nouvelle technologie de processeurs ; par conséquent, la notion d'un processeur unique devient très rare. Nous allons présenter trois types de systèmes parallèles : les systèmes multiprocesseurs, les systèmes superscalaires dotés de la technologie hyperthreading et les systèmes multicoeurs. Tous ces systèmes sont adaptés à la technologie multithreading. Ils peuvent être aussi rassemblés ensemble dans un même système.

3.1 Les systèmes multiprocesseurs

Ces systèmes sont apparus en 1960, dotés d'au moins deux processeurs intégrés sur la même carte mère. Ils sont utilisés pour augmenter la scalabilité de traitement et acclimater le système à la complexité du problème traité, c'est-à-dire disposé de plus de puissance de calcul.

Un système multiprocesseur symétrique implique que tous les processeurs du système son identiques. Chaque processeur contient une mémoire cache privée et tous les processeurs sont interconnectés par un bus de données à la mémoire centrale ainsi qu'aux périphériques.

Voir figure ci- dessous (Figure 2.1). Toutes les ressources matérielles et logicielles sont gérées par un seul système d'exploitation. Les applications multitâches (multithreads) sont fortement accélérées grâce au parallélisme du traitement [9].

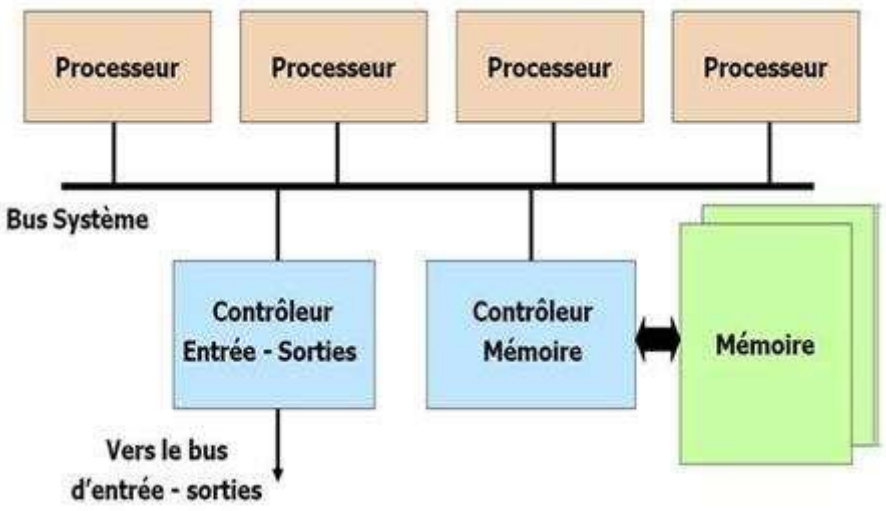


FIG.2.1 Architecture multiprocesseur [9] .

Le noyau du système d'exploitation doit être fondé sur le concept de threads pour pouvoir tirer profit des capacités des systèmes multiprocesseurs. Plusieurs facteurs limitent la scalabilité des systèmes multiprocesseurs tels que les conflits d'accès au niveau matériel (bus), logiciel (système d'exploitation) et aussi le problème de la mémoire centrale, qui reste un facteur limitant l'extensibilité de ces systèmes malgré le recours à la technique de la mémoire cache.

3.2 Processeurs super-scalaires et technologie Hyper-threading

Parmi une suite d'instructions, les processeurs superscalaires sont capables d'exécuter plusieurs instructions simultanées. Ils disposent de plusieurs unités de calcul et sont capables de détecter l'absence de dépendance entre les instructions et les exécuter en désordre. Le Processeur se charge de détecter les instructions qui peuvent être exécutées en parallèle.

On peut éviter ainsi avec cette technologie, la modification des programmes pour exploiter le parallélisme. Cette technologie est conçue pour accélérer le déroulement d'un thread et a été étendue pour permettre d'accélérer plusieurs threads simultanément d'où le nom SMT (Simutaneous Multi- threading). Intel lui a donné le nom de l'Hyper- threading [10].

Elle permet un usage plus efficace des ressources du processeur en exécutant simultanément les threads sur un seul processeur comportant au moins deux unités logiques de traitement.

De nos jours, la majorité des processeurs utilisent la technologie Hyper-threading tels que les processeurs PENTIUM 4, INTEL, I3-I5, I7, les processeurs d'IBM. Cependant cette technologie est limitée par deux facteurs principaux.

- Le degré de parallélisme réel dans le flux d'instructions, à titre d'exemple, un nombre limité de parallélisme au niveau des instructions.
- La complexité de contrôle de dépendances associé à l'application et le coût de l'ordonnancement.

3.3 Les systèmes multicoeurs

La technologie multicoeur est apparue en 2001 avec le premier processeur d'*IBM POWER4*, ensuite *INTEL* et *AMD* ont lancé en 2005 leurs processeurs multicoeurs sur le marché d'ordinateurs personnels. Cette technologie consiste à graver au moins deux unités de calcul (cœur) sur une même puce de silicium afin d'augmenter la puissance de calcul sans influer sur la fréquence de l'horloge, ainsi la dissipation thermique par effet de joule sera réduite. On entend par terme multicoeur symétrique que les cœurs d'un processeur multicoeurs sont tous identiques, par exemple les processeurs multicoeurs d'*INTEL*.

Dans d'autre cas, on peut incorporer plusieurs processeurs assez différents sur la même puce, on peut également opter pour un cœur principal entouré de plusieurs cœurs plus spécialisés qu'on dénomme alors « multicoeurs asymétriques » tel que le processeur *CELL* D'*IBM* et *OMAP* de Texas instrument par exemple.

Les processeurs multicoeurs ont une architecture *MIMD* (Multiple Instruction Multiple Data), ils peuvent exécuter en parallèle plusieurs threads sur différentes données. Chaque cœur dispose d'une mémoire cache privée. Une mémoire partagée permet la communication entre les différents cœurs.

Des difficultés de cohérence de cache et de synchronisation peuvent apparaitre en cas de communication à travers la mémoire partagée, dans ce cas le matériel s'en charge de fournir certaines instructions pour faciliter la communication ou la synchronisation entre les différents cœurs [9].

4. Intérêts d'une architecture à multi-composants

Différents critères peuvent motiver le choix d'une architecture multi-composant [11].

a. Puissance de calcul : Les architectures multiprocesseurs permettent de combiner la puissance de calcul et la spécificité de plusieurs composants pour atteindre les performances nécessaires.

b. Rapidité de développement : Un *ASIC* requiert un temps de développement très long (conception, réalisation, tests) alors qu'une plateforme multi-composant utilise des processeurs existants. On trouve également ce genre de plateformes chez certains fabricants comme *Vitec Multimédia* ou *Sundance*. La programmation des composants peut être réalisée dans un langage de haut niveau.

c. Prototypage : Le prototypage d'applications est une étape utile dans un processus de développement. Une architecture proche du produit final permet de prouver la faisabilité et de mettre en évidence la plupart des goulots d'étranglements sur lesquels les efforts de développement doivent être concentrés.

d. Faible coût : Pour les marchés, où le développement d'un ASIC serait trop coûteux face à un faible volume de vente, les solutions multiprocesseurs peuvent s'avérer profitables.

e. Evolutivité : Les cartes multi-composant sont généralement construites autour d'un ou plusieurs processeurs standards (GPP ou DSP) programmables en C. Les applications sont donc rapidement portées sur cible, de plus, l'application comme la plateforme peuvent être modifiées en limitant les conséquences (redéveloppement).

f. Flexibilité : Une carte multiprocesseur est réutilisable pour plusieurs applications beaucoup plus facilement qu'un composant dédié. Malgré l'augmentation des capacités d' intégration des composants et de l'émergence des composants multicoeurs, les plateformes multiprocesseurs restent un compromis pour la réalisation de prototypes et les marchés de petites séries.

