

# Segmentation d'images texturées par régularisation de classificateurs

Les tests réalisés dans le chapitre précédent montrent qu'un réglage fin des paramètres de notre méthode de segmentation par régularisation de graphes est nécessaire afin d'optimiser sa précision. Procéder à ces réglages requiert cependant de posséder un certain *a priori* concernant les caractéristiques de textures utilisées. Malheureusement, celui-ci n'est pas nécessairement disponible ou exploitable et, sous réserve de l'obtention d'un tel *a priori*, trouver les paramètres optimaux reste – comme le montrent nos tests – une opération complexe pour laquelle il n'existe actuellement aucune méthode non-supervisée.

Nous avons par ailleurs vu qu'une sélection des caractéristiques les plus pertinentes permet d'optimiser la méthode, aussi bien en ce qui concerne la qualité des résultats que la vitesse de convergence. Pour des raisons pratiques – exposées section 2.2 – nous avons réduit à deux le nombre de paramètres requis par notre méthode<sup>1</sup>. Il devient donc impossible d'inhiber « algorithmiquement » certaines caractéristiques de textures en leur attribuant un poids nul ; seule une modification des données d'entrée permet d'y parvenir.

Les travaux que nous présentons dans ce chapitre visent à corriger ces lacunes tout en conservant les deux approches non-locales sur lesquelles se base notre précédente méthode : les caractéristiques de textures et la régularisation de graphes. Afin d'y parvenir, nous devons permettre au seul élément en relation avec les caractéristiques de textures, le terme d'attache aux données, de prendre en compte la pertinence de chacune des caractéristiques.

Une solution consisterait à revenir au critère d'attache aux données de Chan et Vese pour images multispectrales [Chan *et al.*, 2000]. Or, alors que le nombre de canaux dans une image multispectrale est généralement faible, le nombre de caractéristiques de textures que nous sommes amenés à considérer peut être relativement important. Il est donc inenvisageable d'aboutir manuellement à un paramétrage optimal de ce critère.

Un problème de segmentation d'images peut être vu comme un problème de classifi-

---

1. L'algorithme de Chan et Vese pour images multivaluées originel requiert deux paramètres pour chaque caractéristique de textures.

cation de données. L'objectif est en effet d'attribuer une classe à chacun des pixels d'une image d'après un vecteur de caractéristiques. Dans la méthode que nous avons présentée précédemment, le terme d'attache aux données est l'élément qui a pour rôle d'indiquer si un vecteur de caractéristiques de textures est typique de l'objet à identifier ou bien de l'arrière-plan. Son rôle est donc similaire à celui d'un classificateur binaire. Nous pouvons d'ailleurs noter que Ta évoque un lien entre le modèle de Chan et Vese (dans sa version scalaire) et l'algorithme des k-moyennes [Ta, 2009].

Afin de permettre une meilleure modélisation du problème, nous proposons de nous baser sur un classificateur supervisé. L'idée est alors d'intégrer la réponse de ce classificateur au terme d'attache aux données.

Nous commencerons ce chapitre par un état de l'art sur la classification de données, en s'intéressant plus particulièrement aux méthodes supervisées. Nous détaillerons ensuite notre approche, puis terminerons par une présentation des résultats obtenus sur différents types d'images : textures artificielles et images médicales 2D et 3D.

Les résultats de ces travaux ont été publiés dans [Faucheux *et al.*, 2013].

## 3.1 Classification supervisée de données

La classification de données est une branche de l'apprentissage automatique. Ce domaine vise à produire des « machines » qui, à partir d'un jeu de données caractéristique d'un système, sont capables d'inférer un ensemble de règles visant à modéliser le système dans son ensemble. De manière générale, cette modélisation permet d'associer un label à chaque individu de l'espace considéré.

Ces techniques s'opposent donc aux systèmes experts, des outils d'aide à la décision conçus pour appliquer un ensemble de règles à de nouveaux jeux de données. À la différence des méthodes d'apprentissage automatique, les systèmes experts requièrent que les règles aient au préalable été fixées par un expert humain, selon un formalisme spécifique à l'outil. L'intérêt des systèmes experts a cependant décliné avec la généralisation des outils informatiques : ces derniers permettant d'obtenir facilement de grands volumes de données, il est devenu impossible pour un expert humain de fournir une modélisation pertinente d'un système complexe.

Les méthodes d'apprentissage automatique se divisent en plusieurs catégories qui identifient le mode d'apprentissage. Les trois principales sont : l'apprentissage supervisé, l'apprentissage non-supervisé et l'apprentissage semi-supervisé. Nous allons voir que lorsque les données sont représentées à l'aide de graphes, ou encore que la modélisation du système est soumise à certaines contraintes, il est alors possible d'établir un pont entre certaines de ces catégories.

En apprentissage supervisé, un ensemble d'apprentissage constitué de  $n$  individus et de leurs labels respectifs est requis. Cet ensemble, choisi pour être le plus représentatif possible des différentes classes que l'on souhaite identifier, est utilisé pour entraîner l'algorithme et générer la modélisation du problème. En comptant sur la capacité du modèle à généraliser ce qu'il a assimilé durant la phase d'apprentissage, celui-ci peut ensuite être appliqué à tout nouvel individu d'un ensemble de test afin de lui associer son propre label.

### 3.1. CLASSIFICATION SUPERVISÉE DE DONNÉES

---

Un algorithme d'apprentissage est qualifié de non-supervisé lorsque son objectif est de détecter des structures cachées dans un ensemble de test. Puisque ces structures sont *a priori* inconnues, l'algorithme se base sur des notions de proximité et d'homogénéité afin d'identifier des groupes composés des données les plus similaires.

L'apprentissage semi-supervisé consiste à utiliser à la fois des données labellisées et non-labellisées pour entraîner l'algorithme. Afin de compléter l'information contenue dans l'ensemble d'apprentissage, des notions de régularité ou de proximité sont utilisées. La labellisation automatique de certaines données – qui viendront s'ajouter à l'ensemble d'apprentissage initial – permet en particulier de rendre un algorithme supervisé plus robuste. Cette technique minimise en effet l'éventualité d'un sur-apprentissage lorsque les données labellisées ne sont disponibles qu'en faible quantité [Zhu, 2005, Chapelle et Schölkopf, 2006].

Certaines méthodes d'apprentissage supervisé peuvent être vues comme des problèmes d'interpolation : des marqueurs initiaux sont fournis pour un ensemble d'individus qui doivent être utilisés pour calculer le label d'individus « intermédiaires ». Le problème est alors résolu par diffusion des labels présents dans l'ensemble d'apprentissage aux individus non-labellisés. Cette diffusion se déroule de proche en proche, en suivant les règles de voisinage imposées par la structure des données.

L'une des hypothèses sur lesquelles s'appuie l'apprentissage semi-supervisé est que, dans un espace de grande dimension, les données reposent en réalité dans une variété de faible dimension. Cela s'illustre aisément à l'aide d'un graphe de similarité : les nœuds similaires sont généralement situés dans des sous-graphes présentant une forte connexité (voir figure 3.1). La topologie du graphe constitue ainsi une représentation implicite de ces variétés, permettant à un processus de diffusion de se dérouler selon un principe géodésique (en suivant la structure des données) et non plus selon un principe Euclidien. L'algorithme est donc capable de tirer parti des relations qui ont été établies avec des données potentiellement non labellisées pour mener à bien sa tâche. Pour cette raison, les méthodes d'apprentissage automatique qui se basent sur des graphes sont souvent qualifiées de semi-supervisées.

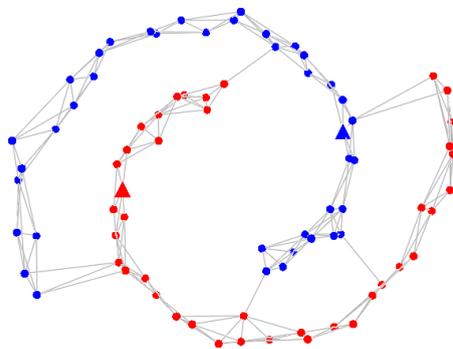


FIGURE 3.1 – Illustration de la notion d'apprentissage semi-supervisé dans un graphe. Ce graphe de similarité est un graphe des 5-plus proches voisins. Les nœuds en forme de triangle composent l'ensemble d'apprentissage. Les relations de voisinage mettent en évidence la structure des données, ce dont un algorithme d'apprentissage pourra tirer parti afin d'identifier les deux spirales.

Une autre hypothèse de l'apprentissage semi-supervisé concerne la régularité des labels : si deux données sont proches dans l'espace d'entrée, alors leurs labels doivent être similaires. Étant donné que les algorithmes de régularisation imposent ce type de contraintes, ils peuvent eux aussi être qualifiés de semi-supervisés.

L'approche que nous avons présentée dans le chapitre précédent repose à la fois sur l'utilisation de graphes pour représenter les données et sur une méthode de régularisation. Elle entre donc pleinement dans la catégorie des algorithmes semi-supervisés. Nous pouvons cependant voir un lien avec une quatrième catégorie d'algorithmes, moins répandue : l'apprentissage partiellement supervisé [Ambroise et Govaert, 2000]. Ces algorithmes ont la particularité de pouvoir travailler à partir d'informations incomplètes, par exemple « l'individu  $x$  n'appartient pas à la classe A ». Notre méthode de segmentation par régularisation de graphes entre dans cette catégorie car l'ensemble d'apprentissage que nous définissons ne concerne qu'une seule classe. Le critère d'attache aux données de Chan et Vese ne dispose donc initialement que d'une modélisation partielle du problème, mais celle-ci est affinée au fil des itérations.

Dans l'approche que nous proposons dans ce chapitre, nous nous appuyons sur un classificateur afin de modéliser les textures à identifier.

Les méthodes non supervisées ont été exclues puisqu'elles se basent sur des critères d'homogénéité ou de densité pour détecter les différentes classes. Or, les données que nous souhaitons manipuler en premier lieu sont des images, dans lesquelles les données sont réparties dans l'espace avec une densité constante. De plus, nous employons des caractéristiques de textures qui, du fait de leur non-localité, induisent un flou dans l'espace de représentation des données. Il est ainsi relativement complexe d'exprimer des critères d'homogénéité.

En traitement d'images, les données sont généralement disponibles en quantité suffisante pour permettre la construction d'un ensemble d'apprentissage pertinent. Nous n'avons donc pas jugé nécessaire de recourir à des approches semi-supervisées.

Nous avons ainsi opté pour l'utilisation d'un classificateur supervisé. Dans la suite de cette section, nous présentons les trois classificateurs supervisés les plus populaires : la méthode des  $k$  plus proches voisins, les réseaux de neurones artificiels et les machines à vecteurs supports. Nous pouvons d'ores et déjà noter qu'aucun de ces classificateur ne tire parti de la structure de graphe sur laquelle repose le processus de régularisation. Ces classificateurs ne peuvent donc pas être vus comme des approches semi-supervisées.