5. Difficultés de mise en œuvre

Le portage d'une application sur une architecture multi-composant n'est pas trivial, les principales difficultés résident dans la parallélisation des calculs et dans les dépendances de données [11].

5.1 Parallélisme et ordonnancement

La parallélisation de l'algorithme constitue la première difficulté. Il faut dispatcher l'application sur les différentes unités de calcul pour pouvoir profiter d'une architecture multiprocesseur. Il faut également bien étudier les dépendances des données entre les différentes opérations pour pouvoir paralléliser des opérations indépendantes. L'ordonnancement a son importance dans le fait de permettre de réduire les attentes de données conséquentes à une sous-utilisation des processeurs et ce pour une meilleure efficacité de l'architecture parallèle. Cependant quand des opérations critiques sont déportées sur un composant dédié sans le ré-ordonnancement des traitements ou l'adaptation de l' algorithme, il est à craindre une utilisation sous-optimale. Effectivement, le processus comme le calcul, transmet les données sur le coprocesseur et attend le résultat, ceci se traduit par une période d' inactivité et par conséquent une accélération réduite. Ainsi, un système constitué d'autant d'unités de calcul dédiées que d'opérations critiques est difficile à exploiter de manière efficace dans la mesure où il conduirait à une sous-utilisation de ces composants.

5.2 Les systèmes de communications

Une application distribuée implique forcément des communications et synchronisations inter-composants. La distribution et l'ordonnancement des opérations doivent prendre en compte les aspects de bande passante sur les liens de communication entre composants. En effet, la saturation des communicateurs conduit à l'inactivité des composants, et donc à une utilisation sous-optimale de l' architecture. Il est donc important de bien dimensionner les bandes passantes inter-processeurs ou de réduire les échanges de données en distribuant les opérations

de manière plus efficace.Les systèmes multi composants sont couramment employés pour accélérer le temps de calcul.

Cependant, pour assurer une meilleure rentabilité, les temps de développement qui restent longs doivent être réduits. En outre, l'hétérogénéité de l'architecture doit être transparente pour le concepteur . Les bibliothèques spécialisées et IP ainsi que les langages de haut niveau participent à cet effort.

6. Conclusion

Dans ce chapitre, nous avons passé en revue Les différents types de parallélisme, les différentes architectures parallèles pour les systèmes généralistes . Nous avons examiné les inconvénients et les avantages de ces architectures adaptées à la technologie multithreading .

Finalement, une architecture multicoeurs (multithreading) permet l'amélioration d'une programmation multitâche si elle dispose d'un système d'exploitation et d'une architecture homogène. Manquant de généricité , cette approche est généralement utilisée après les phases de conception. Ainsi, afin d' accélérer les développements sur des architectures parallèles, pour des processeurs avec de plus en plus de cœurs de calcul ou pour les plateformes multi-composants, des dispositifs de conception d' application parallélisées sont nécessaires.

CHAPITRE 3 :

Conception et approche multithreading

1. Introduction

L'utilisation de la segmentation, pour la détection et l'extraction des objets intéressés à partirdes images médicales, joue un rôle très important dans le traitement d'image numérique.

Le développement d'une méthode de segmentation automatique des images médicales, est nécessaire afin d'apporter à l'utilisateur une aide pour l'interprétation des images obtenues. Cette segmentation peut être réalisée à travers différents sous-objectifs [14]:

- Extraction et sélection de caractéristiques.
- Conception et validation d'un classifieur.
- Mise en place d'une étape de régularisation pour la segmentation automatique des
- régions d'intérêt.
- Validation de la méthode de segmentation.

En partant de ces point, plusieurs méthodes de segmentation ont étés depuis plusieurs décennies [15]. Cependant définir une bonne segmentation reste difficile d'autant plus que la solution dépend du but recherché. Le choix de l'algorithme de segmentation le plus approprié face à une application donnée est une question d'actualité. En effet, estimer la performance des algorithmes de segmentation est une tâche importante qui ne cesse de susciter l'intérêt des chercheurs [13]. De ce fait, l'attention a été attirée sur la nécessité d'avoir recours à des études de comparaison des résultats issus des différents algorithmes de segmentation quand des images semblables sont segmentées [16].

Différentes mesures sont utilisées pour comparer le résultat de segmentation avec une vérité terrain (*GT* : Ground Truth), qui est une segmentation considérée comme «parfaite» et qui a normalement été obtenue manuellement par des experts humains, qui ont classé chaque pixel dans les régions atteintes et non atteintes. À la fin, une image binaire est générée et cette image représente un modèle de comparaison.

Le processus d'évaluation par comparaison avec une *GT* doit être effectué individuellement. Cela signifie que, pour chaque algorithme à tester, plusieurs évaluations sont effectuées, une pour chaque mesure. Une comparaison des différentes méthodes de segmentation, par rapport au GT, est utilisée pour évaluer la qualité de segmentation dans l'extraction des régions atteintes. En outre, le processus doit être exécuté sur d'énormes ensembles de données, ce qui entraîne des coûts élevés de traitement informatique et est une tâche fastidieuse pour le chercheur.

Pour améliorer l'efficacité de la phase de test, une architecture multithreading peut être utilisée en évitant le traitement séquentiel répété ; les ressources informatiques et les délais de traitement seront réduits.

Dans notre travail, nous nous intéressons au problème d'évaluation et de comparaison des performances d'un ensemble d'algorithmes de segmentation, dans le cadre de détection automatique des zones d'intérêt dans des images médicales, dans une structure de traitement parallèle, en exploitant le paradigme multithreading de programmation parallèle pour réduire le temps de calcul.

2. Méthodes de segmentation

Parmi les approches très répandues de la segmentation d'image, on trouve la segmentation par seuillage qui consiste à classer, suivant le nombre de classes, les différents pixels d'une image, en se basant sur l'histogramme de niveaux de gris de l'image ou sur la courbe d'énergie.

2.1. Seuillage à base d'histogramme

Les méthodes de seuillage globale reposent sur l'exploitation de l'histogramme de toute l'image. L'histogramme est une courbe monodimensionnelle qui caractérise la distribution des niveaux de gris, il est décrit par une fonction monovariable discrète $h(i)$ ou $p(i)$ qui représente respectivement la fréquence ou la probabilité d'apparition du niveau de gris i, tel que:

$$p(i) = \frac{h(i)}{N} \quad (1)$$

$h(i)$ Étant le nombre de pixels ayant le niveau de gris i et N le nombre total de pixels dans l'image.

2.1. Seuillage à base de la courbe d'énergie

Une courbe d'énergie [21] est semblable à l'histogramme de l'image mais elle permet de prendre en compte les informations contextuelles spatiales de l'image pour sélectionner le seuil optimal de l'mage.

La courbe d'énergie(Energy curve) consiste à calculer l'énergie associée à chaque valeur de niveau de gris de l'image afin de générer la courbe d'énergie en prenant en compte les informations contextuelles spatiale de l'image selon le principe suivant :

Soit $I = \{l_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\}$ une image de taille m × n où l_{ij} est la valeur de gris de l'image I à la position de pixel *(i, j)*. Soit L la valeur de gris maximale de l'image I. La corrélation spatiale entre les pixels voisins de l'image I est modélisée en définissant le système de voisinage N d'ordre d, pour la position donnée spatiale (i, j) que $N_{ij}^{d} = \{(i + u, j + v), (u, v) \in N^{d}\}$. Selon la valeur de d, le système de voisinage assumer différentes configurations. Dans [S. Patra et al. , 2014] seuls les systèmes de voisinage de second ordre est considéré, à savoir, $(u, v) \in \{(\pm 1, 0), (0, \pm 1), (1, \pm 1), (-1, \pm 1)\}$. La Figure 3 représente de second ordre N^2 de pixels voisins du pixel à la position spatiale (i, j).

Pour calculer l'énergie de l'image I à la valeur l gris ($0 \leq l \leq$ L), premièrement, il faut générer une matrice binaire à deux dimensions $Bl = \{ b_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\}$ tel que $b_{ij} = 1$ si $l_{ij} > l$; sinon $b_{ij} = -1$. Ainsi, la valeur de chaque élément dans b_{ij} dans B_l est 1 ou -1 en fonction de la valeur l_{ij} du niveau de gris du pixel correspondant et la valeur de 1.

Soit $C = \{c_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\}$ une autre matrice de telle sorte que $c_{ij} = 1, \forall (i, j)$ .Ensuite la valeur énergétique E_l de l'image I au valeur niveau de gris l est défini comme :

$$E_l = -\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{pq \in N_{ij}^2} b_{ij} * b_{pq} + \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{pq \in N_{ij}^2} C_{ij}\, C_{pq} \qquad (2)$$

Le deuxième terme de l'expression, écrite sur le côté droit de l'équation, est un terme constant et assurela valeur énergétique $E_l \geq 0$.

De l'équation (2), on peut voir que pour une image I, la valeur de l'énergie au niveau gris l sera zéro lorsque tous les éléments de la matrice générée B_l sont soit 1 ou -1 c'est-à-dire, tous les pixels de l'image I ont des valeurs de niveaux de gris soit supérieure à l ou moins de l. Dans le cas **contraire,** l'énergie sera positive.

(i-1, j-1)	*(i-1, j)*	*(i-1, j+1)*
(i, j-1)	*(i, j)*	*(i, j+1)*
(i+1, j-1)	*(i+1, j)*	*(i+1, j+1)*

FIG 3.3 : Matrice de voisinage de 2eme ordre de pixel (i,j)

La courbe d'énergie obtenue en (2) a des caractéristiques intéressantes. Soient les valeurs de niveaux de gris des pixels dans l'intervalle $[t_1, t_2]$ représentent un objet dans l'image. Pour $l = t_1$, les éléments dans la matrice Bl correspondant aux pixels dans l'objet seront 1. Si nous augmentons la valeur de l, quelques éléments dans la matrice Bl seront passés de 1 à -1, comme un résultat l'énergie va augmenter. L'énergie augmente jusqu'à certaine valeur de l, puis diminue à mesure que les entrées voisines dans Bl changent aussi de 1 à -1. Pour $l = t_2$, toutes les entrées Bl correspondent aux pixels de l'objet sera -1. Ainsi, une courbe d'énergie de forme de cloche sera générée dans l'intervalle $[t_1, t_2]$.

a) Méthodes d'OTSU

Elle est considérée comme la méthode de référence dans le domaine du seuillage d'histogrammes. Dans cette méthode [18], l'opération de seuillage est vue comme une séparation (un partitionnement) des pixels d'une image en deux classes C_1 (fond), C_2 (objet) à partir d'un seuil t .

La classe « fond » regroupe tous les pixels ayant un niveau de gris inférieur au seuil t alors que la classe « objet » contient tous les pixels de niveau de gris supérieur à t. Ces deux classes peuvent être désignées en fonction du seuil t comme suit :

C_1 {0,1,…. t}, C_2{t+1,..., L-1}

Soient σ_w^2 la variance d'une classe, σ_B^2 la variance interclasse et σ_T^2 la variance totale telles que :

$$\sigma_B^2 = P_1\, P_2\, (\mu_2 - \mu_1),\ \sigma_T^2 = \sum_{i=1}^{L-i} P_i\, (i - \mu_1)^2 \qquad (1)$$

$$\sigma_w^2 = \sum_{i=0}^{t} p_i\, (i - \mu_1)^2 + \sum_{i=t+1}^{L-i} P_2\, (i - \mu_2)^2 \qquad (2)$$

$$\sigma_T^2 = \sigma_B^2 + \sigma_w^2 \qquad (3)$$

μ_1 , μ_2 et $\boldsymbol{\mu}$ désignent respectivement les niveaux de gris moyen des classes C_1, C_2et de l'image tels que :

P_1 et P_2 représentent respectivement les probabilités à priori des classes C_1 et C_2 tels que :

$$P_1 = \sum_{i=1}^{t} p_i\ ,\ P_2 = \sum_{t+1}^{L-1} p_i \text{ et } P_1 + P_2 = 1 \qquad (3)$$

Le seuil optimum t* peut être déterminé en maximisant un des trois critères suivant :

$$\lambda(t) = \frac{\sigma_B^2}{\sigma^2},\quad \eta = \frac{\sigma_B^2}{\sigma^2},\ \kappa = \frac{\sigma_T^2}{\sigma^2} \qquad (4)$$

Les trois critères sont équivalents, mais pour des raisons de simplicité, la variance interclasse est la plus utilisée. Le seuil optimal *t** est celui qui maximise cette variance [Otsu, 1979] :

$$t^* = Arg \max \sigma_B^2 \qquad (5)$$

Selon la base du calcul de variances qui se soit l'histogramme de l'image ou bien l'énergie de l'image ; on peut distinguer méthode **Otsu et méthode Otsu Energie** respectivement.

b) Méthodes de kittler et Illingworth

Proposée par Kittler et Illingworth **[20]**, cette méthode considère l'histogramme des niveaux de gris comme une estimée de la fonction de densité de probabilité $p(i)$ d'un mélange de populations formées des niveaux de gris des objets et du fond.

On peut supposer que chacune des deux composantes du mélange forment une distribution normale avec une moyenneμ_i, un écart type $\sigma_i$et une probabilité à priori p_i. On a ainsi :

$$P(\frac{i}{C_k}) = \frac{1}{\sqrt{2\Pi\sigma_k^2}} \exp(-\frac{(i-\mu_k)^2}{2\sigma_k^2})$$

$$p(i) = \sum_{k=0}^{i} P_k \, P(\frac{i}{C_k})$$

Le seuil optimal est obtenu en résolvant l'équation quadratique ci-dessous :

$$\frac{(i-\mu_0)^2}{\sigma_0^2} + \log\sigma_0^2 - 2\log P_0$$

$$P_0 = \frac{(i-\mu_0)^2}{\sigma^2} + \log\sigma_0^2 - 2\log P_1$$

Ou en minimisant soit la fonction $J_1(t)$ suivante :

$$J_1(t) = 1 + 2\,[P_0(t)\log\sigma_0(t) + P_1(t)\log\sigma_1(t)] - 2\,[\log P_0(t) + P_1(t)\log P_1(t)] \qquad (6)$$

Avec :

$$\sigma_0^2 = \frac{\sum_{n=0}^{t}(i-\mu_0(t)^2)}{P_0\,(t)}\ ,\ \sigma_0^2\,(t) = \frac{\sum_{t+1}^{l-1}(i-\mu_0(t)^2)h(i)}{P_1\,(t)}$$

$$P_0\,(t) = \sum_{i=0}^{t} h(i) \text{ et } P_1\,(t) = \sum_{i=t+1}^{l-1} h(i)$$

$$\mu_0(t) = \frac{\sum_{i=0}^{t} i\,h(i)}{P_0\,(t)} \text{ et } \mu_1(t) = \frac{\sum_{i=t+1}^{l-1} i\,h(i)}{P_1\,(t)}$$