#### 3.1.1 Méthode des $k$ plus proches voisins

La méthode des  $k$  plus proches voisins<sup>2</sup> est l'un des algorithmes de classification de données les plus simples [Fix et Hodges, 1951]. Cette méthode ne repose pas sur un algorithme d'apprentissage à proprement parler, mais sur une heuristique qui réalise une classification par analogie : la classe attribuée à un individu inconnu correspond à la classe des individus connus les plus similaires. Plus précisément, il s'agit d'appliquer le principe du vote majoritaire à partir des  $k$  individus de l'ensemble d'apprentissage les plus proches.

Il n'y a donc pas de phase d'apprentissage, tous les calculs sont effectués lors la phase

---

2. Souvent abrégée  $k$ -PPV, ou  $k$ -NN, de l'anglais *k-nearest neighbors*.

### 3.1. CLASSIFICATION SUPERVISÉE DE DONNÉES

---

de classification. Pour chaque individu à classer, il est nécessaire de calculer la distance entre cet individu et ceux présents dans l'ensemble d'apprentissage. Plusieurs mesures de distances sont couramment employées : distance Euclidienne ou de Manhattan pour les données numériques [Weijer et Schmid, 2006], distance de Hamming pour les données symboliques [Smith *et al.*, 1994] ou encore distance de Mahalanobis lorsque l'on est en présence de données bruitées [Yates *et al.*, 2003].

Étant dépourvu de phase d'apprentissage, ce classificateur est apprécié pour sa capacité à assimiler instantanément de nouvelles connaissances. Cependant, derrière l'apparente simplicité de cette méthode, des précautions doivent être prises quant à la constitution de l'ensemble d'apprentissage.

Tout d'abord, il est important de disposer de données dont les attributs sont normalisés. Les mesures de distance traditionnelles ne permettent pas de tenir compte de l'amplitude des valeurs selon chacune des dimensions. Appliqué à notre problématique, cela signifie qu'une caractéristique de textures présentant de faibles variations pourrait être masquée par une caractéristique de plus forte amplitude.

De plus, les individus présents dans cet ensemble doivent être identiquement distribués dans leur espace. En effet, étant donné que seuls les  $k$  plus proches voisins sont considérés, il y a un risque d'introduire un biais dans la classification si une classe est statistiquement plus représentée qu'une autre.

Enfin, lorsque l'espace des individus est composé d'un nombre important de dimensions, les mesures de distance perdent de leur intérêt [Beyer et Goldstein, 1999]. Cette méthode est ainsi sujette à ce qui est couramment appelé la « malédiction de la dimension ». Afin de remédier à cela, il convient d'utiliser des techniques de sélection de caractéristiques, telles que l'analyse en composantes principales. Cependant, cela revient à la mise en place d'une phase d'apprentissage.

#### 3.1.2 Réseaux de neurones artificiels

Un neurone biologique est un élément de base du système nerveux. Il s'agit d'une cellule qui peut être excitée par une impulsion nerveuse et qui, en fonction de son excitation, est capable d'émettre de nouvelles impulsions. Les neurones sont présents en grande quantité dans un cerveau, et sont impliqués dans l'essentiel des fonctions cognitives d'un être vivant. Le cerveau fonctionne en effet selon une approche connexionniste : des processus complexes émergent de réseaux d'unités fonctionnelles simples interconnectées (les neurones).

Les réseaux de neurones artificiels constituent une classe d'outils de modélisation inspirés du fonctionnement du cerveau. Introduits dans [Rosenblatt, 1958], le principe est d'interconnecter de nombreuses unités de calcul simples, appelées neurones formels, afin de permettre la résolution de problèmes complexes.

Présenté dans [Rumelhart *et al.*, 1986], l'un des modèles de réseaux de neurones les plus populaires est le perceptron multicouche<sup>3</sup>, connu pour être un approximateur universel de fonctions [Hornik *et al.*, 1989, Hartman *et al.*, 1990]. Il s'agit d'un réseau à propagation avant composé de plusieurs couches de neurones de type perceptron. Chaque neurone représente une unité de calcul capable de résoudre un problème de séparation linéaire,

---

3. Aussi appelé MLP, de l'anglais *Multi Layer Perceptron*.

### 3.1. CLASSIFICATION SUPERVISÉE DE DONNÉES

et reçoit en entrée l'ensemble des valeurs calculées par les neurones de la couche qui le précède (sauf la couche d'entrée qui reçoit les données du problème). Cette interconnexion de neurones permet la résolution de problèmes non-linéaires. La figure 3.2 ainsi que les formules (3.1) et (3.2) décrivent le fonctionnement d'un perceptron d'indice  $j$  :

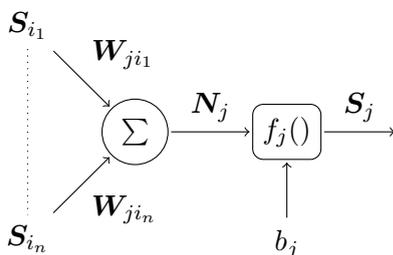


FIGURE 3.2 – Schéma d'un neurone formel  $j$  de type perceptron.

$$N_j = \sum_{i \in I} s_i \cdot w_{ji} \quad (3.1)$$

$$s_j = f_j(N_j - b_j) \quad (3.2)$$

avec  $I = \{s_1 \dots s_n\}$  l'ensemble des entrées de ce neurone,  $\{w_{j1} \dots w_{jn}\}$  les poids associés aux entrées,  $b_j$  un biais et  $f_j()$  une fonction de transfert, généralement une fonction de type sigmoïde ou log-sigmoïde.  $s_j$  représente la sortie de ce neurone.

Un MLP est ainsi constitué de plusieurs couches. La première est appelée couche d'entrée et reçoit les données du problème, tandis que la dernière est appelée couche de sortie et contient la réponse du réseau. Les couches intermédiaires sont appelées couches cachées. Chaque couche d'un MLP est fortement connectée à la couche précédente : les entrées d'un neurone correspondent aux sorties de ceux de la couche précédente<sup>4</sup>. Le nombre de neurones sur la couche d'entrée est égal à la dimension du problème, tandis que le nombre de neurones de sortie peut varier en fonction de sa modélisation. Le nombre de couches et de neurones cachés est à fixer en fonction de la complexité du problème. La figure 3.3 schématise la structure d'un MLP composé de quatre entrées, une couche cachée de cinq neurones et d'une unique sortie.

Un réseau de neurones artificiels nécessite un apprentissage afin de résoudre un problème particulier. Cette phase revient à fixer les poids et les biais de chacun des neurones. L'algorithme de rétropropagation du gradient [Rumelhart *et al.*, 1986], la première méthode permettant d'entraîner un perceptron multicouche, est encore largement utilisée aujourd'hui. Elle consiste à présenter chaque individu de l'ensemble d'apprentissage au réseau de neurones. Les valeurs obtenues en sortie sont ensuite comparées à celles attendues et les poids et biais de chaque neurone sont mis à jour selon leur influence sur l'erreur. Ce processus est exécuté itérativement jusqu'à ce que la modélisation soit suffisamment vraisemblable.

Il est ainsi courant de diviser l'ensemble des données disponibles pour l'apprentissage en deux sous-ensembles : l'ensemble d'apprentissage et l'ensemble de validation. Le premier

4. Sauf les neurones de la couche d'entrée qui reçoivent les données du problème.

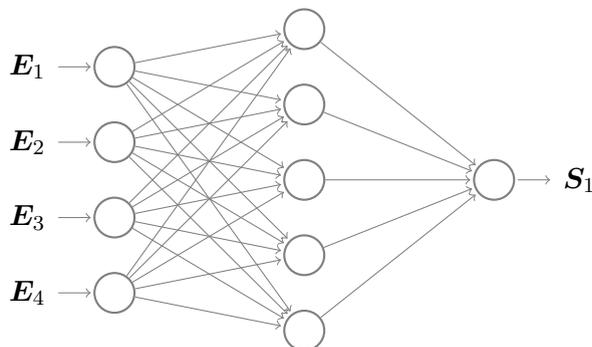


FIGURE 3.3 – Structure d’un perceptron multicouche composé de quatre entrées, une couche cachée de cinq neurones et d’une unique sortie.

reste utilisé pour entraîner le réseau alors que le second permet de mesurer l’aptitude du réseau à généraliser et ainsi de prévenir le sur-apprentissage. Un troisième sous-ensemble, l’ensemble de test, peut aussi être utilisé afin de choisir le nombre de couches et de neurones cachés que doit comporter le réseau pour modéliser au mieux et le plus efficacement le problème.

### 3.1.3 Machines à vecteurs supports

Les machines à vecteurs supports<sup>5</sup> constituent l’une des plus récentes approches de la classification supervisée de données. Elles sont notamment reconnues pour leur bonnes performances vis à vis du traitement de problèmes non linéairement séparables [Suykens, 2001]. Ce résultat s’oppose cependant à l’une des théories sur lesquelles cette approche repose : la notion de marge maximale.

Étant donné un jeu de données composé de deux classes d’éléments et un algorithme chargé de calculer la frontière séparant les données, la marge est définie comme la distance minimale qui existe entre la frontière de séparation et les données les plus proches (voir figure 3.4). La maximisation de cette marge constitue une heuristique dont le but est de réduire l’erreur de généralisation de l’algorithme d’apprentissage en charge du calcul de la frontière : plus la marge est importante, moins l’algorithme a de chances de mal classer un individu inconnu.

Dans une machine à vecteurs supports, cette notion de marge maximale est appliquée au calcul d’un hyperplan séparateur. Soit  $\mathbb{R}^n$  l’espace de définition des données. Un hyperplan de cet espace correspond à l’ensemble des  $\mathbf{x} \in \mathbb{R}^n$  tel que :

$$\mathbf{w}^T \cdot \mathbf{x} + w_0 = 0 \quad , \quad (3.3)$$

avec  $\mathbf{w} = (w_1 \dots w_n)$  et  $w_0$  ses paramètres.

Soit  $X = \{(\mathbf{x}_1, c_1) \dots (\mathbf{x}_p, c_p)\}$  un ensemble d’apprentissage avec  $c_k \in \{-1, +1\}$  la classe d’appartenance de l’individu  $\mathbf{x}_k$ . Le calcul de l’hyperplan de marge minimale est

5. Aussi appelées SVM, de l’anglais *Support Vector Machines*. Par analogie avec l’acronyme anglais, le terme Séparateurs à Vaste Marge est parfois employé.