Soit la fonction de critère $J_2\,(t)$:

$$J_2\,(t) = N^2(\omega_0\ log\frac{\sigma_0}{\omega_0 N^2} + \ \omega_1\ log\frac{\sigma_1}{\omega_1 N^2}) \qquad (7)$$

Avec :

$$\sigma_0^2 = \frac{1}{\omega_0 N^2}\sum_{i=0}^{t}(i-\mu_0)^2 h(i)\ ,\ \sigma_1^2 = \frac{1}{\omega_1 N^2}\sum_{i=t}^{l-1}(i-\mu_1)^2 h(i)$$

Les paramètresω_0 , ω_1 , μ_0, μ_1 sont calculés comme suit :

$\omega_0 = \sum_{0\le i\le t} P_i$, $\mu_0 = \sum_{0\le i\le t} i\frac{P_i}{\omega_0}$ et $\omega_1 = \sum_{t\le i\le l-1} P_i$, $\mu_1 = \sum_{t\le i\le l-1} i\frac{P_i}{\omega_0}$

Le calcul de $J_1\,(t)$ et $J_2\,(t)$ se base sur l'histogramme de l'image ou bien l'énergie de l'image d'où on distingue deux méthodse de kittler et Illingworth **: Kitler et kitler énergie** respectivement.

c) Méthodes de Kapur

La méthode de Kapur [19] est basée sur le principe de maximisation de l'entropie de Shannon. Elle suppose que les classes « objet » et « fond » possèdent deux densités de probabilité indépendantes.

$$p_i = \frac{h(i)}{N}\,,\ P_1 = \sum_{i=0}^{t} P_i\,,\ P_2 = \sum_{i=t+1}^{l-1} p_i \ avec\ P_1 + P_2 = 1$$

Les entropies de Shannon associées aux deux distributions sont :

$$H_1(t) = -\sum_{i=1}^{t} \frac{p_i}{P_1} \log\left(\frac{p_i}{P_1}\right) \text{ et } H_2(t) = -\sum_{i=t+1}^{L-1} \frac{p_i}{P_2} \log\left(\frac{p_i}{P_2}\right) \qquad (8)$$

Le seuil optimal t^* est alors défini comme étant le niveau de gris qui maximise les deux entropies, c'est-à-dire que :

$$t^* = \arg\max\{H_1(t) + H_2(t)\}$$

Cette méthode peut être également étendue au calcul de plusieurs seuils $T = \{t_1, t_2, \ldots, t_{k-1}\}$, Il s'agira alors de maximiser le critère suivant :

$$J(T) = \sum_{k=1}^{K} H_k$$

Avec :

$$H_k = \sum_{i=t_k-1}^{t_k-1} \frac{p_i}{P_k} \log\left(\frac{p_i}{P_k}\right) et\ P_k = \frac{1}{N} \sum_{i=t_k-1}^{t_k-1} h(i)$$

Le calcul de entropies se base sur l'histogramme de l'image ou bien l'énergie de l'image d'où on distingue deux méthodes de **Kapur et Kapur énergie** respectivement.

2. Métrique d'évaluation

La relation comparative de l'information valide entre les deux segmentations conduit à une plus grande certitude de l'information valide, pour atteindre une plus grande homogénéité, réduisant ainsi l'entropie. Plusieurs métriques d'évaluation peuvent être utilisés :

a. L'exactitude (p).

b. La sensibilité (r).

c. Le taux de faux positifs (FPR) .

d. Indice Jaccard (JI).

e. L'indice de distance Jaccard(JDI).

f. L'indice Dice Coefficient (Fm)

Ces métriques sont calculés à partir des paramètres suivants qui sont déterminés à partir de matrice deconfusion :

TP les vrais positifs signifient des pixels correctement détectés .

FP les faux positifs détectent les pixels n'existe pas vraiment dans la zone.

TN les vrais négatifs sont la détection correcte de la maladie .

FN les faux négatifs sont une détection incorrecte de la maladie.

3. Approche multithreading pour l'évaluation de performance des méthodes de segmentation

Dans notre travail, nous nous adapté un processus de parallélisation générique qui peut être appliqué à tout système composé d'étapes de traitement individuelles, où certains d'entre eux pourraient être échangés par différents algorithmes et testés à des fins de comparaison.

Tous ces algorithmes seraient mis en œuvre en parallèle, et les résultats pourraient être comparés instantanément, évitant l'exécution répétitive de l'ensemble du système.

Dans notre cas, le système est composé des étapes suivantes, où les étapes mises en évidence sont celleà être parallélisées

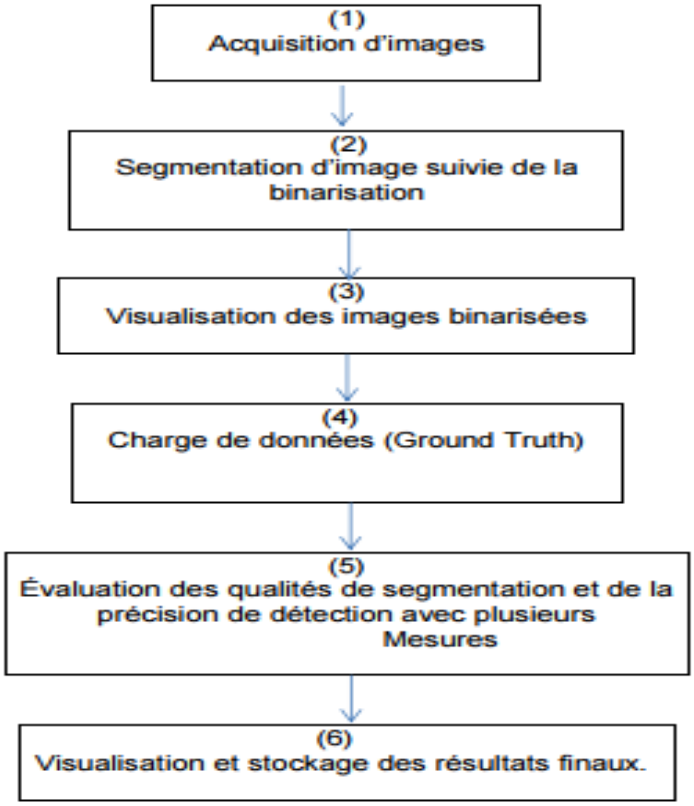


FIG 3.1 Processus d'évaluation d'un algorithme de segmentation .