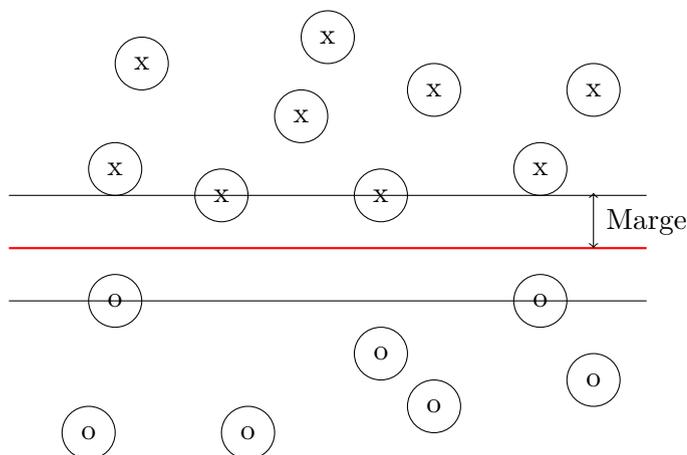


FIGURE 3.4 – Principe de marge maximale. Toute frontière de séparation présentera une marge inférieure à celle représentée par la ligne rouge.

obtenu par résolution du problème suivant :

$$\arg \max_{\mathbf{w}, w_0} \left\{ \frac{1}{\|\mathbf{w}\|} \min_k (c_k(\mathbf{w}^T \cdot \mathbf{x}_k + w_0)) \right\} . \quad (3.4)$$

Il s'agit d'un problème d'optimisation quadratique que de nombreuses méthodes permettent de résoudre. Nous pouvons notamment citer celles basées sur les algorithmes génétiques [Pai et Hong, 2005, Zhang *et al.*, 2009] ou encore les méthodes de recuit simulé [Lin *et al.*, 2008, Zhang *et al.*, 2009].

Après calcul de l'hyperplan séparateur, la classification d'un nouvel individu est obtenue en mesurant sa distance à l'hyperplan, par projection orthogonale. Le signe de cette distance est alors indicatif de la classification.

Dans cette configuration, l'hyperplan n'est calculé qu'à partir des données les plus proches de la frontière. Il suffit donc d'une donnée bruitée ou erronée à proximité de la frontière pour décaler l'hyperplan séparateur, réduisant ainsi la capacité de généralisation du classificateur. Afin de rendre les machines à vecteurs supports plus robustes, une configuration relaxée du problème appelée « marges souples » fut proposée [Cortes et Vapnik, 1995]. Celle-ci autorise certains éléments de l'ensemble d'apprentissage à se trouver à une distance de l'hyperplan inférieure à la marge, ou même à être mal classés.

La frontière étant un hyperplan, l'approche par marge maximale ne peut être appliquée avec succès qu'à des problèmes linéairement séparables. L'originalité des machines à vecteurs supports est due à Vapnik qui proposa de remédier à cette limitation en projetant les données d'entrée dans un espace de plus grande dimension [Vapnik, 1995]. Comme l'illustre la figure 3.5, cette opération peut en effet permettre de rendre linéairement séparable un problème qui ne l'est pas dans son espace d'origine. Cet opérateur de projection s'appelle une fonction noyau.

À l'inverse de nombreux algorithmes de classification qui tentent de calculer une frontière non linéaire, ce sont ici les données qui sont modifiées afin de permettre la résolution d'un problème « plus simple ». Cette innovation n'est cependant pas sans défaut puisqu'en

### 3.1. CLASSIFICATION SUPERVISÉE DE DONNÉES

---

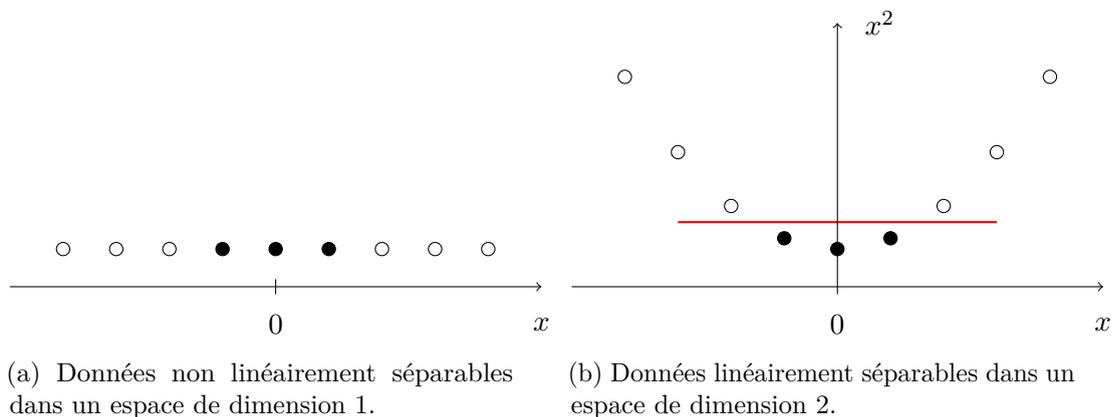


FIGURE 3.5 – Exemple de problème qui n'est linéairement séparable que lorsque les données sont projetées dans un espace de plus grande dimension.

l'absence d'*a priori* concernant les données, il est nécessaire de recourir à des fonctions noyau qui maximisent la probabilité de rendre le problème linéairement séparable. L'espace dans lequel sont projetées les données peut ainsi être de très grande dimension, rendant le problème insolvable. La solution consiste à appliquer « l'astuce du noyau » (*kernel trick*) [Aizerman *et al.*, 1964]. Pour des fonctions noyau respectant certaines propriétés, cette astuce rend inutile l'étape de projection, réduisant ainsi considérablement la complexité des opérations.

#### 3.1.4 Bilan des méthodes de classification supervisée

La méthode des  $k$  plus proches voisins est souvent choisie pour sa simplicité de mise en œuvre. Nous ne l'avons cependant pas retenue pour plusieurs raisons, essentiellement liées à l'absence de phase d'apprentissage capable de synthétiser les données.

Le but recherché par l'ajout d'un classificateur à notre méthode de segmentation est de permettre une pondération automatique des différentes caractéristiques de textures. L'algorithme des  $k$  plus proches voisins étant uniquement basé sur une comparaison de mesures de distance, il est, au même titre que le critère d'attache aux données de Chan et Vese, incapable d'identifier la pertinence des différentes caractéristiques de textures et donc de les pondérer. Nous sommes par ailleurs amenés à travailler avec un nombre important de caractéristiques de textures, ce qui est un contexte défavorable aux  $k$ -PPV puisque ce classificateur est sujet à la malédiction de la dimension. Nous pouvons aussi noter que la classification d'un individu est un processus relativement long, puisque cela nécessite de calculer sa distance à chacun des éléments de l'ensemble d'apprentissage.

La littérature cite régulièrement les mérites des machines à vecteurs supports, qui sont notamment reconnues comme sensiblement plus performantes que les réseaux de neurones, tant en termes de précision que de rapidité d'exécution [Dal Moro *et al.*, 2006, Kumar *et al.*, 2011]. L'objectif de nos travaux n'étant pas l'évaluation des performances de classificateurs, nous nous sommes tournés vers celui qui nous est le plus familier : les réseaux de neurones artificiels. Ce choix nous permet notamment de limiter les risques liés à une mauvaise

configuration des classificateurs, et ainsi de nous focaliser sur leur intégration dans le processus de régularisation. La méthode que nous proposons reste compatible avec l'ensemble des classificateurs supervisés connu ce jour. Des travaux visant à permettre l'utilisation de classificateurs différents sont actuellement menés.

### 3.2 Contribution

Dans la méthode de segmentation d'images par régularisation de graphes présentée au chapitre précédent, le terme d'attache aux données de Chan et Vese fait office de classificateur binaire. En employant un classificateur supervisé tel que ceux présentés précédemment, nous sommes en mesure de gérer un nombre  $n$  arbitraire de classes. La méthode de segmentation que nous présentons ici a donc été conçue dans le but de permettre une segmentation multiclassée. Nous distinguerons cependant le cas où le nombre de classes est égal à deux du cas où il est supérieur à deux.

Toutes les images que nous allons traiter à l'aide de cette méthode doivent être converties sous forme de graphes de similarité selon l'une des techniques décrites section 1.3.

Nous considérons qu'il est possible d'obtenir – pour chaque objet que nous souhaitons identifier dans une image – un ensemble de nœuds dont les attributs de textures sont représentatifs de la texture des objets. Ces  $p$  nœuds constituent un ensemble d'apprentissage  $X = \{(\mathbf{x}_1, c_1) \dots (\mathbf{x}_p, c_p)\}$ , avec  $\mathbf{x}_k \in \mathbb{R}^m$  un vecteur de caractéristiques de textures et  $c_k \in [c_1 \dots c_n]$  un symbole qui identifie la classe associée.

#### 3.2.1 Segmentation biclasse

Nous nous plaçons ici dans le cadre où, à partir de caractéristiques de textures, nous souhaitons localiser dans une image un unique objet. Celui-ci se distingue donc de l'arrière-plan, d'où la présence de deux classes que nous nommerons  $c_1$  (objet) et  $c_2$  (arrière-plan).

À partir de l'ensemble d'apprentissage, nous sommes en mesure d'entraîner un classificateur. Celui-ci se formalise par une règle de décision  $f(\mathbf{x})$  qui associe à tout vecteur de caractéristiques de textures  $\mathbf{x} \in \mathbb{R}^m$  la probabilité d'appartenance à la première classe :

$$f(\mathbf{x}) = p(c_1|\mathbf{x}) \quad , \quad (3.5)$$

Les classes  $c_1$  et  $c_2$  étant complémentaires, la probabilité d'appartenance de  $\mathbf{x}$  à la seconde classe est donnée par :

$$p(c_2|\mathbf{x}) = 1 - p(c_1|\mathbf{x}) \quad . \quad (3.6)$$

Une fois l'apprentissage réalisé, le classificateur obtenu permet de traiter toute image présentant des textures similaires. La classification de chacun des nœuds du graphe de similarité associé à une image nous permet d'obtenir une segmentation probabiliste de celle-ci.

Du fait de la présence de bruit dans l'image, et donc de caractéristiques de textures sensiblement différentes, la segmentation retournée par le classificateur peut être assez

## 3.2. CONTRIBUTION

---

grossière. Cela est d'autant plus vrai que nous employons des caractéristiques de textures car la non-localité des attributs génère un flou dans les régions à proximité des frontières des objets. De faibles variations de la texture suffisent alors à induire le classificateur en erreur. Afin de corriger les imperfections ainsi générées, une solution consiste à régulariser cette segmentation. La sortie du classificateur étant une fonction définie sur l'ensemble des nœuds du graphe ( $f \in \mathcal{H}(V)$ ), nous pouvons la considérer comme une nouvelle donnée sur laquelle nous proposons d'appliquer un algorithme de segmentation par régularisation de graphe. La figure 3.6 résume le fonctionnement de cet algorithme.