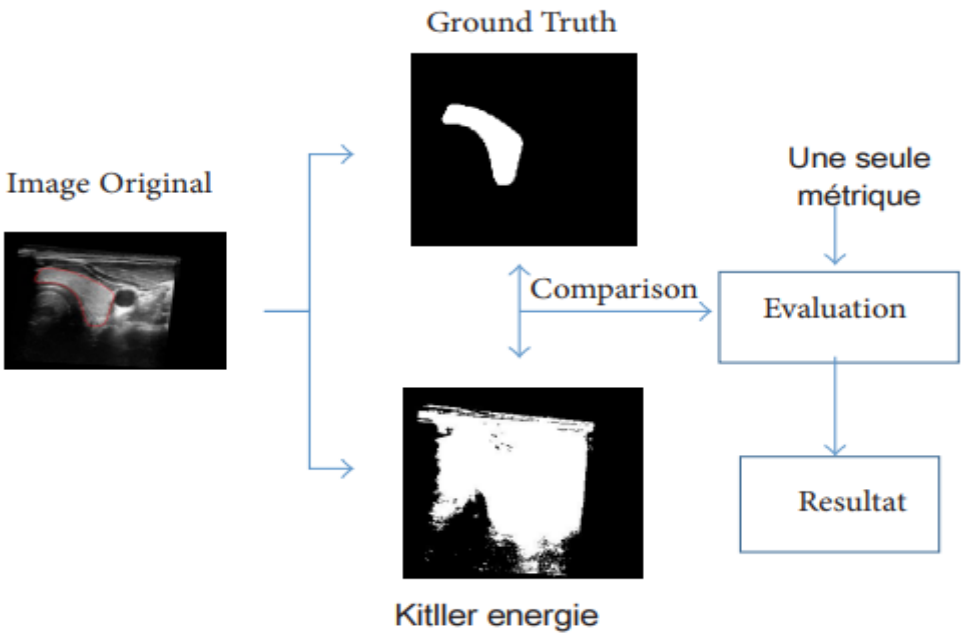


FIG 3.2 Exemple d'évaluation de segmentation d'une image échographique par la methode Kitler energie

Pour éviter l'exécution de l'ensemble du programme cinq fois (pour tester cinq algorithmes de segmentation, chaque algorithme a énorme de données), les algorithmes décrits ont été mis en oeuvre en parallèle à l'étape (2).

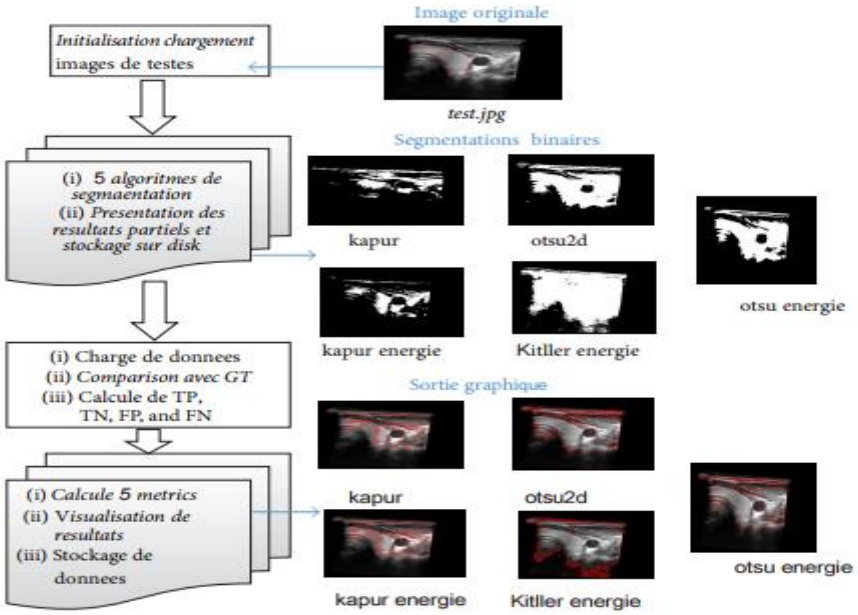


FIG. 3.2 Flux de travail du processus parallélisé,
(où les pièces ombragées sont les tâches
fonctionnant sur plusieurs threads).

De cette façon, chaque image de test devait être chargée une seule fois, et les cinq résultats pouvaient être traités en parallèle, comparant visuellement les résultats et économisant les mesures de qualité dans des fichiers de données. En conséquence logique, la 5ème étape du calcul métrique a également dû être parallèle pour obtenir toutes les mesures pour les cinq résultats simultanément. Ces procédures ont été appliquées dans certaines parties du processus dont nous souhaitons échanger les algorithmes, comme expliqué ci-dessus. Le processus contient deux parties principales qui sont intensives sur le plan du calcul : les segmentations d'images suivies de la binarisation d'une part et l'évaluation des qualités de segmentation et de la précision de détection par de multiples mesures d'autre part.

Ce n'est que dans ces régions que les tâches ont été parallélisées; toutes les autres étapes sont encore effectuées de façon séquentielle. La charge de données (étape (4)) pourrait être omise, mais comme les résultats sont stockés dans la mémoire, il a été inclus comme une possibilité de séparer l'ensemble du processus en deux parties, qui pourraient être effectuées individuellement si nécessaire et d'ajouter de la flexibilité lors de la substitution des algorithmes individuels.

4. Architecture du système

L'architecture du système considère deux types de parallélisme : un parallélisme de contrôle et un autre des données. Le parallélisme de contrôle est exécuté en particulier quand effectuer des processus de segmentation et d'évaluation de chacune des méthodes, car ils sont exécutés simultanément et avec une charge de travail équilibrée. D'autre part, le parallélisme des données est présenté en partageant des informations matrices de l'image originale, la matrice de l'image GT, et les paramètres de la matrice de confusion.

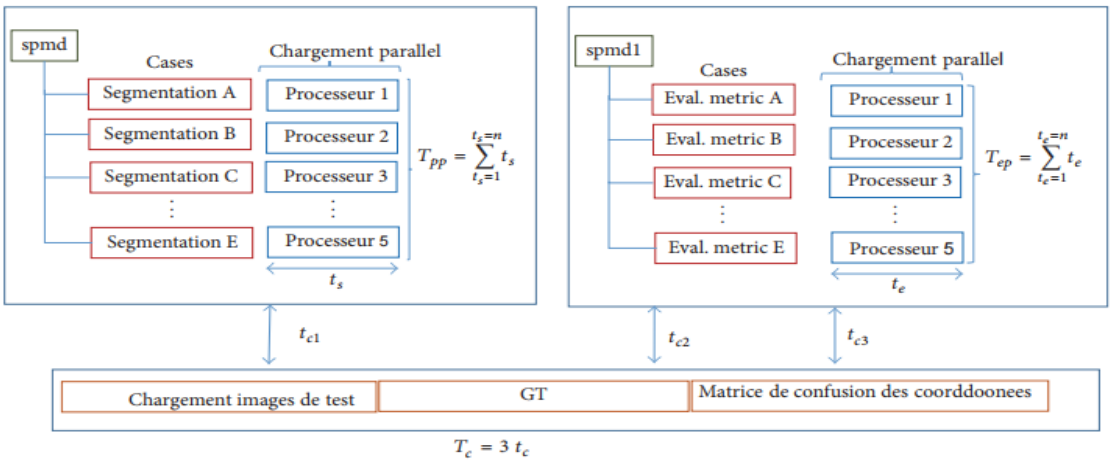


FIG 2.3 Architecture parallèle utilisée dans l'évaluation des différentes méthodes de segmentation

3.1 Parallélisme de contrôle

Nous permet d'exécuter toutes les segmentations en même temps et indépendamment de chacun des processeurs disponibles. Dans cette approche, une première phase de décomposition du programme en tâches ou « cas » (T) est effectuée et une exécution simultanée des segmentations est définie.

Ensuite, la base algorithme de la segmentations, sous forme parallèle dans ce cas, sera composée de cinq tâches: T1 ,T2 ,T3 ,T4 ,T5 , de sorte que tous les cas sont exécutés simultanément , avec chaque tâche d'une durée prédéterminée t1 ,t2 ,t3 ,t4 ,t5 , que chaque méthode de segmentation a un temps d'exécution différent, comme le fait chaque mesure à évaluer. L'architecture a été conçue de telle sorte que lorsqu'un processeur est libre, il aide immédiatement dans le traitement des autres, de sorte que la charge de calcul est distribuée d'une manière équilibrée. Cela garantit que le temps de traitement nécessaire pour exécuter les segmentations parallèles (Tpp) sera la somme des temps partiels de sorte que Tpp= 5ts estime le temps total par parallélisme de la même manière qu'il faut toutes les mesures pour exécuter(Tep).

3.2 Parallélisme des données

Un autre avantage de l'utilisation d'une architecture parallèle est la réduction des temps de communication (tc) en partageant des informations d'usage commun et en évitant les calculs redondants. Puisque t est égal dans ce cas, le temps de communication total (Tc) sera 3tc . Le partage d'informations par le biais du parallélisme des données assure l'efficacité du calcul, en particulier compte tenu de la structure de matrice des informations. L'exploitation du parallélisme des données vient de la prise de conscience que certaines applications agissent sur des structures de données régulières (matrices d'images, données de la matrice de confusion), répétant le même calcul sur chaque élément de la structure; de cette façon, l'algorithme manipule des structures de données régulières. L'idée est d'exploiter la régularité des données en effectuant le même calcul sur différentes données en parallèle; par exemple, si une segmentation est effectuée selon la méthode A, cette méthode utilisera les mêmes informations de la matrice d'image originale que la méthode B ou C, et ainsi de suite. En outre, les données obtenues à partir des coordonnées de la matrice de confusion seront utilisées pour appliquer les mesures A, B ou C, et ainsi de suite, de sorte que dans ce type de parallélisme, il associe les données directement avec les processeurs. Comme les calculs sont effectués en parallèle sur des processeurs identiques, il est possible de centraliser le contrôle. Comme les données sont similaires, l'opération de répétition prend le même temps de communication sur tous les processeurs, et le contrôleur envoie l'opération à tous les processeurs de façon synchrone.

Ce type de parallélisme peut être expliqué en ce qui a vu le cas si des processeurs p sont disponibles et si une série de T sont exécutées. deux opérations de base sont présentées dans le parallélisme des données, elles peuvent être exprimées telles que la Φ-notation et la ρ-reduction.

La Φ-notation est décrite par une fonction f de dimension m et m vecteurs de la même taille. Lanotation applique la fonction f sur l'ensemble des vecteurs en parallèle. L'opération de réduction est une fonction binaire f sur un seul vecteur r, telle que la fonction f est appliquée aux deux premiers éléments de V, puis f est appliquée au résultat du calcul précédent et l'élément suivant du vecteur V . Les deux Φ et ρ sont suffisamment généraux pour exprimer un grand nombre d'opérations sur des données régulières, telles que les vecteurs et les matrices, lorsqu'ils tentent d'exploiter le parallélisme des données.

Dans la Φ-notation, par exemple, l'opération (Φ(+V1, V2)) la somme des vecteurs V1 et V2 pour produire le résultat vector r; cela permet l'apparition du parallélisme des données. Dans la ρ-reduction, par exemple, (+, V) calcule la somme de tous les éléments du vecteur V. Avec certains changements dans cette opération, pour exploiter la propriété associative (de la division des tâches), il est possible de générer un schéma d'exécution parallèle.

3.4 Conclusion

L'application multithreading a été développée à l'aide de
La ParallelComputingToolboxofMATLABR2015b, qui fournit de nouvelles procédures pour effectuer plusieurs processus simultanément (tels que le multitâche) sur les plates-formes multicores.

CHAPITRE 4

Implémentation et résultats

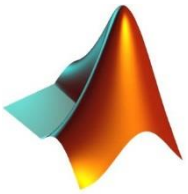
1. Introduction

Dans ce chapitre, nous allons présenter l'environnement de travail, le langage de programmation et les outils que nous avons utilisé pour implémenter notre système d'évaluation parallèle des algorithmes de segmentation d'images. Nous détaillons les principales fonctions de boîte à outils de calcul parallèle MATLAB (PCT : Parallel Computing Toolbox) exploitées pour de développer notre application sur un processeur multicœur ou un cluster d'ordinateurs.

2. Environnement et outils de développement

Notre application a été réalisé sur un en utilisant langage Matlab et l'environnement Matlab avec les fonctions offertes par la boite à outils de calcul parallèle (PCT).

2.1. Environnement de développement



Matlab est un logiciel de calcul numérique commercialisé par la société MathWorks1. Il a été initialement développé à la fin des années 70 par Cleve Moler. Matlab est abré- viation de MATrix LABoratory. Il est avant tout, un programme de calcul matriciel pour le calcul scientifique, l'analyse de données, leur visualisation et le développement d'algorithmes. Son interface propose, un environnement de développement intégré (IDE) pour la programmation d'applications. Le logiciel peut être complété par de multiples toolboxes, c'est-à-dire des boîtes à outils. Celles-ci sont des bibliothèques de fonctions [17].

2.2. Boite à outils de calcul parallèle

La boîte à outils de calcul parallèle MATLAB permet aux utilisateurs de résoudre des problèmes de calcul et de données gourmandes en utilisant des ordinateurs multicœurs et multiprocesseurs, des clusters d'ordinateurs et des GPU. Les utilisateurs peuvent utiliser des fonctions MATLAB de haut niveau pour paralléliser des applications sans programmation OpenMP, MPI et CUDA. L'une des caractéristiques importantes de Parallel Computing Toolbox de MATLAB est que la même application peut être exécutée sur un cœur simple, un processeur multicœur ou un cluster d'ordinateurs sans changer le code. En combinant Parallel Computing Toolbox avec Distributed Computing Server, les utilisateurs peuvent exécuter leurs programmes MATLAB sur des clusters d'ordinateurs, des grilles et des nuages.

3.Initialisation du système Multitâche

3.1 Fonction parpool

Dans le premier sous-processus, le système prépare l'attribution des tâches. Par conséquent, il définit la séquence des tâches pour chaque algorithme et les assigne aux processeurs (ou cœurs) en utilisant la fonction de parpool fournie par Parallel Toolbox. Pour obtenir des performances optimales, le nombre de threads doit être égal au nombre de cœurs (CPUs). S'il y avait plus de threads que de processeurs disponibles, certains cœurs devraient exécuter plusieurs tâches. Le résultat est une augmentation du temps de traitement et un manque d'uniformité dans la capacité de traitement. Dans ce cas, l'évaluation du temps de fonctionnement ne serait pas valide, car la capacité de traitement d'un processeur devait être divisée entre des threads.

3.2 Worker

La Parallel Toolbox de MATLAB applique worker threads et utilise le profil local, qui est un paramètre par défaut de la toolbox. Après l'initialisation, le work folder et ses subfolders sont ajoutés à l'itinéraire de recherche de MATLAB. Par conséquent, il n'est pas nécessaire de faire des spécifications sur l'endroit où se trouvent les fonctions et les fichiers, à condition qu'ils soient situés dans l'un de ces subfolders.

Dans notre cas, cinq algorithmes de segmentation seront testés. Par conséquent, cinq worker threads

sont définis pour les calculs et pour présenter les informations résultantes sur l'écran. De cette façon,

chaque worker exécute les tâches déclarées dans un script qui contient toutes les méthodes de segmentation à évaluer en parallèle.

4.Segmentation et évaluation des résultats

Pour realiser plusieurs segmentations en parallèle sur une même image, un système multitâche attribue une méthode de segmentation à chaque thread. Comme notre travail s'intéresse à l'évaluation de performance de cinq méthodes de segmentation par seuillage, les méthodes de segmentation sont assignées aux structures des cas (de 1 à 5).

4.1 Fonction *smpd* et Fonction switch case

La fonction *smpd* obtient les identifiants et assigne un «worker » à chaque processus de segmentation à l'aide d'instruction switch case . Chaque cas contient une liste des tâches à effectuer par le worker assigné (dans nos cas la segmentation, le calcul des paramètres d'évaluation, la représentation de l'état d'exécution à l'écran, etc.).