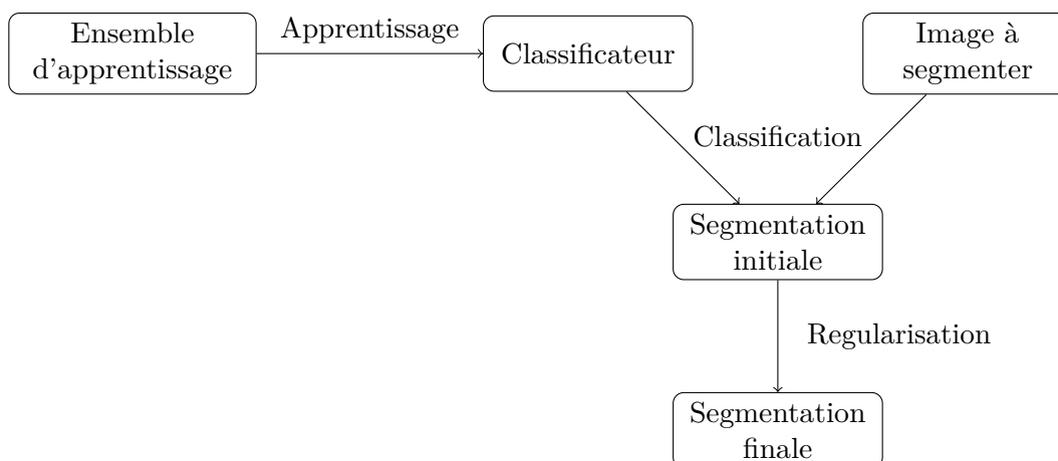


FIGURE 3.6 – Processus de segmentation bivalente par régularisation de classificateurs.

### 3.2.2 Segmentation multiclasse

Les méthodes de régularisation de graphes que nous avons vu précédemment (voir section 1.7.4) ne permettent de régulariser qu'une unique fonction. Dans un contexte de segmentation, le domaine de valeur de cette fonction est considéré comme binaire. Il existe donc une relation d'ordre qui permet de distinguer les deux régions associées aux valeurs 0 et 1 de la variable. Il est cependant impossible d'établir une telle relation d'ordre pour un nombre de classes supérieur à deux : ces algorithmes ne peuvent donc être utilisés que pour des problèmes de segmentation binaire.

Ta a suggéré de traiter les problèmes de régularisation de fonctions vectorielles à l'aide de plusieurs processus de régularisation distincts, chacun d'entre eux s'intéressant à une composante particulière de la fonction [Ta, 2009]. Afin de traiter un problème de segmentation multiclasse, nous allons appliquer une stratégie similaire.

Le classificateur associé désormais à tout vecteur de caractéristiques de textures  $\mathbf{x} \in \mathbb{R}^m$  un vecteur composé des probabilités d'appartenance<sup>6</sup> à chacune des  $n$  classes  $c \in \{c_1 \dots c_n\}$

---

6. Le terme probabilité est ici utilisé par abus de langage, la somme des valeurs obtenues pour chacune des classes n'étant pas nécessairement égale à 1. La majorité des classificateurs fonctionne en effet selon le principe de vote majoritaire :  $p(c_k|\mathbf{x})$  doit être vue comme une mesure de la validité de l'hypothèse « la texture  $\mathbf{x}$  est caractéristique de la classe  $c_k$  ». Ces mesures peuvent être comparées entre elles, mais ne correspondent pas à une mesure de probabilité.

## 3.2. CONTRIBUTION

connues du système :

$$f(x) = (p(c_1|\mathbf{x}), \dots, p(c_n|\mathbf{x})) \quad . \quad (3.7)$$

Après classification de l'ensemble des nœuds du graphe, les résultats obtenus pour chaque classe sont ensuite considérés indépendamment afin d'être régularisés. Nous faisons cependant l'impasse sur l'étape de seuillage pour permettre à une règle de décision de combiner les résultats des régularisation et ainsi attribuer une classe à chacun des nœuds. Dans nos tests, cette décision sera prise d'après un vote majoritaire. La figure 3.7 résume le fonctionnement de cet algorithme.

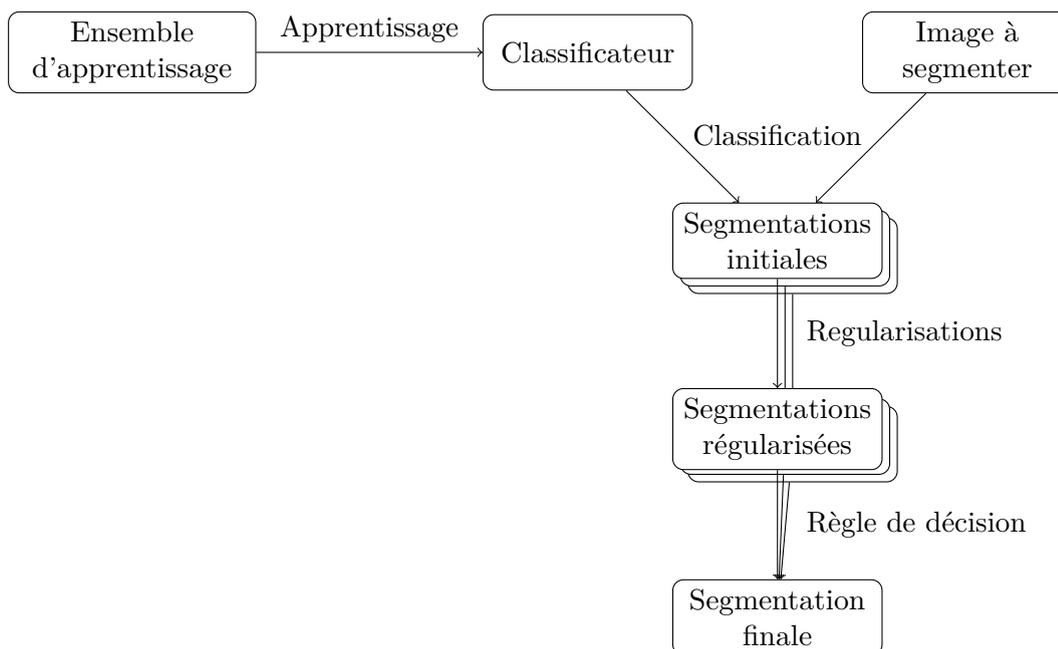


FIGURE 3.7 – Processus de segmentation multiclasse par régularisation de classificateurs.

Il est à noter que cet algorithme est tout à fait applicable à un problème de segmentation biclasse. Cependant, comme nous avons appliqué une stratégie « un contre tous », les deux segmentations retournées par le classificateur sont complémentaires<sup>7</sup>. Les traiter toutes les deux reviendrait alors à traiter deux fois la même information.

### 3.2.3 Architecture et entraînement des réseaux de neurones

Même si l'objectif de nos travaux n'est pas l'évaluation des performances des classificateurs, nous avons cherché à obtenir, pour chacun de nos tests, les réseaux de neurones permettant une résolution optimale des problèmes de classification.

Afin d'éviter tout phénomène de spécialisation, nous avons opté pour un apprentissage avec validation : deux tiers des éléments de la base d'apprentissage  $X = \{(\mathbf{x}_1, c_1) \dots (\mathbf{x}_p, c_p)\}$  sont aléatoirement sélectionnés afin d'être appris et de fixer les poids du réseau tandis que le tiers restant forme l'ensemble de validation et permet d'évaluer la capacité à généraliser du

7. En théorie. Dans la pratique, le classificateur présentera une certaine variabilité.

réseau. Après chaque cycle d'apprentissage du premier ensemble, la configuration du réseau est enregistrée et sa capacité à généraliser est évaluée sur l'ensemble de validation. Après un nombre prédéfini d'itérations (fixé à mille dans nos tests), la configuration présentant la plus faible taux d'erreurs de classification sur l'ensemble de validation est sélectionnée.

Nous avons également cherché à identifier l'architecture de réseaux de neurones la plus adaptée à nos problèmes de classification. Dans un contexte de classification binaire d'individus décrits par au plus dix attributs numériques, et à partir du processus d'apprentissage décrit précédemment, nous avons évalué les performances de réseaux de neurones composés de une et deux couches cachées, elles-mêmes composées de deux à cinq neurones. L'architecture présentant les meilleurs résultats ainsi que la plus faible complexité (en terme de nombre de poids à fixer) est alors retenue. Pour l'ensemble de nos tests, l'architecture composée d'une unique couche cachée de trois neurones permet l'obtention des meilleurs résultats de classification.

### 3.2.4 Choix de l'algorithme de régularisation

L'un des objectifs de cette approche étant de se libérer du problème de fixation des poids de l'approche précédente, notre première intention fut de revenir à l'algorithme proposé par Ta et basé sur le modèle de Chan et Vese (voir section 1.7.4.2). Les données que nous manipulons sont désormais scalaires et la pondération des deux classes est rendue inutile du fait de la présence d'un classificateur capable de gérer la question de l'homogénéité des textures. La présence d'un unique paramètre permettant d'établir un compromis entre régularité et précision de la solution reste nécessaire, mais sa détermination est relativement intuitive : il suffit de diminuer sa valeur pour augmenter la régularité de la segmentation. Des tests nous ont permis de valider le bon fonctionnement de cette approche.

Nous pouvons cependant simplifier le processus de régularisation. L'intérêt majeur du terme d'attache aux données de Chan et Vese est sa capacité à identifier la frontière de séparation entre deux classes d'individus selon un critère d'homogénéité. Or, c'est précisément la tâche accomplie par le classificateur. Celui-ci retourne pour chaque individu une valeur réelle comprise dans l'intervalle  $[0, 1]$ , 0.5 étant le seuil permettant de distinguer une classe d'une autre. Une simple transformation affine  $g : [0, 1] \rightarrow [-1, 1]$  permet alors de transformer la borne 0 en une valeur positive et la borne 1 en une valeur négative, tel que requis par la formulation du processus de régularisation (voir équation (1.43)) :

$$g(x) = 1 - 2x \quad , \quad (3.8)$$

avec  $x$  la sortie du classificateur.

Nous retrouvons ainsi la formulation du terme d'attache aux données utilisé par le modèle de Rudin, Osher et Fatemi (voir section 1.7.4.1). Les tests réalisés nous ont permis d'obtenir des résultats similaires à ceux du modèle de Chan et Vese, mais avec une amélioration sensible du temps de calcul. Il n'est en effet plus nécessaire de recalculer, avant chaque itération, les caractéristiques moyennes de chacune des régions. Les résultats présentés dans la section 3.3 sont obtenus avec ce modèle.

L'algorithme 2 résume la méthode de régularisation basée sur le modèle de Rudin, Osher et Fatemi.