4.2 Type composite

Les variables utilisées par chaque thread font partie d'une autre valeur de type composite, qui est un type spécial de tableau. Il a le même nombre d'éléments que les processus de worker existent et est utile pour récupérer les données d'anciennes segmentations.

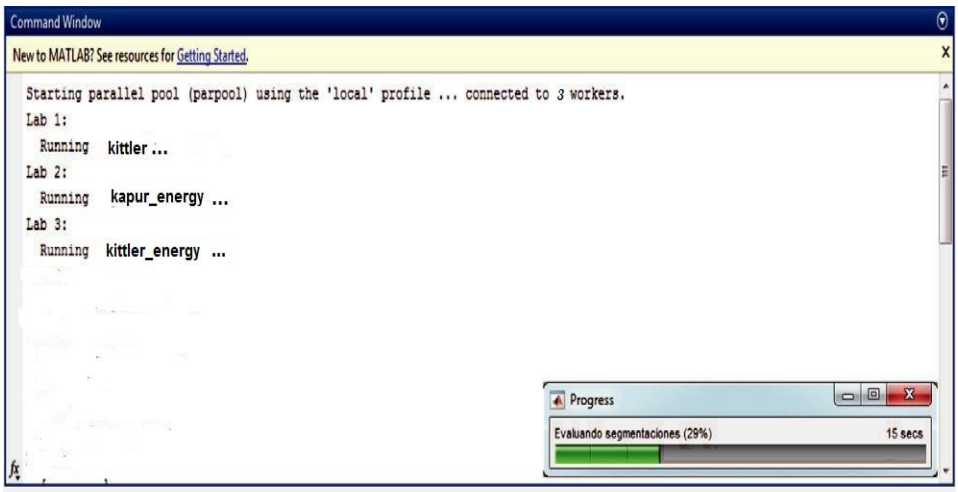


Fig 4.1 fenêtre de commande pendant un processus avec 3 workers

La figure 4.1 montre un exemple de processus en cours d'exécution avec 3 workers en parallèle. Sur la droite, la fenêtre d'état de progression est montrée, qui nous informe sur l'état d'exécution lors de l'évaluation des mesures .

5. Charge et comparaisons de données

La charge de données et les comparisons sont des tâches complémentaires sans besoin de parallelisation. La charge de données est exécutée trois fois pour charger l'image source, la GT et les images binaires obtenues par chaque processus de segmentation à partir d'un subfolder A, les horaires de traitement stockés pour chaque méthode de segmentation sont également chargés. Le chemin vers les images peut être introduit manuellement ou prédéfinis. Les images chargées sont utilisées pour la présentation des résultats provisoires et pour calculer les paramètres d'évaluation tels que décrits dans la section suivante.

Après l'exécution des segmentations, les images binaires résultantes sont enregistrées dans un répertoire nommé résultats dans un subfolder appelé seg binaire.

6. Évaluation de la qualité de segmentation et de l'exactitude de la détection

L'évaluation de la qualité de segmentation et de la précision de détection est la deuxième tâche principale développée dans cet outil. La comparaison de tous les résultats de segmentation avec la GT est effectuée en analysant la coïncidence pixel. Par conséquent, TP, FP, FN et TN sont calculés en premier.

La qualité de segmentation est ensuite évaluée par les mesures de GCE, OCE, RI et VoI tandis que la précision de détection est mesurée avec des mesures p, r,FPR, JI, JDI, DI et Fs. Par conséquent, cinq cas exécutés en parallèle sont créés de la manière suivante pour répartir le temps de calcul symétriquement

Case 1: GCE
Case 2: OCE
Case 3: RI, VoI
Case 4: p, r, FPR, JI, JDI, Fm,
Case 5: stockage de données, présentation de résultats.

Les cas 1, 2 et 3 contiennent des algorithmes à haute complexité informatique, le cas 4 joint plusieurs calculs qui utilisent les valeurs précédemment obtenues pour TP, FP, FN et TN. Enfin, le dernier cas a été défini pour les tâches complémentaires comme stockage de données, et ainsi de suite. Tous les cas sont attribués avec l'aide de l'instruction parpool comme dans la partie segmentation.

7. Résultat

Nous offrons graphiquement et numériquement les résultats obtenus par le processus multithread et stockons les segmentations et la détection automatique au format .dicom. Les résultats liés à la qualité de la segmentation et à la détection de la précision sont enregistrée dans les fichiers .xls .

Les figures 4.2 , 4.3 ,4.4 montre comment les résultats sont affichés pour l'exemple d'évaluation sur une image de test .