### 3.2. CONTRIBUTION

---



---

**Algorithme 2 :** Algorithme d'approximation du modèle de Rudin, Osher et Fatemi.

---

**Entrées :** un graphe de similarité  $G = (V, E, w)$ , une fonction  $f_0 : V \rightarrow [0, 1]$  à régulariser,  $N$  un nombre d'itérations à réaliser,  $\lambda \in R^+$  le paramètre établissant le compromis entre la régularité et la précision de la solution.

**Sorties :** une fonction  $f : V \rightarrow [0, 1]$ , version régularisée de  $f_0$ .

**pour tous les**  $u \in V$  **faire**

└  $f^0(u) = f_0(u)$

**pour tous les**  $(u, v) \in E$  **faire**

└  $\beta_{uv} = \sqrt{w_{uv}} |f^n(u) - f^n(v)|^{-1}$

**pour**  $n = 0$  **a**  $N - 1$  **faire**

┌ **pour tous les**  $u \in V$  **faire**

$$\gamma = \frac{\sum_{v \sim u} \beta_{uv} \cdot f^n(v) - \lambda(1 - 2f^0(u))}{\sum_{v \sim u} \beta_{uv}}$$

└  $f^{n+1}(u) = \max(\min(\gamma, 1), 0)$

---

#### 3.2.5 Application à la segmentation automatique d'images

Un outil de segmentation d'images est souvent utilisé afin de traiter un grand nombre d'images de même nature. Nous pouvons prendre l'exemple d'un praticien qui aurait à analyser des lésions dont les images proviendraient d'un même imageur médical, ou encore celui d'une chaîne de production dans laquelle la conformité de chaque pièce doit être validée.

Ces images étant de même nature, il est envisageable de pouvoir compter sur un classificateur « expert » capable de traiter un problème particulier, et ainsi automatiser l'utilisation de notre méthode. Un tel classificateur est alors conçu par un expert à partir d'un ensemble de vérités terrain qu'il aura manuellement définies. Une fois le classificateur entraîné, celui-ci peut être sauvegardé et réutilisé. Aucune intervention humaine n'est alors nécessaire.

Le classificateur peut par ailleurs être conçu interactivement, sans nécessiter l'obtention de vérités terrain. Afin de définir l'ensemble d'apprentissage, l'utilisateur peut procéder à une labellisation partielle d'une ou plusieurs images. En entraînant le classificateur sur cet ensemble d'apprentissage, puis en exécutant le processus de segmentation sur ces mêmes images, il est possible de valider le fonctionnement du classificateur. Au cas où les résultats ne seraient pas satisfaisants, il est possible de consolider l'ensemble d'apprentissage en le corrigeant ou bien en y ajoutant de nouveaux pixels. Ce processus peut alors être répété jusqu'à obtention d'un résultat satisfaisant.

### 3.3 Résultats

#### 3.3.1 Segmentation d'images texturées artificielles

Nous avons dans un premier temps souhaité valider le fonctionnement de notre méthode sur une problématique de segmentation binaire. Nous avons pour cela généré un ensemble de seize images texturées, chacune étant composée d'un objet de forme aléatoire. Un exemple d'image est illustré figure 3.8.

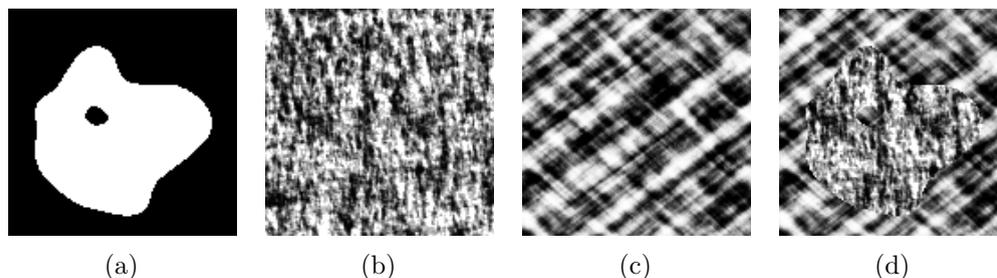


FIGURE 3.8 – Exemple d'image présentant un objet texturé de forme aléatoire. (a) : masque de composition. (b) et (c) : textures. (d) : image résultante.

Les textures employées proviennent de la base d'images Brodatz [Brodatz, 1966]. Nous n'avons cependant considéré que celles présentant des microtextures, les caractéristiques d'Haralick étant relativement peu adaptées aux macrotextures. Nous avons par ailleurs égalisé l'histogramme de chacune des images afin de rendre leurs distributions de niveaux de gris homogènes, rendant ainsi impossible de les discriminer à l'aide de statistiques de premier ordre.

La texture en chacun des pixels des images a été décrite à l'aide de coefficients d'Haralick, calculés dans une région de 11 pixels de diamètre et pour le décalage  $(1, 0)$ , les images ayant avant cela été réduites à 16 niveaux de gris.

Pour ce test, nous nous sommes placés dans le contexte où un classificateur serait fourni par un expert puis utilisé pour segmenter de multiples images de textures similaires. Nous avons donc entraîné seize classificateurs à reconnaître les textures utilisées dans chacune des images, puis sauvegardé leurs configurations.

Afin d'introduire une certaine variabilité dans les textures, les images ont ensuite été bruitées à l'aide d'un bruit blanc Gaussien de niveau croissant<sup>8</sup>. Cette altération a pour but de forcer le classificateur à généraliser ce qu'il a appris à des caractéristiques de textures encore inconnues. Selon le niveau de bruit, cette généralisation générera des erreurs de classification. Nous serons ainsi en mesure d'évaluer la capacité du processus de régularisation à les corriger.

La texture en chacun des pixels de ces images bruitées a été caractérisée selon les mêmes modalités. Chaque image a ensuite été segmentée pour différentes valeurs du paramètre  $\lambda$ , avec exécution du processus de régularisation jusqu'à l'obtention de solutions stables. Pour chaque image et chaque niveau de bruit, la segmentation présentant les meilleurs résultats

8. Le niveau d'un bruit Gaussien correspond à son écart-type.

### 3.3. RÉSULTATS

(en termes de distance générique d'anomalie) a été conservée.

Afin d'évaluer le bénéfice lié à la combinaison d'un processus de régularisation et d'un classificateur, nous avons comparé les résultats de notre méthode à ceux obtenus sans régularisation. Dans ce cas, la sortie du classificateur a été binarisée en considérant un seuil de 0.5.

Nous avons aussi comparé les performances de cette méthode à celle du chapitre précédent. Nous avons manuellement défini une initialisation pour chacune des seize images, sous la forme d'une ellipse située dans l'objet central. Afin de se placer dans les mêmes conditions d'utilisation, nous avons simplifié la première méthode en imposant  $\lambda_1 = \lambda_2$ . Nos deux approches ne comportent ainsi qu'un unique paramètre, dont la valeur optimale (en termes de distance générique d'anomalie) a été déterminée pour chaque niveau de bruit par recherche exhaustive sur un échantillon de valeurs.

Le tableau 3.1 présente les résultats obtenus pour ces trois méthodes sur l'ensemble des images pour le niveau de bruit 8.

Image	Classificateur seul	Régularisation de graphes	Régularisation de classificateurs
1	17.8	11.9	<b>6.3</b>
2	15.1	<b>7.1</b>	11.9
3	15.0	9.5	<b>7</b>
4	8.0	5.5	<b>4</b>
5	4.1	5.3	<b>2.2</b>
6	3.9	6.8	<b>3.7</b>
7	7.3	8.2	<b>4.8</b>
8	11.2	5.4	<b>4.5</b>
9	6.3	4.8	<b>3.9</b>
10	1.3	4.3	<b>0.9</b>
11	17.3	<b>14.5</b>	17.2
12	10.2	5.2	<b>4.2</b>
13	1.8	3.7	<b>1.6</b>
14	2.7	4.2	<b>2.3</b>
15	11.9	6.8	<b>3.2</b>
16	1.2	3.5	<b>1.5</b>
Moyenne	8.5	6.7	<b>4.9</b>
Écart type	5.6	3	4

TABLE 3.1 – Évaluation de la précision de la méthode de segmentation par régularisation de classificateurs sur la série d'images texturées artificielles. Les valeurs indiquées correspondent à la distance générique d'anomalie.

Étant donné que les classificateurs ont été entraînés à reconnaître les textures non bruitées, il leur est impossible de produire de bonnes classifications à partir d'images fortement altérées. Lorsque le niveau du bruit devient trop important, celui-ci n'est plus

### 3.3. RÉSULTATS

---

une source de variabilité mais représente une texture à part entière. Les résultats que nous évoquons par la suite ne concernent donc que les niveaux de bruit 2, 4, 6, 8, 10 et 12.

Les résultats présentés dans les colonnes 2 et 4 du tableau 3.1 illustrent le bénéfice lié à l'application d'un processus de régularisation sur la sortie d'un classificateur. Sur l'ensemble des niveaux de bruit considérés, cette méthode permet de réduire jusqu'à 13.8 % le nombre de pixels mal classés, avec une amélioration moyenne de 3.7 %.

Cette nouvelle méthode se révèle aussi plus précise que celle du chapitre précédent. L'amélioration est certes moins importante, puisque le nombre moyen de pixels mal classés est réduit de seulement 1.9 % (sur l'ensemble des niveaux de bruit). Il n'est cependant plus nécessaire de sélectionner manuellement les caractéristiques de textures les plus pertinentes. Nous pouvons par ailleurs noter une réduction sensible du nombre d'itérations nécessaires à l'obtention d'une solution stable : environ 1000 avec notre nouvelle approche contre plus de 2000 avec la précédente.

Les figures 3.9, 3.10, 3.11 illustrent les résultats obtenus par les différentes méthodes sur trois des images de test. Nous pouvons constater que l'application de la méthode de régularisation sur la sortie du classificateur permet de corriger un certain nombre d'erreurs de classification localisées. Comme nous pouvons le voir dans les figures 3.9 et 3.10, la régularisation peut cependant empêcher l'identification de régions de petite taille, notamment les trous présents dans certains objets.

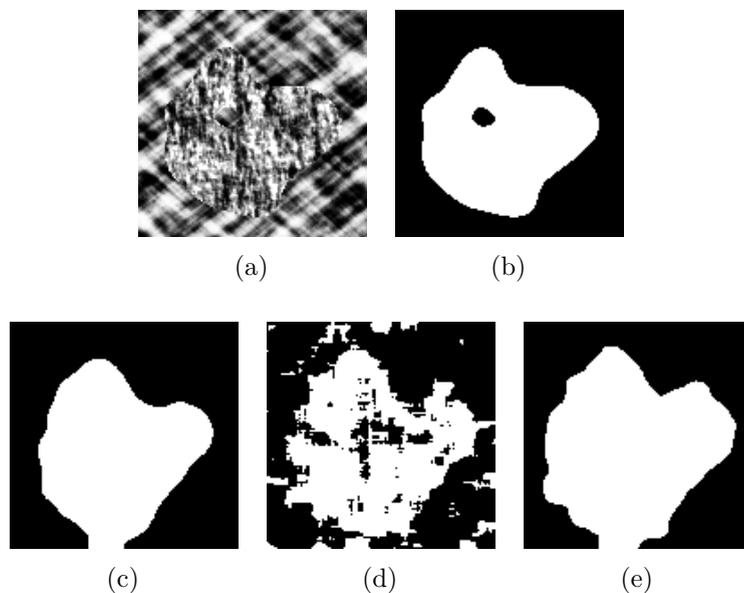


FIGURE 3.9 – Segmentations obtenues par les différentes méthodes sur l'image texturée artificielle 1 pour le niveau de bruit 8. (a) : image bruitée. (b) : vérité terrain. (c) : régularisation de classificateurs. (d) : classificateur seul. (e) : régularisation de graphes.

### 3.3. RÉSULTATS

---

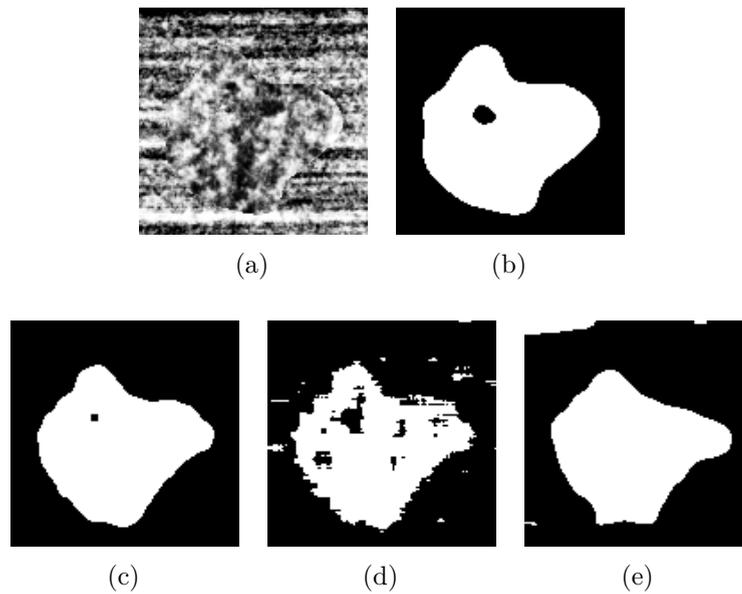


FIGURE 3.10 – Segmentations obtenues par les différentes méthodes sur l'image texturée artificielle 7 pour le niveau de bruit 8. (a) : image bruitée. (b) : vérité terrain. (c) : régularisation de classificateurs. (d) : classificateur seul. (e) : régularisation de graphes.

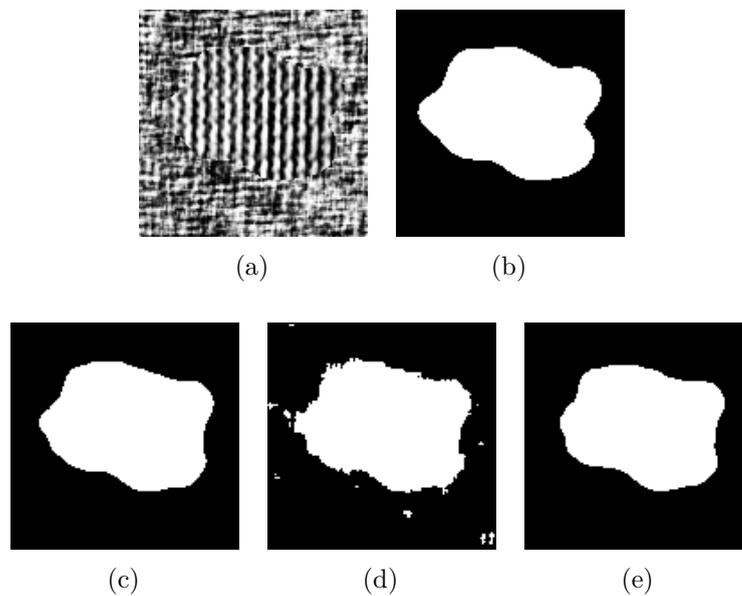


FIGURE 3.11 – Segmentations obtenues par les différentes méthodes sur l'image texturée artificielle 14 pour le niveau de bruit 8. (a) : image bruitée. (b) : vérité terrain. (c) : régularisation de classificateurs. (d) : classificateur seul. (e) : régularisation de graphes.

### 3.3.2 Segmentation d'images médicales 2D

Comme évoqué dans la section 3.2.5, l'une des applications de cette méthode est la segmentation d'images médicales à l'aide d'un classificateur expert. L'objectif de ce test est de valider ce mode d'utilisation.

Dans la série d'images échographiques utilisée lors des tests de la méthode précédente (voir section 2.3.3), trois d'entre elles présentent des textures similaires. Nous nous sommes donc servis de la vérité terrain de l'une d'entre elles afin d'entraîner un classificateur. Une fois celui-ci entraîné à distinguer la lésion des tissus sains selon leurs textures respectives, il a été utilisé pour segmenter les autres images.

Afin d'évaluer la précision de cette méthode, nous avons mesuré les performances de deux autres approches sur les mêmes images de test : la méthode du chapitre précédent et une méthode de segmentation par contours actifs supervisés [Olivier *et al.*, 2008].

L'approche par contours actifs utilise un classificateur binaire afin de guider l'évolution du contour. Ses performances ont été notamment validées dans un contexte de segmentation d'images médicales et pour une configuration similaire à la notre : un perceptron multicouche en tant que classificateur et des textures décrites à l'aide de caractéristiques d'Haralick. Nous avons ainsi pu appliquer une logique de test similaire : le classificateur a été entraîné sur la première image et utilisé sur les deux autres afin de guider l'évolution du contour.

En ce qui concerne la méthode de segmentation par régularisation de graphes, nous avons, comme pour le test précédent, réduit à un le nombre de paramètres.

Pour nos deux méthodes, différentes valeurs du paramètre d'attache aux données ont été testées, et le meilleur résultat conservé. Le tableau 3.2 présente les résultats obtenus par ces trois méthodes sur la série d'images échographiques. La figure 3.12 représente l'image utilisée pour entraîner le classificateur tandis que les figures 3.13 et 3.14 illustrent les résultats obtenus sur les deux autres images par les différentes méthodes.

Image	Régularisation de graphes	Contour actif	Régularisation de classificateurs
2	2.20	2.52	<b>1.84</b>
3	3.19	4.12	<b>2.0</b>

TABLE 3.2 – Comparaison des différentes méthodes de segmentation sur la série d'images échographiques. Les valeurs indiquées représentent la distance générique d'anomalie normalisée.

Les résultats obtenus confirment que ce mode d'utilisation est parfaitement applicable : un classificateur expert permet de segmenter de multiples images de nature similaire. En fixant la valeur du paramètre de régularisation, il est donc envisageable de rendre la phase de segmentation totalement automatique.

Nous pouvons par ailleurs constater que cette nouvelle méthode est là aussi plus précise que la précédente, ce qui permet de confirmer la supériorité du classificateur face au critère d'attache aux données de Chan et Vese. Le bénéfice est donc double, puisque nous obtenons à la fois une simplification du paramétrage de la méthode ainsi qu'une pondération automatique des différentes caractéristiques de textures.

### 3.3. RÉSULTATS

---

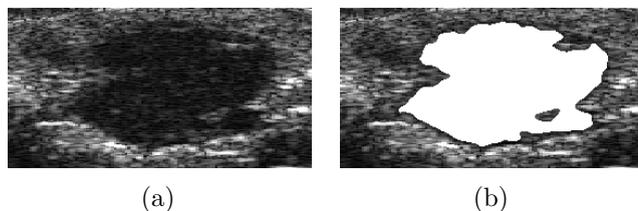


FIGURE 3.12 – Image échographique (a) et vérité terrain (b) utilisées pour entraîner le classificateur.

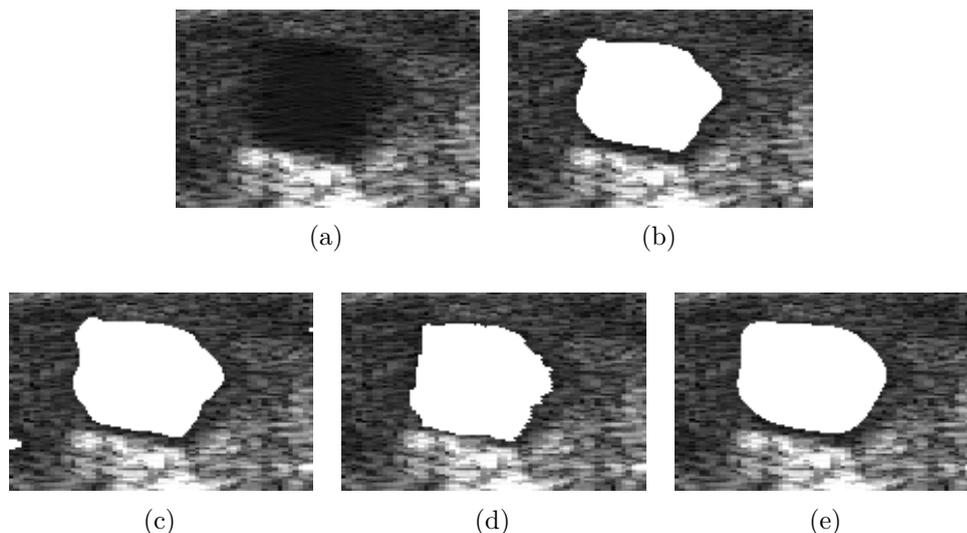


FIGURE 3.13 – Résultats obtenus sur l'image échographique 2D n° 2 par les différentes méthodes. (a) : image originale. (b) : vérité terrain. (c) : régularisation de graphes. (d) : contours actifs. (e) : régularisation de classificateurs.

Afin d'illustrer la souplesse de notre approche, nous l'avons appliquée à de la segmentation d'images acquises par microscopie confocale. Les images obtenues par cette technique sont très différentes de celles obtenues en échographie, puisqu'elles sont relativement nettes. Par ailleurs, la microscopie confocale produit naturellement des images 3D, sous forme de séries de coupes acquises à différentes profondeurs.

L'image considérée pour ce test correspond à une vue du système olfactif de l'abeille, les éléments circulaires présents dans l'image 3.15a étant les glomérules<sup>9</sup>.

Pour ce test, nous avons voulu segmenter plusieurs coupes successives d'une même image. À partir de la vérité terrain de la première coupe, nous avons donc entraîné un classificateur à distinguer les glomérules des autres tissus, puis segmenté les coupes suivantes. Le tableau 3.3 résume les résultats obtenus sur une série de coupes. La segmentation obtenue pour la coupe numéro 4 est illustrée figure 3.15c.