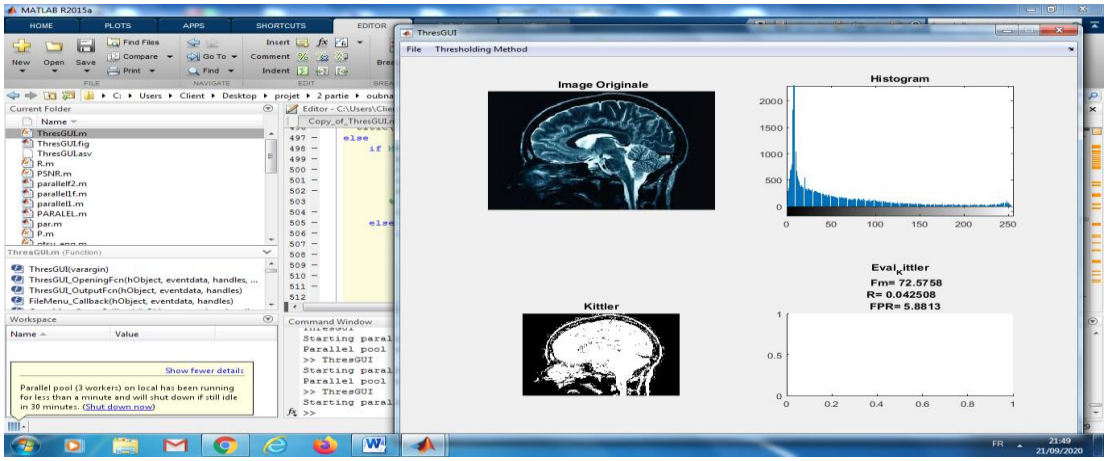


FIG 4.2 kittler

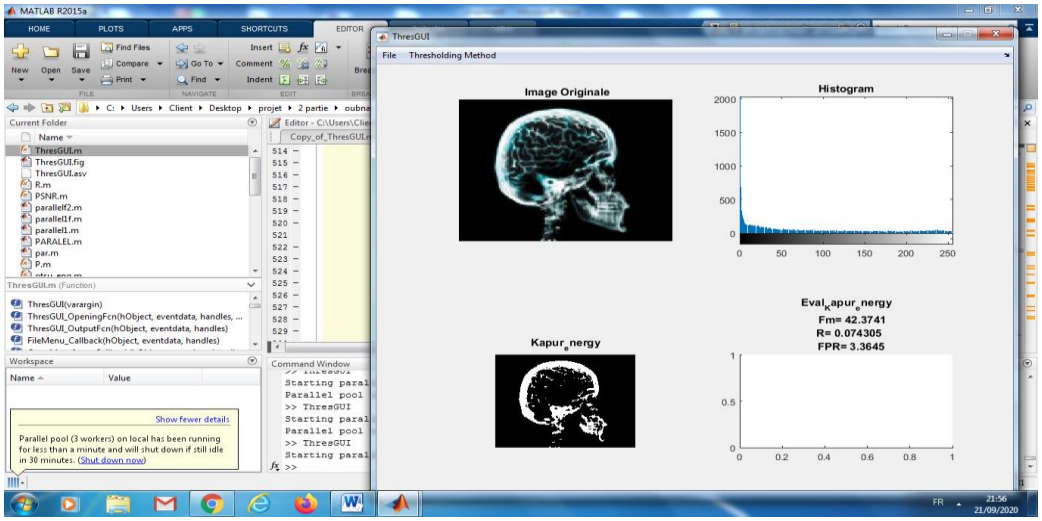


FIG 4.3 kittler energie

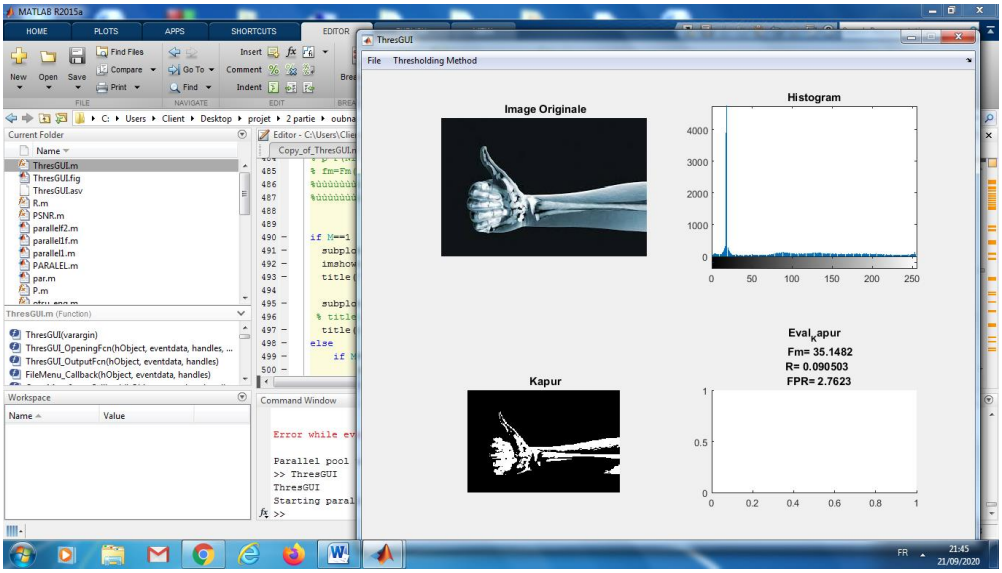


FIG4.3 kapur

Afin de présenter les résultats de manière équilibrée, nous avons utilisé un schéma distribué. Les images de segmentation stockées dans le sous-dossier sont considérées comme des résultats. Grâce à de nouveaux multithreads, le logiciel présenté génère les contours dans les images binaires, qui se chevauchent ensuite avec les images sources. Cette procédure est mise en œuvre en parallèle pour chaque méthode de segmentation évaluée. Les résultats de l'évaluation sont apparus dans le même schéma .

8. Conclusion

Dans ce chapitre, nous avons exploiter la boîte à outils de calcul parallèle MATLAB qui nous a facilité le déploiement du parallélisme dans les parties intensives en calcul d'un programme et l'évaluation de performance de différents algorithmes dans le même système,

Conclusion générale

Dans notre travail, nous nous sommes interessées à l'evaluation de performance de cinq méthodes de segmentation à base de seuillage, pour detecter des régions d'interets dans des images medicales. Le processus d'évaluation des resultats des methodes de segmentation considérés, par un ensemble de metriques avec l'utilisation d'une verité de terrain (segmentation de reference), ont été mis en oeuvre en parallèle, en exploitant les outils offerts par la boîte à outils Matlab de calcul parallèle, qui prennent en charge les modèles de programmation parallèle suivants: parallèlisme de données, mémoire distribuée (passage de messages), données multiples à programme unique (SPMD) et données multiples à programme multiple (MPMD).

Les resulats d'évaluation obtenus sont interessants et montrent l'importance de la boîte à outils Matlab de calcul parallèle pour l'évaluation parallele de performance des methodes de segmentation, ceci nous encorage de continuer le travail dans ce domaine, dans le future, en utilisant d'autres methodes de segmentation et d'autres metriques d'evaluation.

Bibliographie

[1] Christophe Rosenberger , Contribution à l'évaluation d'algorithmes de traitement d'images,Interface homme-machine , Université d'Orléans, 2006.

[2] Henry Cruz, Martina Eckert, Juan M. Meneses, and J. F, Fast Evaluation of Segmentation Quality with Parallel Computing , Martínez 2017

[3] Pierre LAURENT, Plateforme d'évaluation multi-critères pour les algorithmes de traitement d' images médicales MONTRÉAL, LE 24 JANVIER 2017

[4] Walid Barhoumi Ezzeddine Zagrouba Evaluation de la segmentation des images médicales , une méthode supervisée utilisant conjointement l'information région et contour Institut Supérieur d' Informatique , 2 Rue Abou Rayhane El Bayrouni, 2080 Ariana,Tunisie, Conference Paper January 2006.

[5] Sébastien CHABRIER , Evaluation de résultats de segmentation d'images , Massachusetts Institute of Technology.

[6] Jose Aguilar , Introduction to Parallel Computing , University of the Andes (Venezuela) 2004 .

[7] Michael Skuhersky , Introduction to Parallel Computing ,Janury 2005.

[8] Naoui Moulkheir, Segmentation d'images par modelles statistiques de forme et d'apparence Université d'Oran.

[9] YAHI Amira, Implantation de la chaîne inter-prédiction utilisée dans l'encodeur H.264/AVC sur une plateforme multi-composants , Université Badji Mokhtar d'Annaba.

[10] Intel Hyper-Threading Technology Technical User's Guide, Janvier 2003
http://www.utdallas.edu/edsha/parallel/2010S/Intel-HyperThreads.pdf

[11] Nicolas Ventroux,Contrôle en ligne des systèmes multiprocesseurs hétérogènes embarqués ,élaboration et validation d'une architecture , thèse, École Doctorale MATISSE, UNIVERSITÉ DE RENNES 1,2006.

[12] Henry Cruz, Martina Eckert, Juan M. Meneses, and J. F. Martínez , Fast Evaluation of Segmentation Quality with Parallel Computing , 2017

[13] I. Hacihaliloglu, R. Abugharbieh, A. Hodgson, and R. Rohling, "Bone surface localization in ultrasound using image phase-based features," Ultrasound in Medicine & Biology, vol. 35, no. 9, pp. 1475–1487, 2009.

[14] D. Pham, C. Xu, and J.Prince, "A survey of current methods in medical image segmentation". Annual Review of Biomedical Engineering, 2:315–337. (2000)

[15] Y. J. Zhang, "An Overview of Image and Video Segmentation in the Last 40 Years". PP1-16, in Y.J. Zhang (réd.) Advances in Image and Video Segmentation. IRM Press, Hershey (Penn.), (2006).

[16] Y J Zhang "A Survey On Evaluation Methods For Image Segmentation". Pattern Recognition Vol 29 No 8 pp 1335-1346(1996).

[17] Jérome Cadieux Yassine Ariba, Manuel matlab , Icam de Toulouse.

[18] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 9, 1, 1979, pp. 62-66.

[19] J.N. Kapur, P.K. Sahoo and A.K.C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram", Journal of Computer Vision Graphics Image Processing, 29, 1985, pp. 273-285.

[20] J. Kittler and J. Illingworth.1986 .Minimum Error Threshol- ding." Pattern Recognition, 19(1), 41-47.

[21] P. Swarnajyoti et al. A novel context sensitive multilevel thresholding for image segmentation Applied Soft Computing 23 (2014) 122–127