Comme l'illustre l'image 3.15c, notre approche peine à distinguer la frontière séparant

---

9. Structures chargées de transmettre aux neurones les signaux émis par les cellules chémoréceptrices (capable de détecter des substances chimiques).

### 3.3. RÉSULTATS

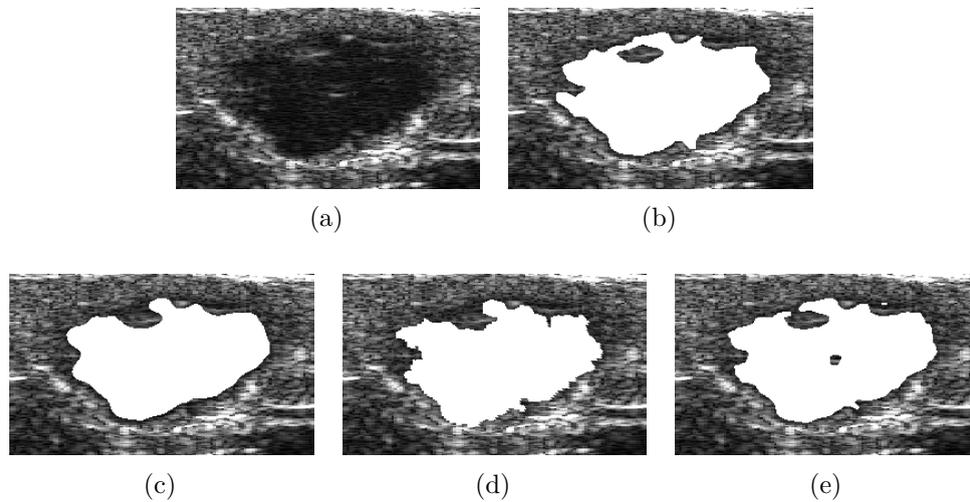


FIGURE 3.14 – Résultats obtenus sur l’image échographique 2D n° 3 par les différentes méthodes. (a) : image originale. (b) : vérité terrain. (c) : régularisation de graphes. (d) : contours actifs. (e) : régularisation de classificateurs.

Image	Distance générique d’anomalie normalisée	Distance de Hausdorff moyenne
2	7.1	0.86
3	8.1	0.90
4	9.1	0.92
5	8.9	0.66

TABLE 3.3 – Évaluation de la précision de la méthode de segmentation par régularisation de classificateurs sur des images de microscopie confocale.

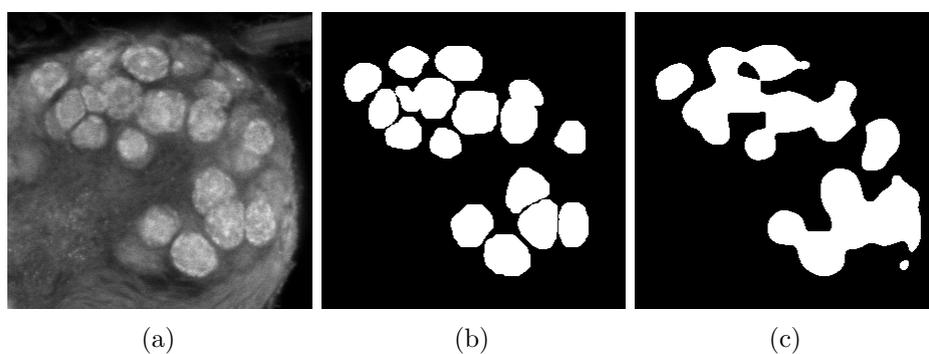


FIGURE 3.15 – Résultat obtenu par la méthode de segmentation par régularisation de classificateurs sur l’image de microscopie confocale. (a) : image originale. (b) : vérité terrain. (c) : segmentation.

des glomérules très proches. Ce comportement est essentiellement dû à l’utilisation de caractéristiques de textures qui sont par définition non-locales et font disparaître ce genre

### 3.3. RÉSULTATS

---

de détails. La longueur totale des frontières étant importante dans ces images, le nombre de pixels mal classés augmente rapidement (voir la seconde colonne du tableau 3.3). Afin d'en tenir compte, nous pouvons évaluer la précision de notre méthode à l'aide de la distance de Hausdorff moyenne, qui mesure la distance moyenne entre un pixel mal classé et le contour le plus proche. Comme l'indique la seconde colonne du tableau 3.3), les pixels mal classés se trouvent à moins d'un pixel de distance des contours de la vérité terrain. Selon ce critère, notre méthode reste donc précise.

#### 3.3.3 Segmentation d'images IRM 3D

Dans ce dernier test, nous avons souhaité évaluer les performances de notre méthode sur un problème de segmentation qui soit à la fois multiclasse et 3D.

Concevoir une image artificielle texturée respectant ces critères est relativement complexe. Alors que plusieurs techniques d'imagerie médicale sont capables d'acquérir de telles images, peu de jeux de données sont disponibles. L'un des rares à être fourni accompagné des vérités terrain est mis à disposition par le Centre d'Analyse Morphométrique de l'Hôpital Général du Massachusset<sup>10</sup>. Ce jeu de données est composé de vingt images de cerveaux sains, acquises en imagerie par résonance magnétique (IRM). Chaque image est composée d'environ 55 coupes mesurant 256 pixels de côté. Quatre régions sont labellisées dans chaque image : substance blanche, substance grise, liquide céphalo-rachidien et autres (crâne, arrière-plan, globes oculaires...). La figure 3.16 représente une coupe extraite d'une de ces images ainsi que la vérité terrain associée.

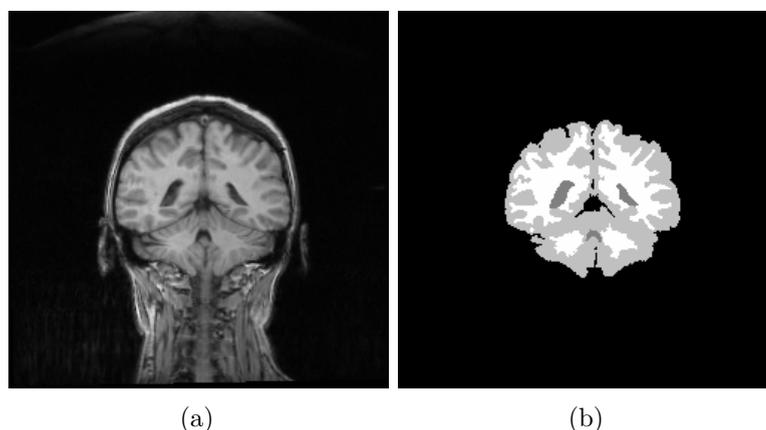


FIGURE 3.16 – Exemple d'image de cerveau acquise en imagerie par résonance magnétique. (a) : image acquise. (b) : vérité terrain. Dans la vérité terrain, les pixels noirs correspondent à la classe « autre », les gris foncé au liquide céphalo-rachidien, les gris clair à la substance grise, et les blancs à la substance blanche.

Afin de traiter une telle image 3D, nous nous sommes placés dans un contexte de segmentation interactive : le praticien doit labelliser une partie des données afin d'entraîner le classificateur, lequel pourra ensuite être utilisé pour segmenter l'image 3D entière. Dans notre cas, nous avons entraîné le classificateur à partir de la vérité terrain d'une des coupes.

10. <http://www.cma.mgh.harvard.edu/ibsr/>

### 3.3. RÉSULTATS

---

Le graphe construit pour ces images correspond à un 6-voisinage, dans lequel la valeur 1 fait office de mesure de similarité. Concernant les caractéristiques d'Haralick, nous avons fixé à 16 le nombre de niveaux de gris, et considéré le décalage (1, 0). Les régions que nous souhaitons identifier étant parfois de très petite taille, nous avons dû réduire la taille de la fenêtre de calcul. Les tests réalisés ont montré qu'un diamètre de 9 pixels était tout de même nécessaire afin de correctement caractériser les textures.

Nous avons par ailleurs fixé le nombre d'itérations de régularisation à 50, l'objectif étant ici de corriger les erreurs de classification évidentes et non d'agir sur la régularité des contours. Il est en effet nécessaire d'éviter une régularisation trop importante, au risque de faire disparaître les éléments de petite taille que nous souhaitons identifier.

La figure 3.17 présente le résultat obtenu sur la coupe dont la vérité terrain est illustrée figure 3.16. Les figures 3.18, 3.19 et 3.20 présentent, sous forme de maillage 3D, le résultat obtenu pour l'image complète.

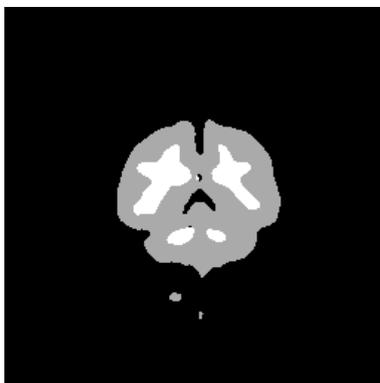


FIGURE 3.17 – Segmentation obtenue sur une des coupes de l'image IRM 3D.

Nous constatons que la segmentation respecte la forme globale des régions. Les éléments de petite taille sont en revanche absents du résultat. Cet effet est essentiellement dû aux caractéristiques de textures qui, du fait de leur non localité, ne permettent pas ce niveau de détail. La région correspondant au liquide céphalo-rachidien, qui est de petite taille, n'est d'ailleurs pas détectée.

Afin d'évaluer la précision de notre méthode, nous avons comparé la segmentation obtenue à la vérité terrain. Nous avons de nouveau choisi de mesurer la distance de Hausdorff moyenne car celle-ci permet de mieux apprécier la précision globale de la segmentation malgré le manque de détails. Le tableau 3.4 présente les résultats obtenus avec et sans régularisation pour chacune des classes identifiées.

Pour les deux régions du cerveau qui ont été identifiées, la distance générique d'anomalie montre un taux de pixels mal classés non négligeable. Ce mauvais score est malheureusement amplifié du fait de la petite taille de ces régions. La distance de Hausdorff moyenne nous indique cependant que ces pixels mal classés restent très proches des contours de la vérité terrain, confirmant ainsi la cohérence de la segmentation calculée. La distance de Hausdorff moyenne est par ailleurs plus importante en l'absence de régularisation, ce qui justifie de nouveau la validité de notre approche.

### 3.3. RÉSULTATS

---

Région	Avec régularisation (DGA)	Avec régularisation ( $\bar{H}$ )	Sans régularisation ( $\bar{H}$ )
Autre	2	0.03	0.03
Substance grise	20	<b>1.54</b>	1.90
Substance blanche	50	<b>2.34</b>	2.78

TABLE 3.4 – Évaluation de la précision de la méthode sur une image d'IRM 3D. DGA : distance générique d'anomalie.  $\bar{H}$  : distance de Hausdorff moyenne.

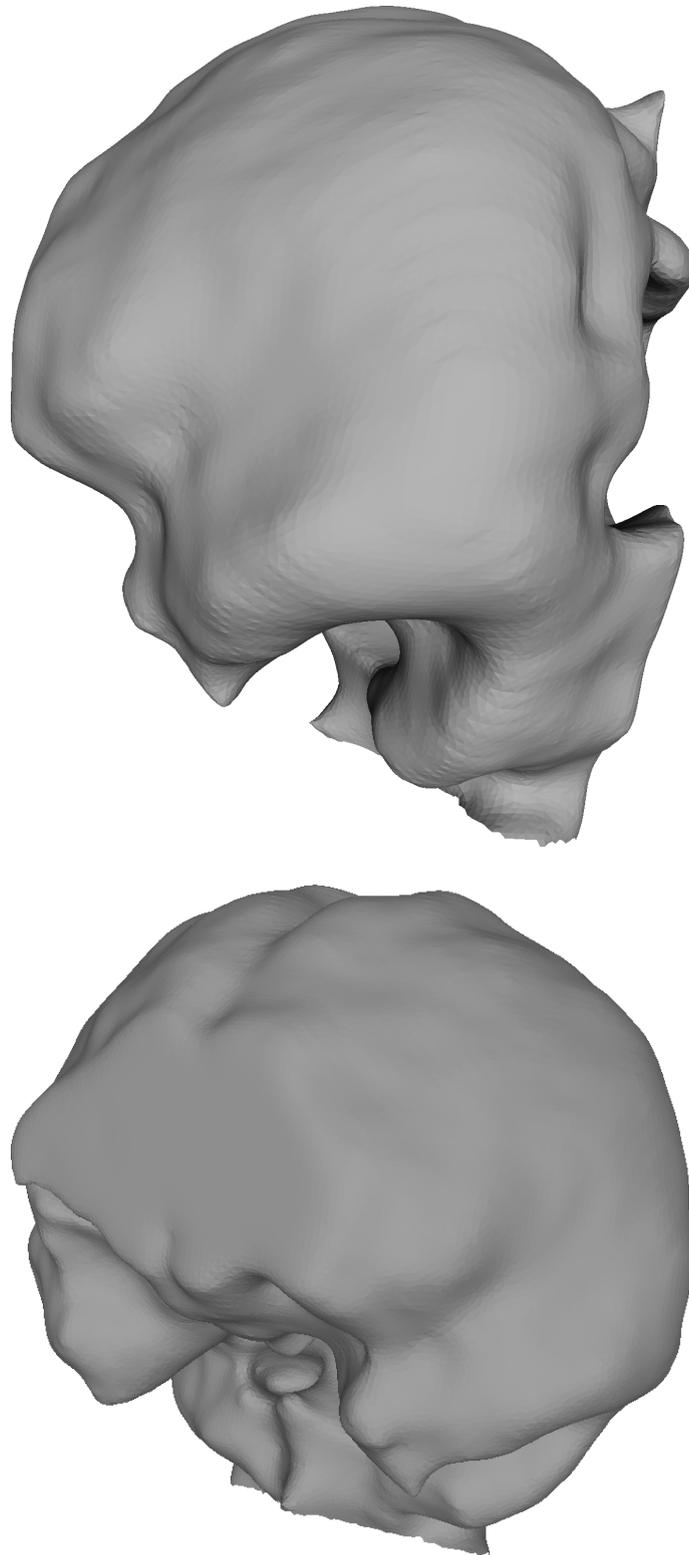


FIGURE 3.18 – Représentation sous forme de maillage de la région correspondant à la substance grise.

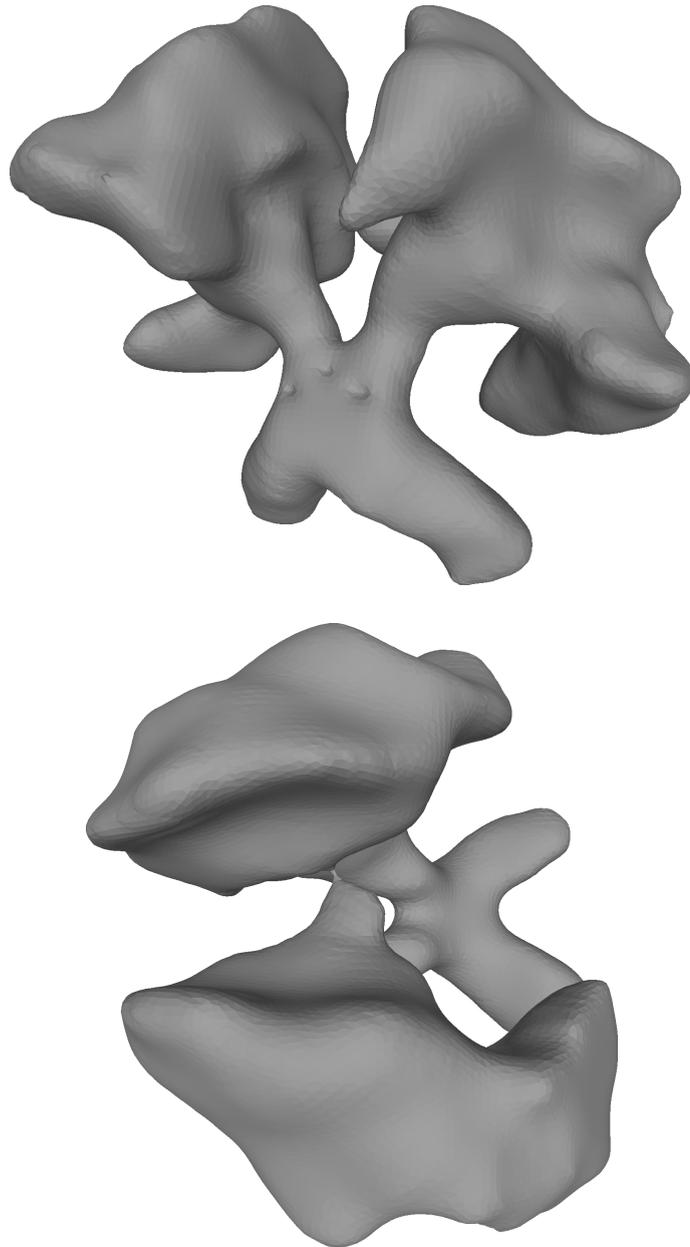


FIGURE 3.19 – Représentation sous forme de maillage de la région correspondant à la substance blanche.

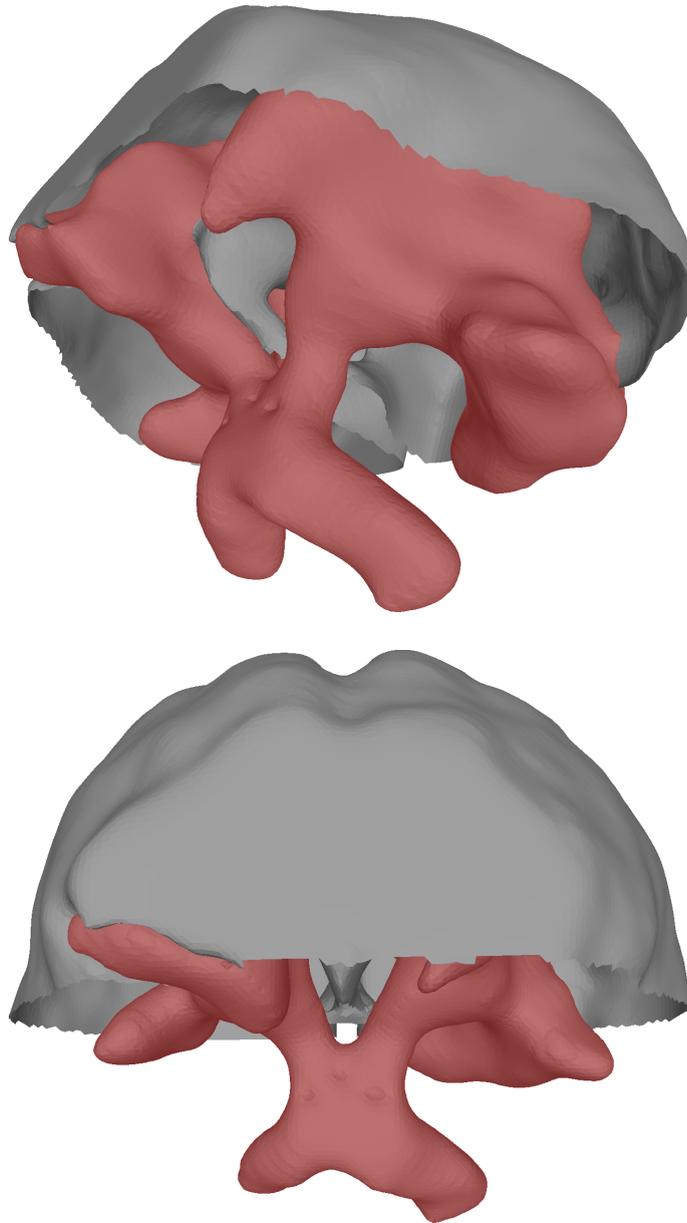


FIGURE 3.20 – Représentation sous forme de maillage de la région correspondant à la substance blanche (en rouge) et à la substance grise. Le maillage de la substance grise a été découpé afin de laisser apparaître la substance blanche qui se trouve à l'intérieur.

### 3.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode de segmentation d'images qui combine un classificateur supervisé à un algorithme de régularisation de graphes. L'objectif de cette approche était d'offrir une solution au problème de détermination des paramètres inhérents à la méthode de segmentation d'images texturées par régularisation de graphes présentée dans le chapitre 2.

Pour rappel, cette dernière se base sur un critère d'attache aux données inspiré des travaux de Chan et Vese [Chan *et al.*, 2000] afin d'identifier la texture des différentes régions de l'image. Comme l'ont cependant montré nos tests (voir section 2.3), ce critère nécessite la détermination d'un ensemble de paramètres afin de pondérer les différentes caractéristiques de textures d'après leur pertinence. Malheureusement, il n'existe actuellement aucune méthode supervisée permettant d'effectuer ce paramétrage.

Grâce à l'utilisation d'un classificateur supervisé, nous sommes en mesure de modéliser plus précisément le problème puisque la phase d'apprentissage tient compte de la pertinence des différentes caractéristiques. Un classificateur est donc utilisé en lieu et place du critère d'attache aux données. Les tests réalisés ont confirmé une amélioration de la précision du processus de segmentation tout en offrant un paramétrage simplifié de la méthode. Étant donné qu'aucun descripteur n'est adapté à tous types de textures, nous pouvons par ailleurs envisager d'en multiplier le nombre et ainsi d'aboutir à un outil de segmentation générique.